- Unit 2- Virtualization

Virtualization

- Cloud computing is based on virtualization.
- It is partitioning of single physical server into multiple logical servers. Once the physical server is divided, each logical server behaves like a physical server and can run an operating system and applications independently.
- For software developers and testers virtualization comes very handy, as it allows developer to write code that runs in many different environments and more importantly to test that code.
- Creation of a virtual machine over existing operating system and hardware is known as **Hardware Virtualization**. A Virtual machine provides an environment that is logically separated from the underlying hardware.
- The machine on which the virtual machine is going to create is known as **Host Machine** and that virtual machine is referred as a **Guest Machine**
- This virtual machine is managed by a software or firmware known as **hypervisor or virtual machine monitor (VMM)**

- Levels of virtualization

1) **Instruction Set Architecture level**
2) **Hardware abstraction level**
3) **Operating System level**
4) **Library Support level**
5) **User Application level**

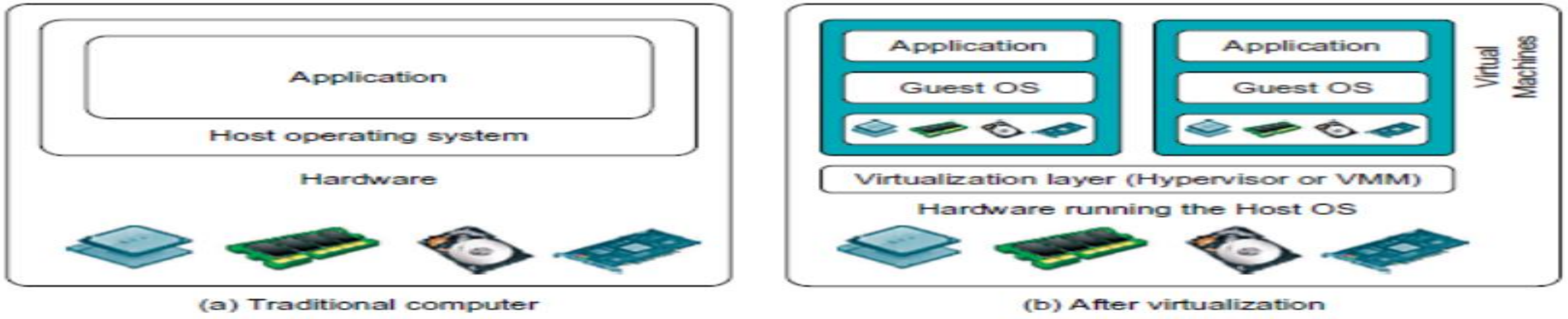


FIGURE 3.1

The architecture of a computer system before and after virtualization, where VMM stands for virtual machine monitor.

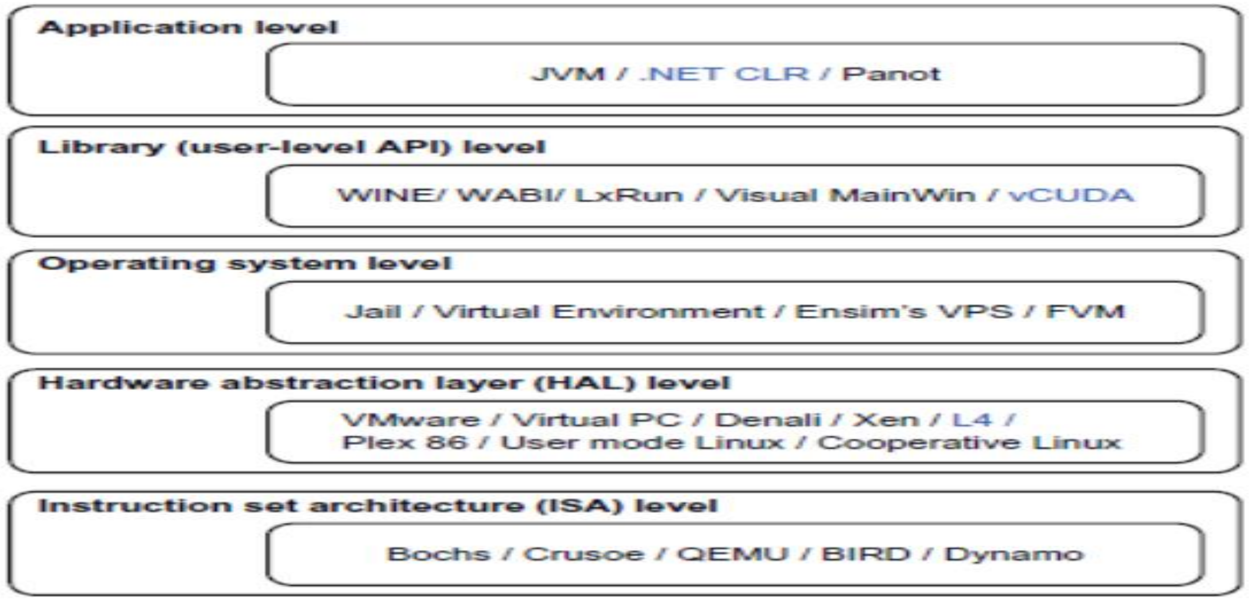

Application level
JVM / .NET CLR / Panot
Library (user-level API) level
WINE/ WABI/ LxRun / Visual MainWin / vCUDA
Operating system level
Jail / Virtual Environment / Ensim's VPS / FVM
Hardware abstraction layer (HAL) level
VMware / Virtual PC / Denali / Xen / L4 / Plex 86 / User mode Linux / Cooperative Linux
Instruction set architecture (ISA) level
Bochs / Crusoe / QEMU / BIRD / Dynamo

FIGURE 3.2

Virtualization ranging from hardware to applications in five abstraction levels.

User Application level

- *Virtualization at the application level virtualizes an application as a VM. On a traditional OS, an application often runs as a process. Therefore, application-level virtualization is also known as process-level virtualization. The most popular approach is to deploy high level language (HLL) VMs. In this scenario, the virtualization layer sits as an application program on top of the operating system, and the layer exports an abstraction of a VM that can run programs written and compiled to a particular abstract machine definition. Any program written in the HLL and compiled for this VM will be able to run on it. The Microsoft .NET CLR and Java Virtual Machine (JVM) are two good examples of this class of VM.*
- App virtualization (application virtualization) is the separation of an installation of an application from the client computer accessing it.
- From the user's perspective, the application works just like it would if it lived on the user's device. The user can move or resize the application window, as well as carry out keyboard and mouse operations

How application virtualization works

- Although there are multiple ways to virtualize applications, IT teams often take a server-based approach, delivering the applications without having to install them on individual desktops. Instead, administrators implement remote applications on a server in the company's data center or with a hosting service, and then deliver them to the users' desktops.
- To make this possible, IT must use an application virtualization product. Application virtualization vendors and their products include Microsoft App-V, Citrix XenApp, Parallels Remote Application Server, and VMware ThinApp or App Volumes -- both of which are included with VMware Horizon View. VMware also offers Horizon Apps to further support app virtualization.
- The virtualization software essentially transmits the application as individual pixels from the hosting server to the desktops using a remote display protocol such as Microsoft RemoteFX, Citrix HDX, or VMware View PCoIP or Blast Extreme. The user can then access and use the app as though it were installed locally. Any user actions are transmitted back to the server, which carries them out.
- With streaming applications, the virtualized application runs on the end user's local computer. When a user requests an application, the local computer downloads its components on demand. Only certain parts of an application are required to launch the app; the remainder download in the background as needed.
- Once completely downloaded, a streamed application can function without a network connection. Various models and degrees of isolation ensure that streaming applications do not interfere with other applications, and that they can be cleanly removed when the user closes the application.

- Example JVM, asp.net, whatsapp, paytm

Instruction Set Architecture level

- At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine. For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation.
- With this approach, it is possible to run a large amount of legacy binary code written for various processors on any given new hardware host machine.
- Instruction set emulation leads to virtual ISAs created on any hardware machine.
- The basic emulation method is through code interpretation.
- An interpreter program interprets the source instructions to target instructions one by one. One source instruction may require tens or hundreds of native target instructions to perform its function. Obviously, this process is relatively slow. For better performance, dynamic binary translation is desired. This approach translates basic blocks of dynamic source instructions to target instructions. The basic blocks can also be extended to program traces or super blocks to increase translation efficiency.
- Instruction set emulation requires binary translation and optimization.

Hardware Virtualization: When the virtual machine software or virtual machine manager *(VMM) is directly installed on the hardware system* is known as hardware virtualization. After virtualization of hardware system we can install different operating system on it and run different applications on those OS.

Operating system Virtualization: When the virtual machine software or virtual machine manager *(VMM) is installed on the Host operating system* instead of directly on the hardware system is known as operating system virtualization.

Library support level : *Most applications use APIs exported by user-level libraries rather than using lengthy system calls by the OS. Since most systems provide well-documented APIs, such an interface becomes another candidate for virtualization. Virtualization with library interfaces is possible by controlling the communication link between applications and the rest of a system through API hooks.*

Merits of different approaches

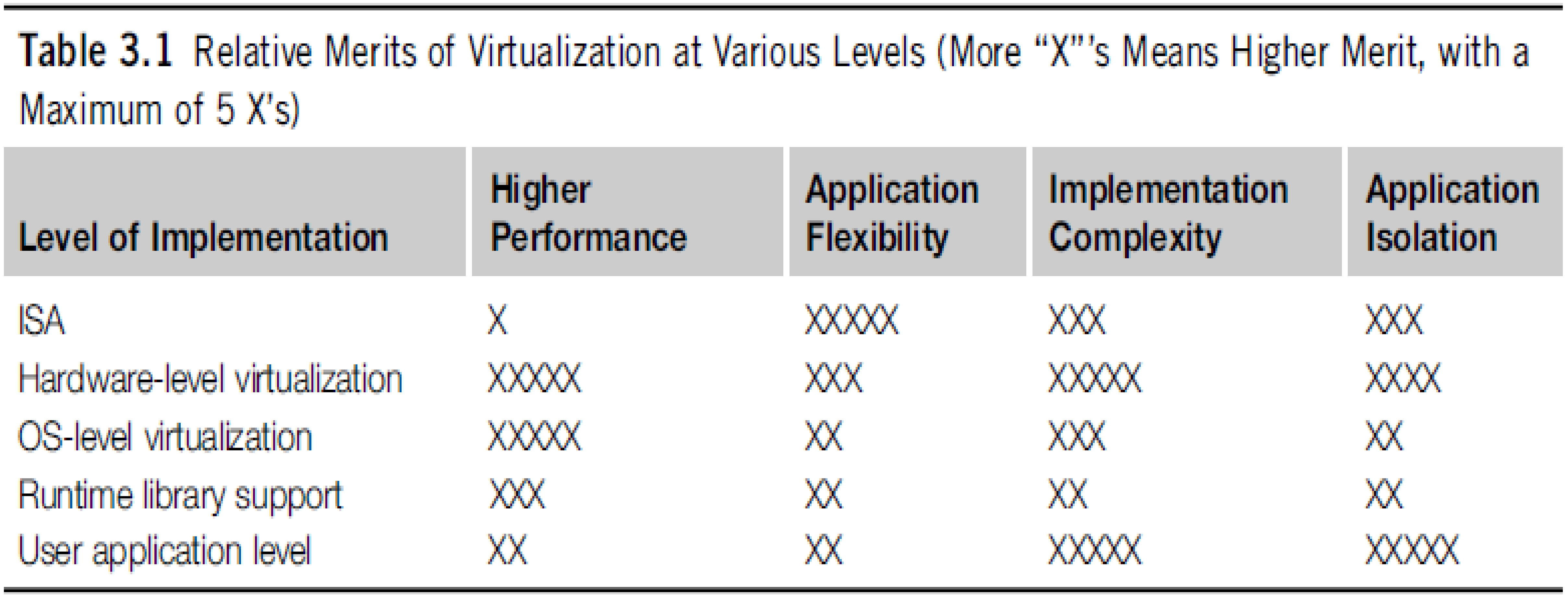


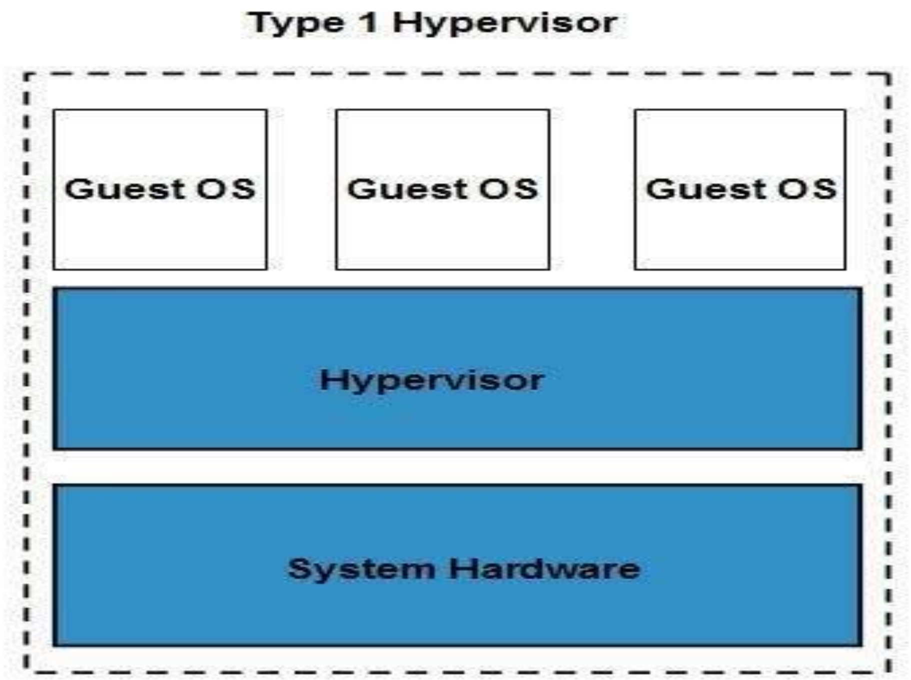
Table 3.1 Relative Merits of Virtualization at Various Levels (More "X"'s Means Higher Merit, with a Maximum of 5 X's)

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	X	XXXXX	XXX	XXX
Hardware-level virtualization	XXXXX	XXX	XXXXX	XXXX
OS-level virtualization	XXXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXXX	XXXXX

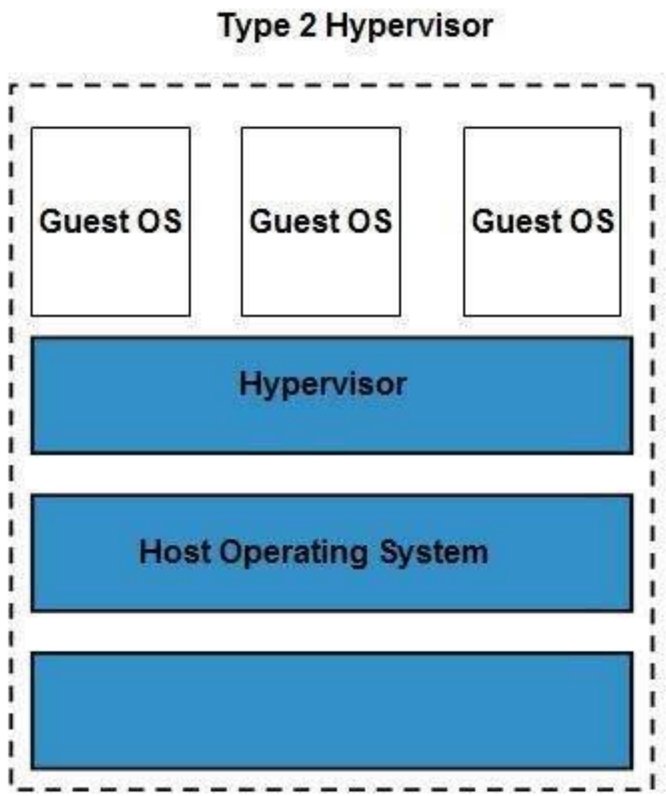
Hardware virtualization

- The hypervisor is a firmware or low-level program that acts as a virtual machine manager. The hypervisor provides hypercalls for the guest OSes and applications. They are of two types:
- Type 1 hypervisor & Type 2 hypervisor
- Type 1 hypervisor: executes on bare system. LynxSecure, RTS Hypervisor, Oracle VM, Sun xVM Serve examples of Type 1 hypervisor

Type 1 hypervisor does not have any host operating System because they are installed on a bare system

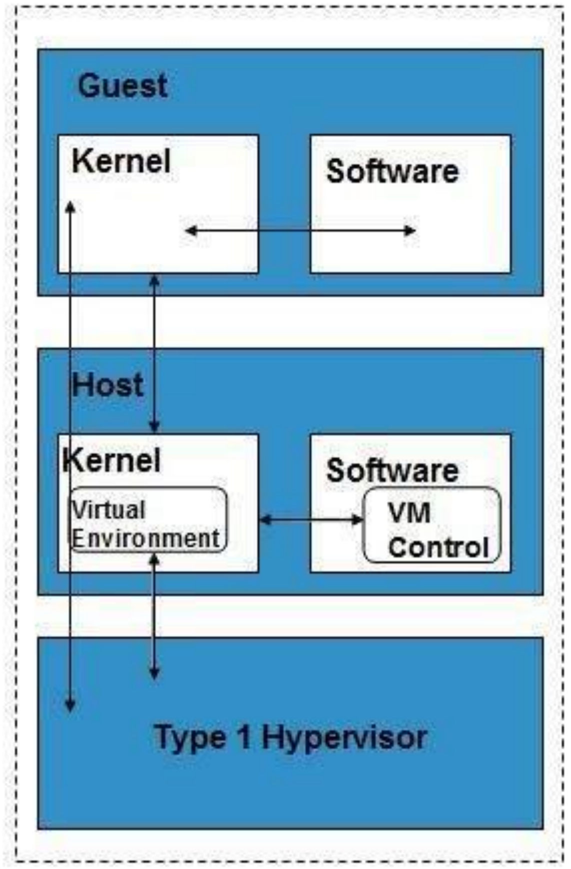


- Type 2 hypervisor :is a software interface that emulates the devices with which a system normally interacts. Containers, KVM, Microsoft Hyper V, VMWare Fusion, Virtual Server 2005 R2, Windows Virtual PC and **VMWare workstation 6.0** are examples of Type 2 hypervisor



Types of hardware virtualization

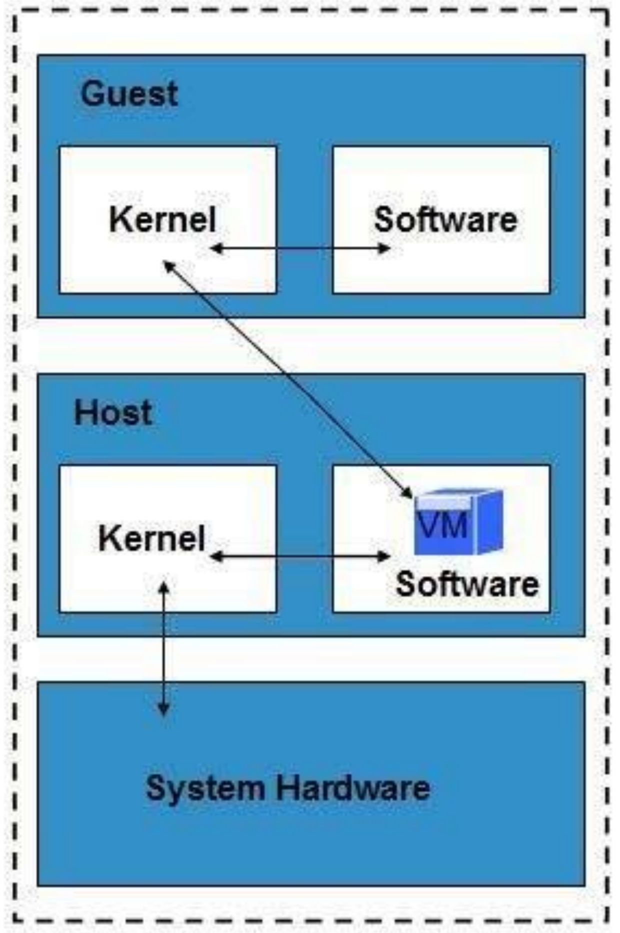
- Full virtualization
- Emulation virtualization
- Paravirtualization



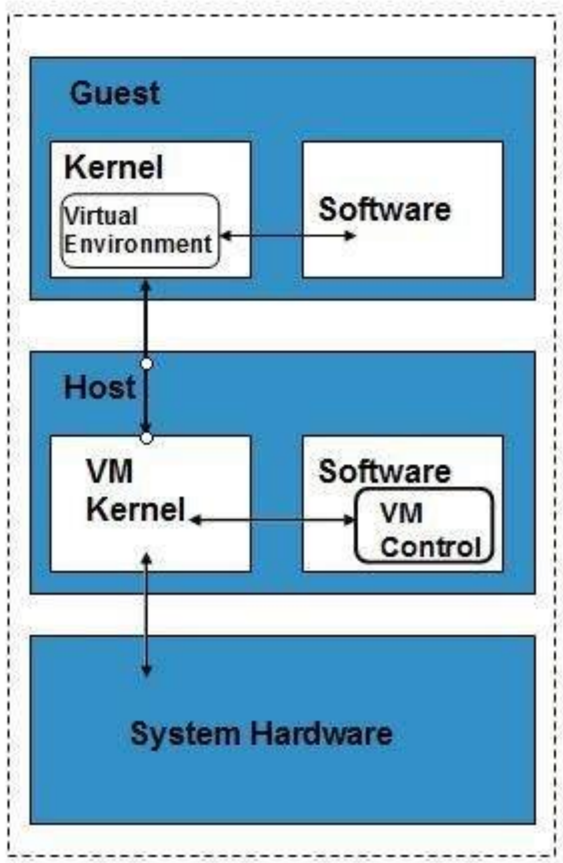
Full virtualization-In **full virtualization,** the underlying hardware is completely simulated. Guest software does not require any modification to run.

In this the guest operating system is unaware that it is in a virtualized environment, and therefore hardware is virtualized by the hypervisor so that the guest can issue commands to what it thinks is actual hardware, but really are just simulated hardware devices created by the hypervisor

- Emulation virtualization : In **Emulation,** the virtual machine simulates the hardware and hence becomes independent of it. In this, the guest operating system does not require modification.



- Paravirtualization: In **Paravirtualization,** the hardware is not simulated. The guest software run their own isolated domains. The guest operating system (the one being virtualized) is aware that it is a guest and accordingly has drivers that, instead of issuing hardware commands, simply issues commands directly to the host operating system.



Software virtualization

- It provides the ability to the main computer to run and create one or more virtual environments. It is used to enable a complete computer system in order to allow a guest OS to run. For instance letting Linux to run as a guest that is natively running a Microsoft Windows OS (or vice versa, running Windows as a guest on Linux).
- Examples
- Operating System Virtualization – hosting multiple OS on the native OS
- Application Virtualization – hosting individual applications in a virtual environment separate from the native OS
- Service Virtualization – hosting specific processes and services related to a particular application

Operating System Level

- Why OS level?
- It is slow to initialize a hardware-level VM because each VM creates its own image from scratch. In a cloud computing environment, perhaps thousands of VMs need to be initialized simultaneously. Besides slow operation, storing the VM images also becomes an issue. As a matter of fact, there is considerable repeated content among VM images
- Operating system virtualization inserts a virtualization layer inside an operating system to partition a machine's physical resources. It enables multiple isolated VMs within a single operating system kernel. This kind of VM is often called a virtual execution environment (VE), Virtual Private System (VPS), or simply container. From the user's point of view, VEs look like real servers. This means a VE has its own set of processes, file system, user accounts, network interfaces with IP addresses, routing tables, firewall rules, and other personal settings. Although VEs can be customized for different people, they share the same operating system kernel.

Disadvantages of OS extensions

- The guest OS should be of same family.
- For example, a Windows distribution such as Windows XP cannot run on a Linux-based container. However, users of cloud computing have various preferences. Some prefer Windows and others prefer Linux or other operating systems. Therefore, there is a challenge for OS-level virtualization in such cases.

Advantages of OS extensions:
(1) VMs at the operating system level have minimal startup/shutdown costs, low resource requirements, and high scalability;
(2) for an OS-level VM, it is possible for a VM and its host environment to synchronize state changes when necessary.

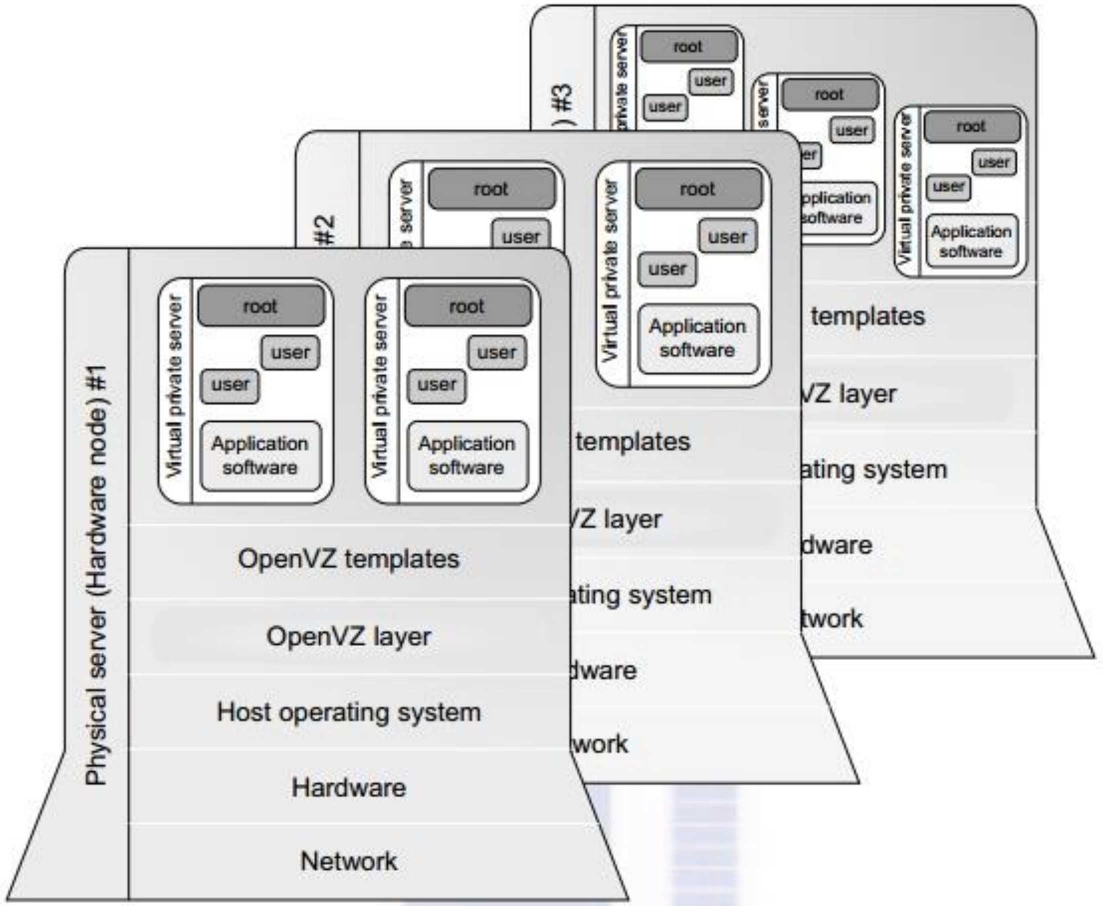


FIGURE 3.3

The OpenVZ virtualization layer inside the host OS, which provides some OS images to create VMs quickly.

Network virtualization

- In network virtualization, multiple sub-networks can be created on the same physical network, which may or may not is authorized to communicate with each other. This enables restriction of file movement across networks and enhances security, and allows better monitoring and identification of data usage which lets the network administrator's scale up the network appropriately. It also increases reliability as a disruption in one network doesn't affect other networks, and the diagnosis is easier.

Subtypes:

- Internal network: Enables a single system to function like a network
- External network: Consolidation of multiple networks into a single one, or segregation of a single network into multiple ones

Storage virtualization

- In this type of virtualization, multiple network storage resources are present as a single storage device for easier and more efficient management of these resources. It provides various advantages as follows:
- Improved storage management in a heterogeneous IT environment
- Easy updates, better availability
- Reduced downtime
- Better storage utilization
- Automated management

In general, there are two types of storage virtualization:

- **Block-** It works before the file system exists. It replaces controllers and takes over at the disk level.
- **File-** The server that uses the storage must have software installed on it in order to enable file-level usage

Middleware support for virtualization

- Library-level virtualization is also known as user-level Application Binary Interface (ABI) or API emulation. This type of virtualization can create execution environments for running alien programs on a platform rather than creating a VM to run the entire operating system. API call interception and remapping are the key functions performed.

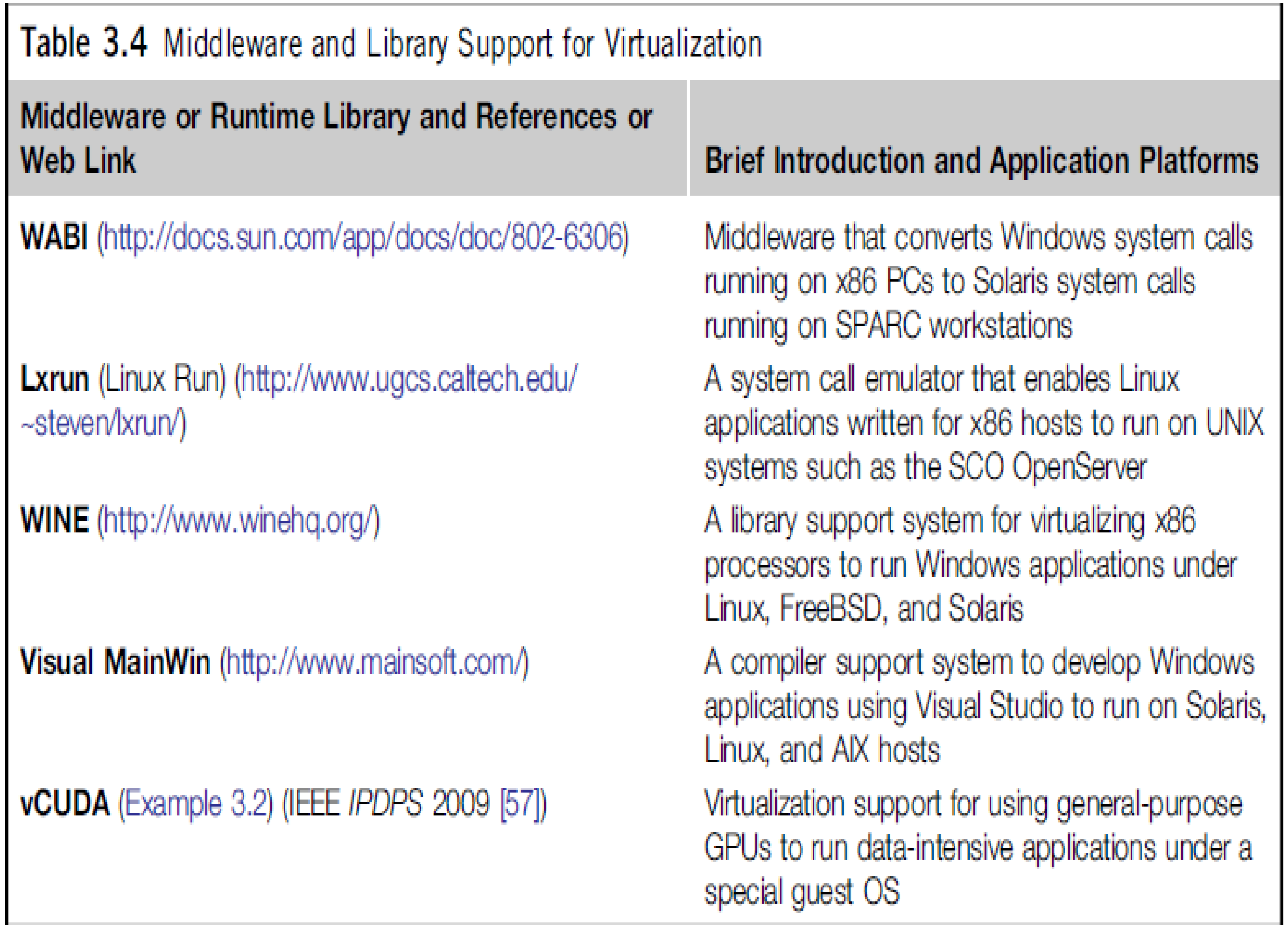


Table 3.4 Middleware and Library Support for Virtualization

Middleware or Runtime Library and References or Web Link	Brief Introduction and Application Platforms
WABI (http://docs.sun.com/app/docs/doc/802-6306)	Middleware that converts Windows system calls running on x86 PCs to Solaris system calls running on SPARC workstations
Lxrun (Linux Run) (http://www.ugcs.caltech.edu/~steven/lxrun/)	A system call emulator that enables Linux applications written for x86 hosts to run on UNIX systems such as the SCO OpenServer
WINE (http://www.winehq.org/)	A library support system for virtualizing x86 processors to run Windows applications under Linux, FreeBSD, and Solaris
Visual MainWin (http://www.mainsoft.com/)	A compiler support system to develop Windows applications using Visual Studio to run on Solaris, Linux, and AIX hosts
vCUDA (Example 3.2) (IEEE *IPDPS* 2009 [57])	Virtualization support for using general-purpose GPUs to run data-intensive applications under a special guest OS

VMM Design Requirements and Providers

- Each time programs access the hardware the VMM captures the process. In this sense, the VMM acts as a traditional OS. One hardware component, such as the CPU, can be virtualized as several virtual copies. Therefore, several traditional operating systems which are the same or different can sit on the same set of hardware simultaneously.

There are three requirements for a VMM.

- First, a VMM should provide an environment for programs which is essentially identical to the original machine.
- Second, programs run in this environment should show, at worst, only minor decreases in speed.
- Third, a VMM should be in complete control of the system resources.
- Any program run under a VMM should exhibit a function identical to that which it runs on the original machine directly. Two possible exceptions in terms of differences are permitted with this requirement: differences caused by the availability of system resources and differences caused by timing dependencies. The former arises when more than one VM is running on the same machine.

VIRTUALIZATION STRUCTURES/TOOLS AND MECHANISMS

Three classes of VM architecture exists

- Hypervisor and Xen Architecture
- Binary Translation with Full Virtualization
- Para-Virtualization with compiler support

Hypervisor (VMM-Virtual Machine Monitor) and Xen Architecture

The hypervisor software sits directly between the physical hardware and its OS. The hypervisor provides **hypercalls** for the guest OS and applications. Depending on functionality a hypervisor is of two types namely

- Micro-kernel –eg Microsoft Hyper-V
- Monolithic – eg VMware ESX

Micro-kernel hypervisor- includes only the basic and unchanging functions(physical memory mgt and processor scheduling).The device drivers and other changeable components are outside the hypervisor.

A monolithic hypervisor implements all the aforementioned functions, including those of the device drivers. Therefore, the size of the hypervisor code

of a micro-kernel hyper-visor is smaller than that of a monolithic hypervisor.

Xen Architecture

- It is a open source hypervisor program developed by Cambridge university.
- It is a micro-kernel hypervisor- separates the policy and mechanism
- It does not include any device drivers natively
- It just provides mechanism by which a guest OS can have direct access to the physical device
- Due to this the size of Xen is very small
- Eg Citrix XenServer and Oracle VM
- Many guest OS runs but only one guest domain 0 controls all.
- It is first loaded when Xen boots without any file system drivers being available. Domain 0 is designed to access hardware directly and manage devices. Therefore, one of the responsibilities of Domain 0 is to allocate and map hardware resources for the guest domains (the Domain U domains).

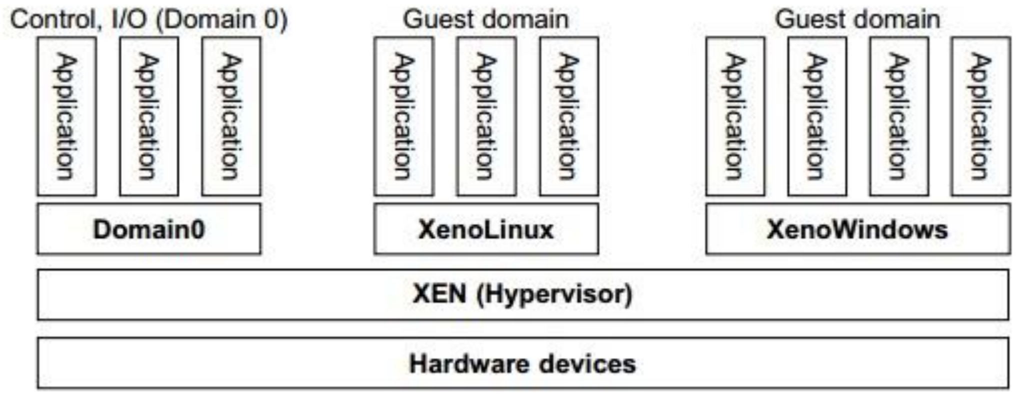


FIGURE 3.5

The Xen architecture's special domain 0 for control and I/O, and several guest domains for user applications.

Binary translation with full virtualization

- The guest OSes and their applications consist of noncritical and critical instructions
- With full virtualization, noncritical instructions run on the hardware directly while critical instructions are discovered and replaced with traps into the VMM to be emulated by software.
- **Why are only critical instructions trapped into the VMM**? This is because binary translation can incur a large performance overhead. Noncritical instructions do not control hardware or threaten the security of the system, but critical instructions do

Binary translation of Guest OS requests using VMM

- This approach was implemented by VMware and many other software companies. As shown in Figure 3.6, VMware puts the VMM at Ring 0 and the guest OS at Ring 1. The VMM scans the instruction stream and identifies the privileged, control- and behavior-sensitive instructions. When these instructions are identified, they are trapped into the VMM, which emulates the behavior of these instructions. The method used in this emulation is called binary translation. Therefore, full virtualization combines binary translation and direct execution. The guest OS is completely decoupled from the underlying hardware. Consequently, the guest OS is unaware that it is being virtualized.

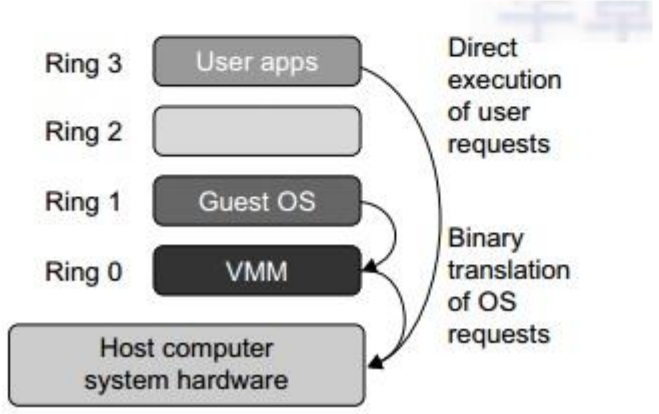

Ring 3
User apps
Ring 2
Ring 1
Guest OS
Ring 0
VMM
Host computer system hardware
Direct execution of user requests
Binary translation of OS requests

FIGURE 3.6

Indirect execution of complex instructions via binary translation of guest OS requests using the VMM plus direct execution of simple instructions on the same host.

Host-based virtualization

1. An alternative VM architecture is to install a **virtualization layer on top of the host OS.**

2. This host OS is still responsible for managing the hardware.

3. The guest OSes are installed and run on top of the virtualization layer.

4. Dedicated applications may run on the VMs.

Certainly, some other applications can also run with the host OS directly. This host-based architecture has some distinct advantages, as enumerated next.

First, the user can install this VM architecture without modifying the host OS. The virtualizing software can rely on the host OS to provide device drivers and other low-level services. This will simplify the VM design and ease its deployment

second , Compared to the hypervisor/VMM architecture, the performance of the host-based architecture may also be low. When an application requests hardware access, it involves **four layers of mapping** which downgrades performance significantly. When the ISA of a guest OS is different from the ISA of the underlying hardware, binary translation must be adopted. Although the host-based architecture has flexibility, the performance is too low to be useful in practice.

Para-virtualization with compiler support

- Para-virtualization needs to modify the guest operating systems.
- A para-virtualized VM provides special APIs requiring substantial OS modifications in user applications.
- Performance degradation is a critical issue of a virtualized system. No one wants to use a VM if it is much slower than using a physical machine.
- The virtualization layer can be inserted at different positions in a machine software stack. However, para-virtualization attempts to reduce the virtualization overhead, and thus improve performance by modifying only the guest OS kernel

The guest operating systems are para-virtualized.
They are assisted by an intelligent compiler to replace the nonvirtualizable OS instructions by hypercalls as illustrated in Figure 3.8.
The traditional x86 processor offers four instruction execution rings: Rings 0, 1, 2, and 3.
The lower the ring number, the higher the privilege of instruction being executed. The OS is responsible for managing the hardware and the privileged instructions to execute at Ring 0, while user-level applications run at Ring 3. The best example of para-virtualization is the **KVM.**

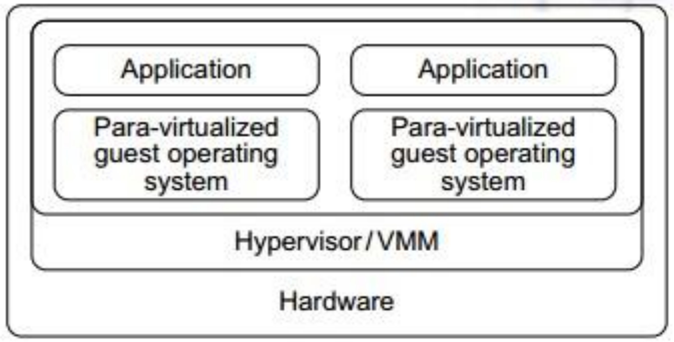

Application
Application
Para-virtualized guest operating system
Para-virtualized guest operating system
Hypervisor / VMM
Hardware

FIGURE 3.7

Para-virtualized VM architecture, which involves modifying the guest OS kernel to replace nonvirtualizable instructions with hypercalls for the hypervisor or the VMM to carry out the virtualization process (See Figure 3.8 for more details.)

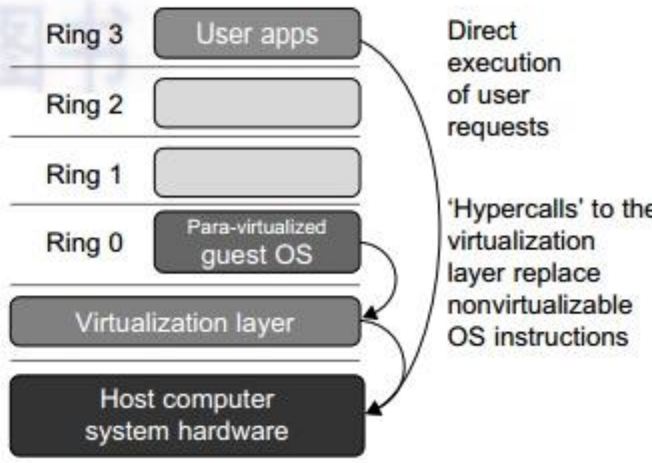

Ring 3
User apps
Ring 2
Ring 1
Ring 0
Para-virtualized guest OS
Virtualization layer
Host computer system hardware
Direct execution of user requests
'Hypercalls' to the virtualization layer replace nonvirtualizable OS instructions

FIGURE 3.8

The use of a para-virtualized guest OS assisted by an intelligent compiler to replace nonvirtualizable OS instructions by hypercalls.

(Courtesy of VMWare [71]

KVM (Kernel-Based VM)

- This is a Linux para-virtualization system—a part of the Linux version 2.6.20 kernel. Memory management and scheduling activities are carried out by the existing Linux kernel. The KVM does the rest, which makes it simpler than the hypervisor that controls the entire machine. KVM is a hardware-assisted para-virtualization tool, which improves performance and supports unmodified guest OSes such as Windows, Linux, Solaris, and other UNIX variants

Virtualization of CPU, Memory and I/O devices

1. Hardware support for Virtualization

To support multiple processes execution simultaneously modern os and processors support two mode of execution namely supervisor (privileged inst) mode and user mode.

This ensures controlled access of critical hardware.

Figure 3.10 provides an overview of Intel's full virtualization techniques. For processor virtualization, Intel offers the VT-x or VT-i technique. VT-x adds a privileged mode (VMX Root Mode) and some instructions to processors. This enhancement traps all sensitive instructions in the VMM automaticallv.

For memory virtualization, Intel offers the **EPT**, which translates the virtual address to the machine's physical addresses to improve performance. For I/O virtualization, Intel implements VT-d and VT-c to support this.

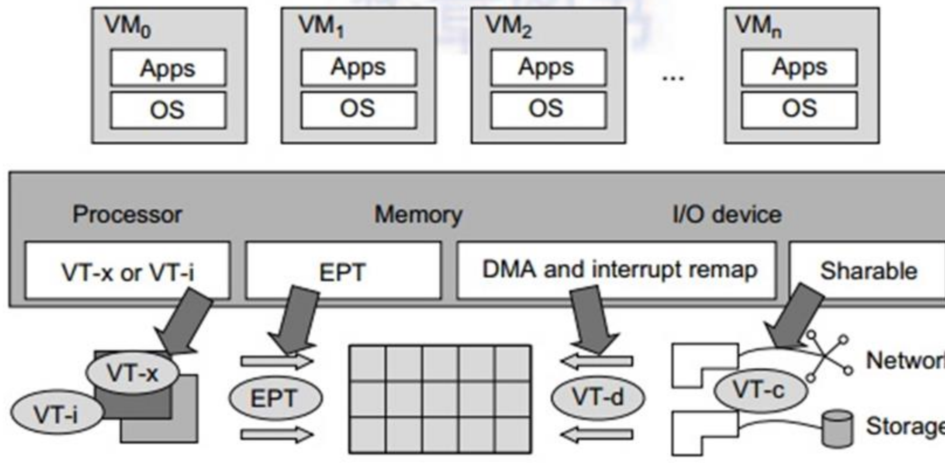


FIGURE 3.10

Intel hardware support for virtualization of processor, memory, and I/O devices.

CPU Virtualization

- **CPU virtualization** involves a single CPU acting as if it were multiple separate CPUs. The most common reason for doing this is to run multiple different operating systems on one machine. CPU virtualization emphasizes performance and runs directly on the available CPUs whenever possible. The underlying physical resources are used whenever possible and the virtualization layer runs instructions only as needed to make virtual machines operate as if they were running directly on a physical machine.
- Unprivileged instructions of VMs run directly on host machine
- Privileged or critical instructions are trapped into VMM where binary translations to host OS is taking place which is executed by hardware.
- Critical instructions are divided into three categories:
  - Privileged
  - control-sensitive- attempt to change the configuration of resources used
  - behaviour-sensitive- different behaviour depending on the configuration of resources including the load and store operations over the virtual memory.

Hardware-Assisted CPU Virtualization

- Intel and AMD add an additional mode called privilege mode level (some people call it Ring-1) to x86 processors.
- Therefore, operating systems can still run at Ring 0 and the hypervisor can run at Ring -1.
- All the privileged and sensitive instructions are trapped in the hypervisor automatically. This technique removes the difficulty of implementing binary translation of full virtualization. It also lets the operating system run in VMs without modification.
- Generally, hardware-assisted virtualization should have high efficiency. However, since the transition from the hypervisor to the guest OS incurs high overhead switches between processor modes, it sometimes cannot outperform binary translation. Hence, virtualization systems such as VMware now use a hybrid approach, in which a few tasks are offloaded to the hardware but the rest is still done in software. In addition, para-virtualization and hardware-assisted virtualization can be combined to improve the performance further

Memory Virtualization

- In a traditional execution environment, the operating system maintains mappings of virtual memory to machine memory using page tables, which is a one-stage mapping from virtual memory to machine memory.
- All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance.
- However, in a virtual execution environment, virtual memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs.
- Two-stage mapping process should be done by guest OS and the VMM.
- Virtual memory to physical memory and physical memory to machine memory.
- The guest OS cannot directly access the actual machine memory. The VMM is responsible for mapping the guest physical memory to the actual machine memory

Memory Virtualization cont..

Since each page table of the guest OS has a Separate page table in the VMM corresponding To it, the VMM page table is called **the shadow Page table**. Nested page tables add another layer of indirection to virtual memory. The MMU already handles virtual-to-physical translations as defined by the OS. Then the physical memory addresses are translated to machine addresses using another set of page tables defined by the hypervisor. Since modern operating systems maintain a set of page tables for every process, the shadow page tables will get flooded. Consequently, the performance overhead and cost of memory will be very high

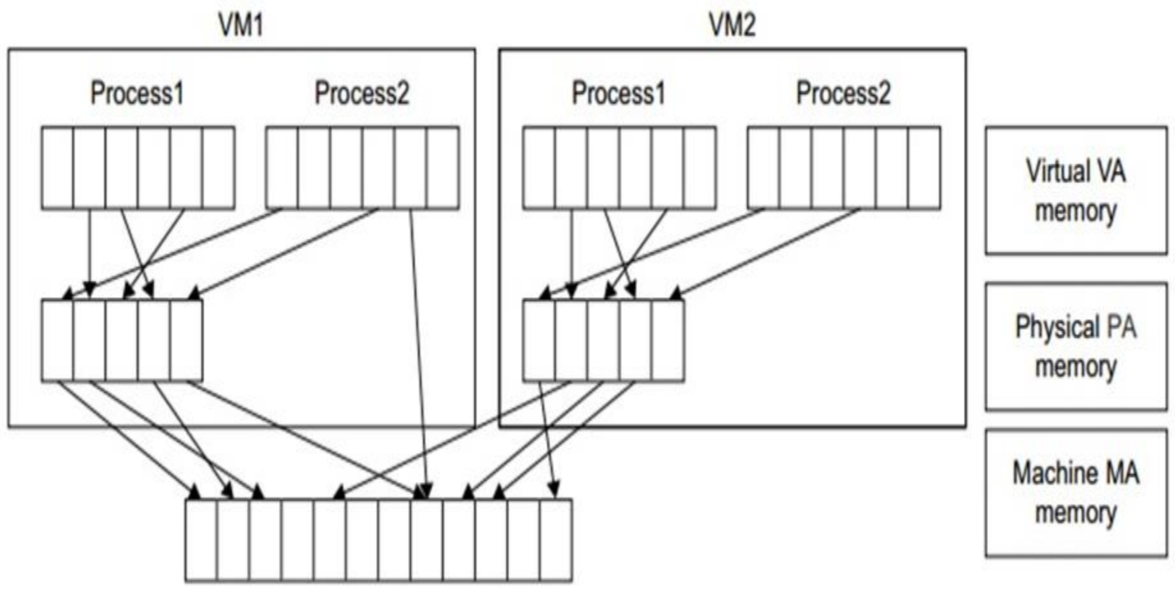


FIGURE 3.12

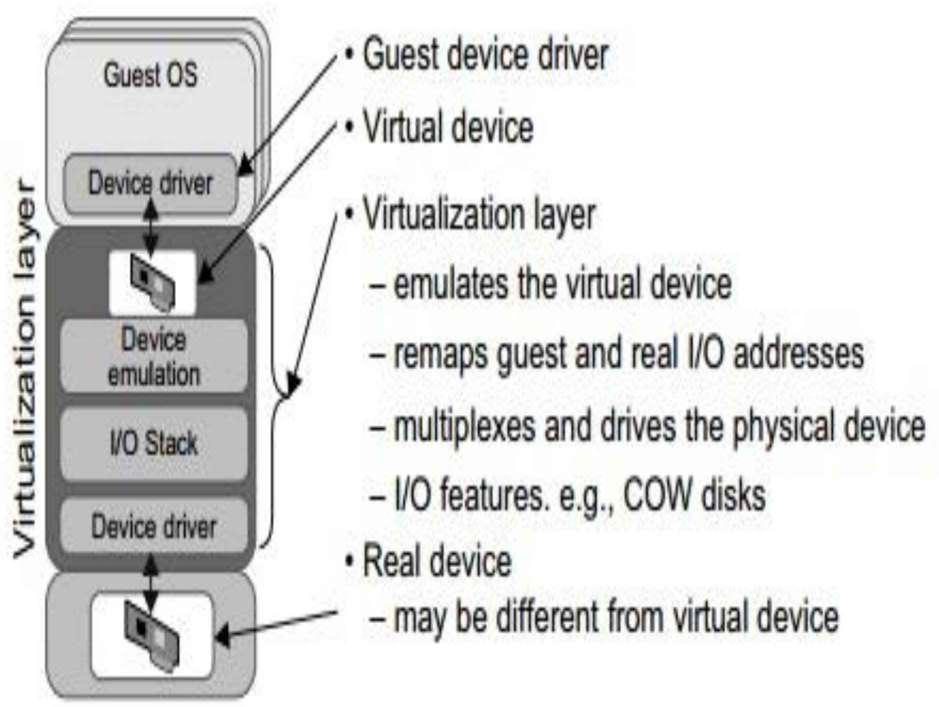
Two-level memory mapping procedure.

I/O Virtualization

- I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware.

Three ways of implementation

1. Full device emulation
2. Para-virtualization
3. Direct I/O



Guest OS
Device driver
Virtualization layer
Device emulation
I/O Stack
Device driver
• Guest device driver
• Virtual device
• Virtualization layer
– emulates the virtual device
– remaps guest and real I/O addresses
– multiplexes and drives the physical device
– I/O features. e.g., COW disks
• Real device
– may be different from virtual device

FIGURE 3.14

Device emulation for I/O virtualization implemented inside the middle layer that maps real I/O devices into the virtual devices for the guest device driver to use.

Full-virtualization- I/O

- All the functions of a device, bus infrastructure , identification, interrupts and DMA are replicated in the S/W.
- This S/W is located in the VMM and acts as a virtual device.
- The I/O access requests of the guest OS are trapped in the VMM which interacts with the I/O devices
- A single hardware device can be shared by multiple VMs that run concurrently

Para-virtualization – I/O

- Xen – follows this
- It is known as split driver model consisting of a front end driver and a backend driver.
- The frontend driver is running in Domain U and the backend driver is running in Domain 0
- They interact with each other via a block of shared memory.
- The frontend driver manages the I/O requests of the guest OSes and the backend driver is responsible for managing the real I/O devices and multiplexing the I/O data of different VMs.
- High CPU overhead is a draw back

Direct I/O virtualization

- Lets the VM access the devices directly.
- It can achieve close-to-native performance without high CPU costs.
- There are a lot of challenges for commodity hardware devices. For example, when a physical device is reclaimed (required by workload migration) for later reassignment, it may have been set to an arbitrary state (e.g., DMA to some arbitrary memory locations) that can function incorrectly or even crash the whole system.
- Intel VT-d supports the remapping of I/O DMA transfers and device-generated interrupts
- Another way is self-virtualized I/O (SV-IO). All tasks are encapsulated in this.
- It provides virtual devices and an associated access API to VMs and a management API to the VMM.
- SV-IO defines one virtual interface (VIF) for every kind of virtualized I/O device, such as virtual network interfaces, virtual block devices (disk), virtual camera devices, and others
- The guest OS interacts with the VIFs via VIF device drivers.
- Each VIF consists of two message queues.
- One is for outgoing messages to the devices and the other is for incoming messages from the devices. In addition, each VIF has a unique ID for identifying it in SV-IO.

VMware Workstation for I/O Virtualization

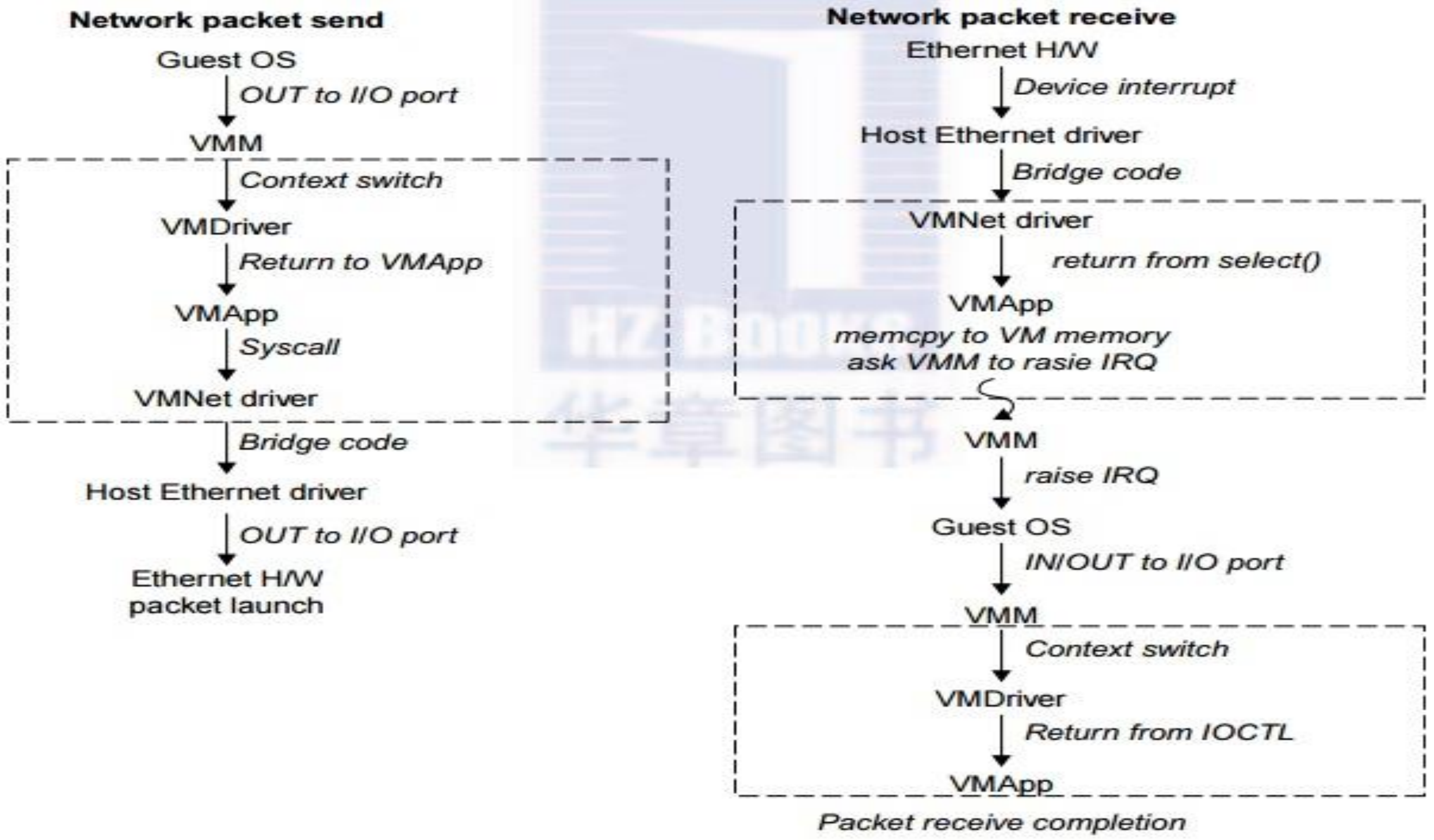


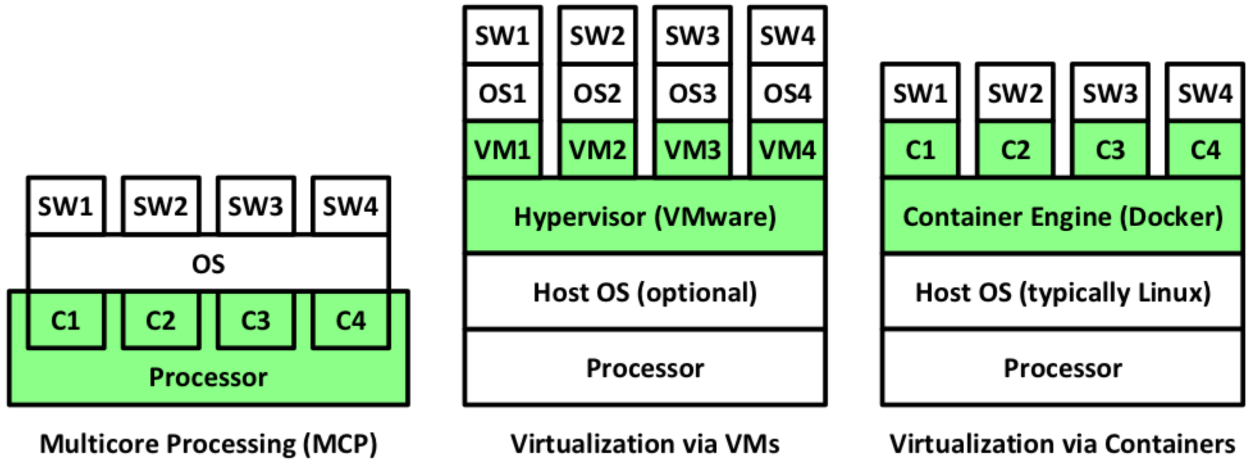
FIGURE 3.15

Functional blocks involved in sending and receiving network packets.

- The virtual NIC models an AMD Lance Am79C970A controller (Network Interface Controller)
- The device driver for a Lance controller in the guest OS initiates packet transmissions by reading and writing a sequence of virtual I/O ports; each read or write switches back to the VMApp to emulate the Lance port accesses.
- When the last OUT instruction of the sequence is encountered, the Lance emulator calls a normal write() to the VMNet driver.
- The VMNet driver then passes the packet onto the network via a host NIC and then the VMApp switches back to the VMM
- The switch raises a virtual interrupt to notify the guest device driver that the packet was sent. Packet receives occur in reverse.

Virtualization in Multi-Core Processors

- This is more complex than uni-core processor.
- Application programs must be parallelized to use all cores fully, and software must explicitly assign task to the cores.



Virtualization in Multi-Core Processors cont..

- Two difficulties

1) application programs must be parallelized to use all cores fully – new programming models, languages, and libraries are needed for this.

2) Software must explicitly assign tasks to the cores, which is very complex – improved scheduling algorithm and resource management policies.

The above can be solved by managing hardware with software. It is located under the ISA(instruction set Architecture) and remains unmodified by the OS or VMM(hypervisor)

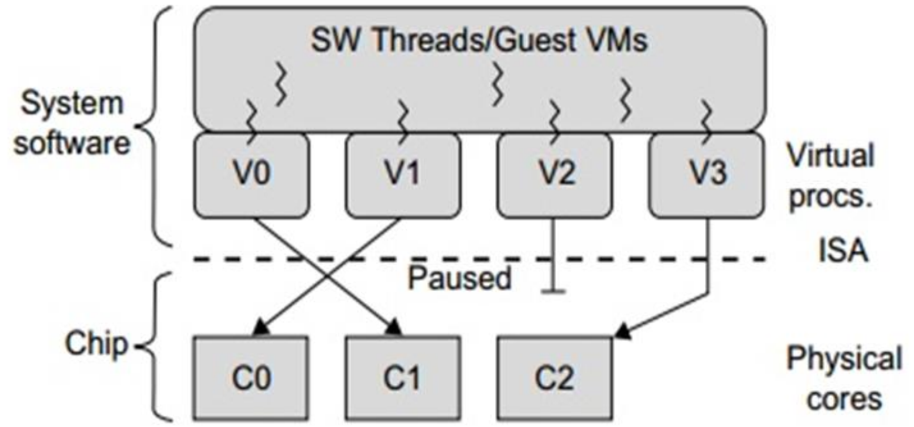


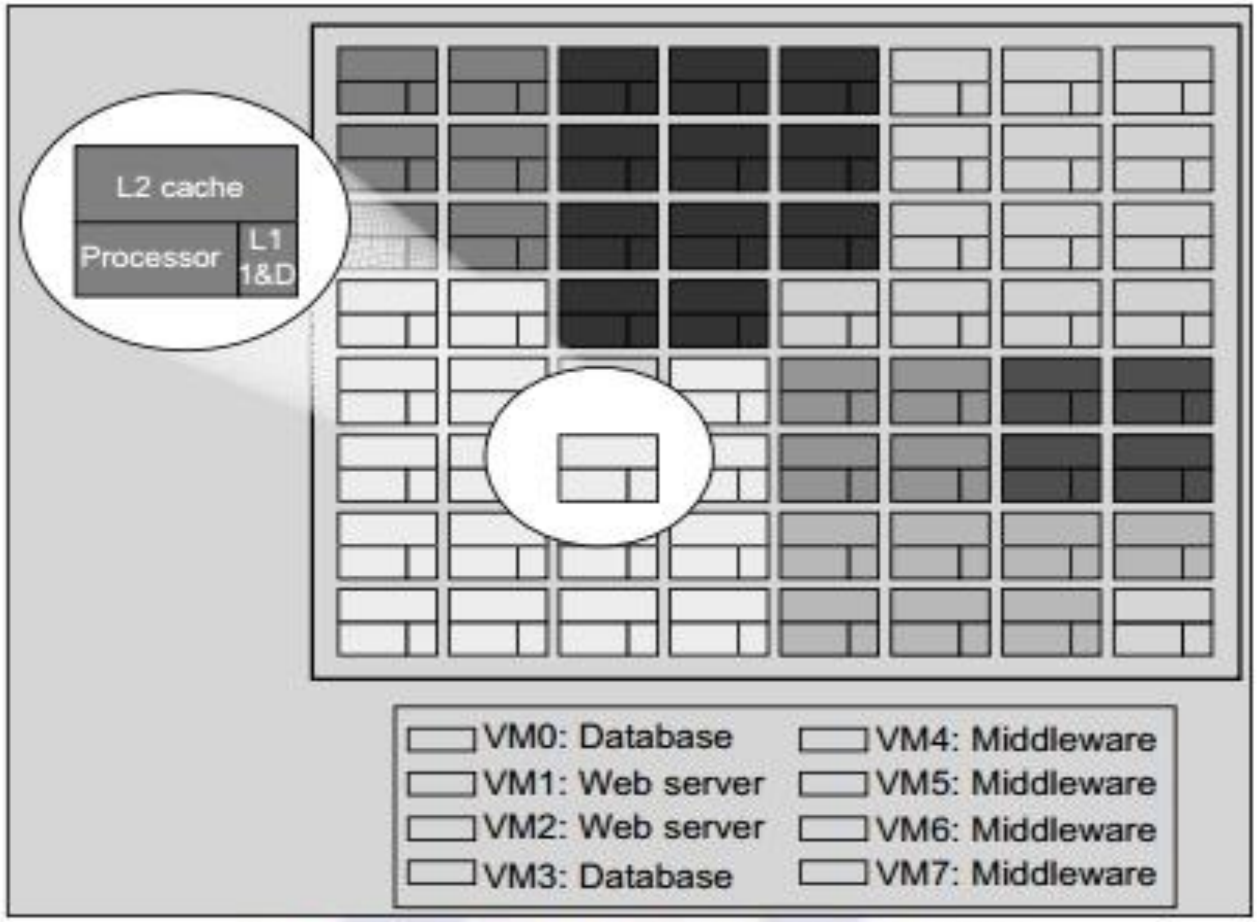
FIGURE 3.16

Multicore virtualization method that exposes four VCPUs to the software, when only three cores are actually present.

Virtual Hierarchy

- Instead of supporting time-sharing jobs on one or a few cores, we can use the abundant cores in a space-sharing, where single-threaded or multithreaded jobs are simultaneously assigned to separate groups of cores for long time intervals.
- To optimize for space-shared workloads, they propose using virtual hierarchies to overlay a coherence and caching hierarchy onto a physical processor. Unlike a fixed physical hierarchy, a virtual hierarchy can adapt to fit how the work is space shared for improved performance and performance isolation.
- A virtual hierarchy is a cache hierarchy that can adapt to fit the workload or mix of workloads

- Space sharing is applied to assign three workloads to three clusters of virtual cores: namely VM0 and VM3 for database workload, VM1 and VM2 for web server workload, and VM4–VM7 for middleware workload



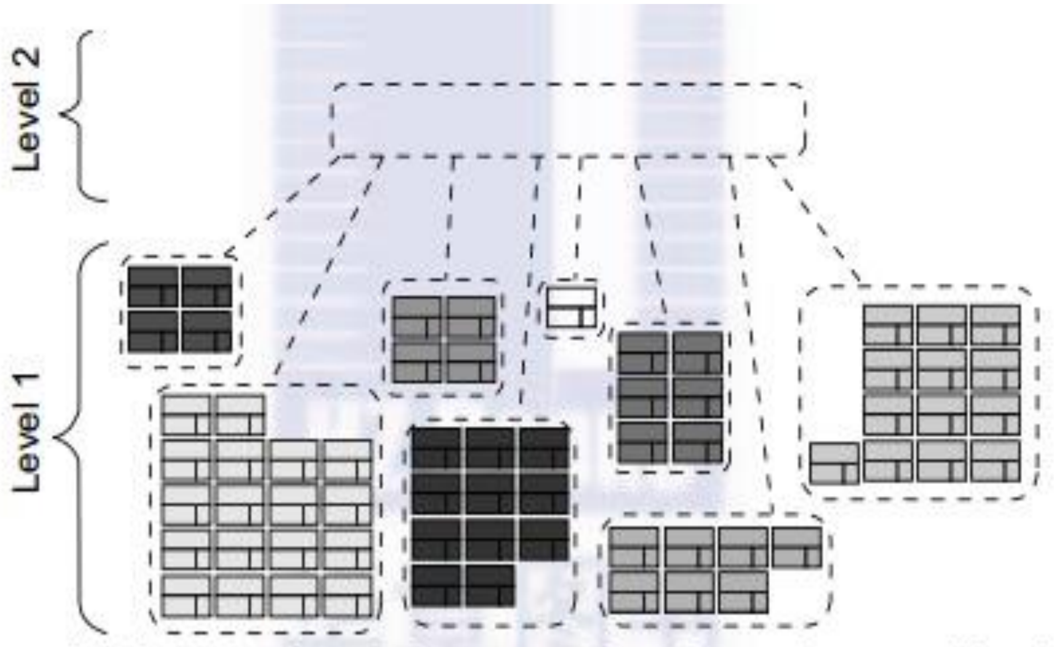
(a) Mapping of VMs into adjacent cores

Each VM operates in a isolated fashion at the first level

The second level maintains a globally shared memory. This facilitates dynamically repartitioning resources without costly cache flushes. .

Furthermore, maintaining globally shared memory minimizes changes to existing system software and allows virtualization features such as content-based page sharing.

A virtual hierarchy adapts to space-shared workloads like multiprogramming and server consolidation



(b) Multiple virtual clusters assigned to various workloads

FIGURE 3.17

CMP server consolidation by space-sharing of VMs into many cores forming multiple virtual clusters to execute various workloads.

VIRTUAL CLUSTERS AND RESOURCE MANAGEMENT

- A physical cluster is a collection of servers (physical machines) interconnected by a physical network such as a LAN
- Physical clusters versus virtual clusters
- Virtual clusters are built with VMs installed at distributed servers from one or more physical clusters. The VMs in a virtual cluster are interconnected logically by a virtual network across several physical networks

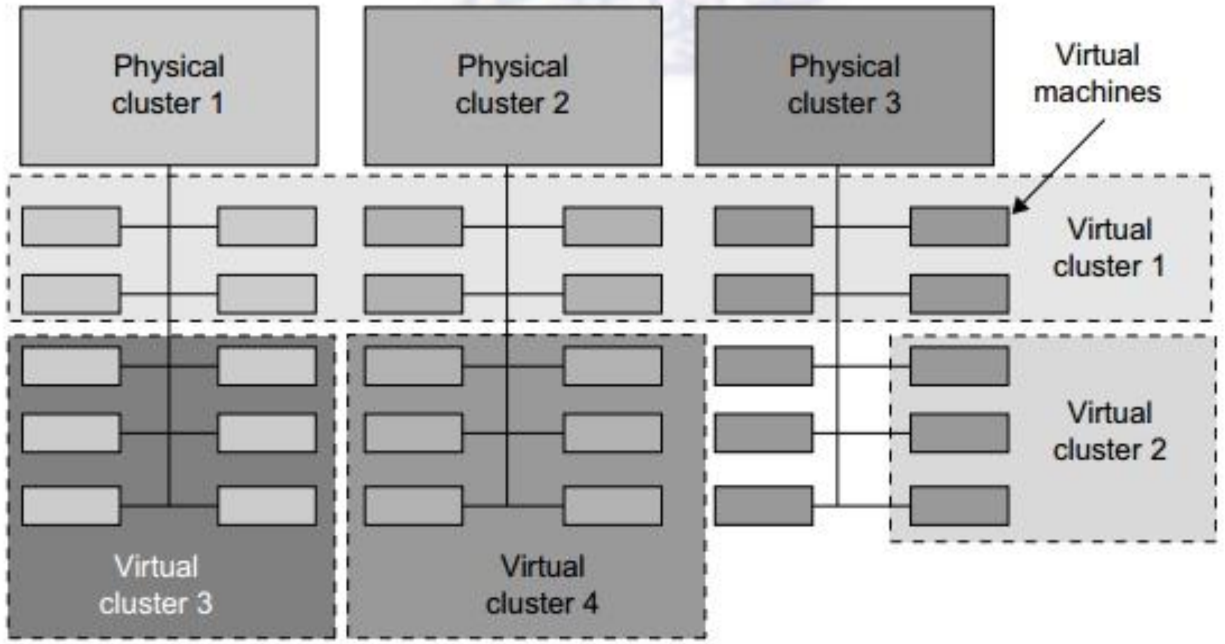


FIGURE 3.18

A cloud platform with four virtual clusters over three physical clusters shaded differently.

- The virtual cluster nodes can be either physical or virtual machines. Multiple VMs running with different OSes can be deployed on the same physical node.
- A VM runs with a guest OS, which is often different from the host OS, that manages the resources in the physical machine, where the VM is implemented.
- The purpose of using VMs is to consolidate multiple functionalities on the same server. This will greatly enhance server utilization and application flexibility.
- VMs can be colonized (replicated) in multiple servers for the purpose of promoting distributed parallelism, fault tolerance, and disaster recovery.
- The size (number of nodes) of a virtual cluster can grow or shrink dynamically, similar to the way an overlay network varies in size in a peer-to-peer (P2P) network.
- The failure of any physical nodes may disable some VMs installed on the failing nodes. But the failure of VMs will not pull down the host system.

It is necessary to effectively manage VMs which involves -virtual cluster deployment, monitoring and management over large-scale clusters, as well as resource scheduling, load balancing, server consolidation, fault tolerance, and other techniques

There are common installations for most users or applications, such as operating systems or user-level programming libraries. These software packages can be preinstalled as templates (called template VMs). With these templates, users can build their own software stacks. New OS instances can be copied from the template VM. User-specific components such as programming libraries and applications can be installed to those instances.

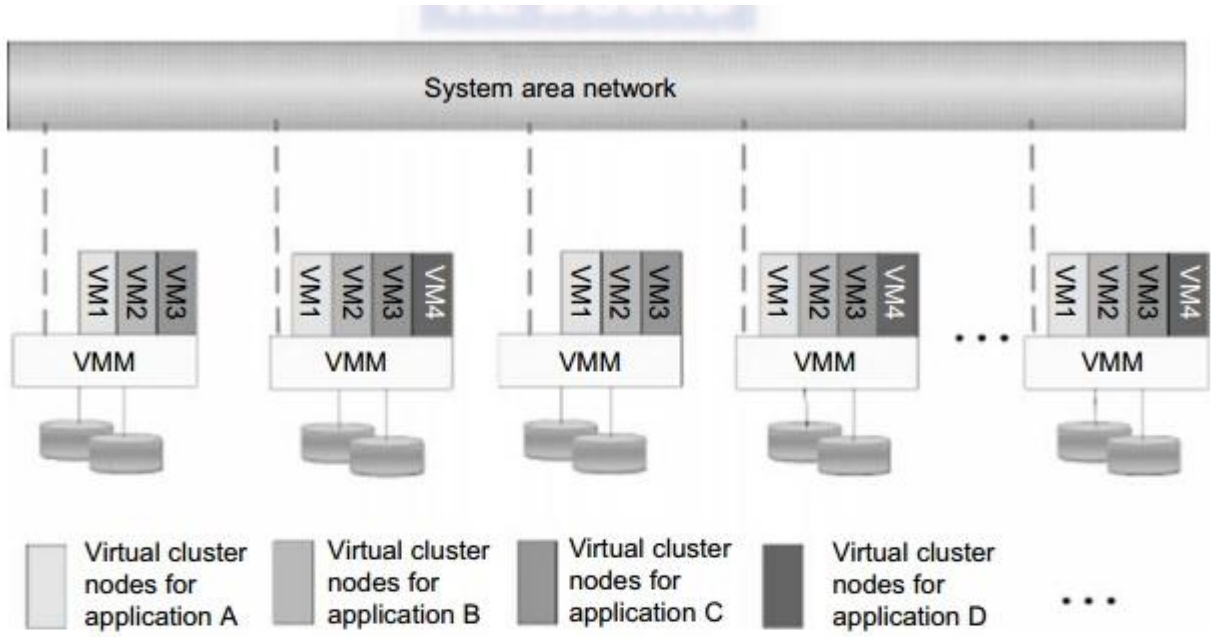

System area network
VM1
VM2
VM3
VM4
VMM
Virtual cluster nodes for application A
Virtual cluster nodes for application B
Virtual cluster nodes for application C
Virtual cluster nodes for application D

FIGURE 3.19

The concept of a virtual cluster based on application partitioning.

Fast Deployment and Effective Scheduling

- Deployment- construct and distribute the software stacks (OS, libraries, applications) to a physical node inside clusters as fast as possible and to quickly switch runtime environments from one user's virtual cluster to another user's virtual cluster.
- If one user finishes using his system, the corresponding virtual cluster should shut down or suspend quickly to save the resources to run other VMs for other users.
- Two advantages- live migration and load balancing
- Live Migration (advantage) - Live migration refers to the process of moving a running virtual machine or application between different physical machines without disconnecting the client or application. Memory, storage, and network connectivity of the virtual machine are transferred from the original host machine to the destination. When properly carried out, this process takes place without any noticeable effect from the point of view of the end user
- Live migration allows an administrator to take a virtual machine offline for maintenance or upgrading without subjecting the system's users to downtime.
- Requires a shared storage

- Load balancing (advantage)- **Load balancing can be achieved using the load index and frequency of user logins.**
- The automatic scale-up and scale-down mechanism of a virtual cluster can be implemented based on this model. Consequently, we can increase the resource utilization of nodes and shorten the response time of systems.
- Mapping VMs onto the most appropriate physical node should promote performance. Dynamically adjusting loads among nodes by live migration of VMs is desired, when the loads on cluster nodes become quite unbalanced.

High performance virtual storage

Four steps to deploy a group of VMs in a target cluster

1. Preparing disk image
2. Configuring the VMs
3. Choosing the destination nodes
4. Executing the VM deployment command on every host

- Many systems use **templates** to simplify the disk image preparation process.
- A **template is a disk image** that includes a preinstalled operating system with or without certain application software.
- Users choose a proper template according to their requirements and make a duplicate of it as their own disk image.
- Templates could implement the COW (Copy on Write) format.
- A new COW backup file is very small and easy to create and transfer. Therefore, it definitely reduces disk space consumption.
- In addition, VM deployment time is much shorter than that of copying the whole raw image file.

VM is configured with

1. Name
2. Disk image
3. Network setting
4. CPU allocation
5. Memory
6. IP address

- One needs to record each VM configuration into a file
- VMs with the same configurations could use pre-edited profiles to simplify the process.
- The deployment principle is to fulfil the VM requirement and to balance workloads among the whole host network.

Live VM migration steps and performance effects

- The potential drawback is that a VM must stop playing its role if its residing host node fails. However, this problem can be mitigated with VM life migration.
- Four ways to manage a virtual cluster.

1. Using a guest-based manager by which the cluster manager resides on a guest system. Example openMosix is a open source Linux cluster running different guest systems on top of Xen hypervisor. (guest only)
2. Build a cluster manager on host systems. The host-based manager supervises the guest systems and can restart the guest system on another physical machine. Eg Vmware HA system (host only)
3. use an independent cluster manager on both host and guest systems to manage virtual cluster.
4. Use an integrated cluster manager on the guest and host systems. In this the manager must be able to distinguish between virtualized and physical resources.

The motivation is to design a live VM migration scheme with negligible downtime, the lowest network bandwidth consumption possible, and a reasonable total migration time.

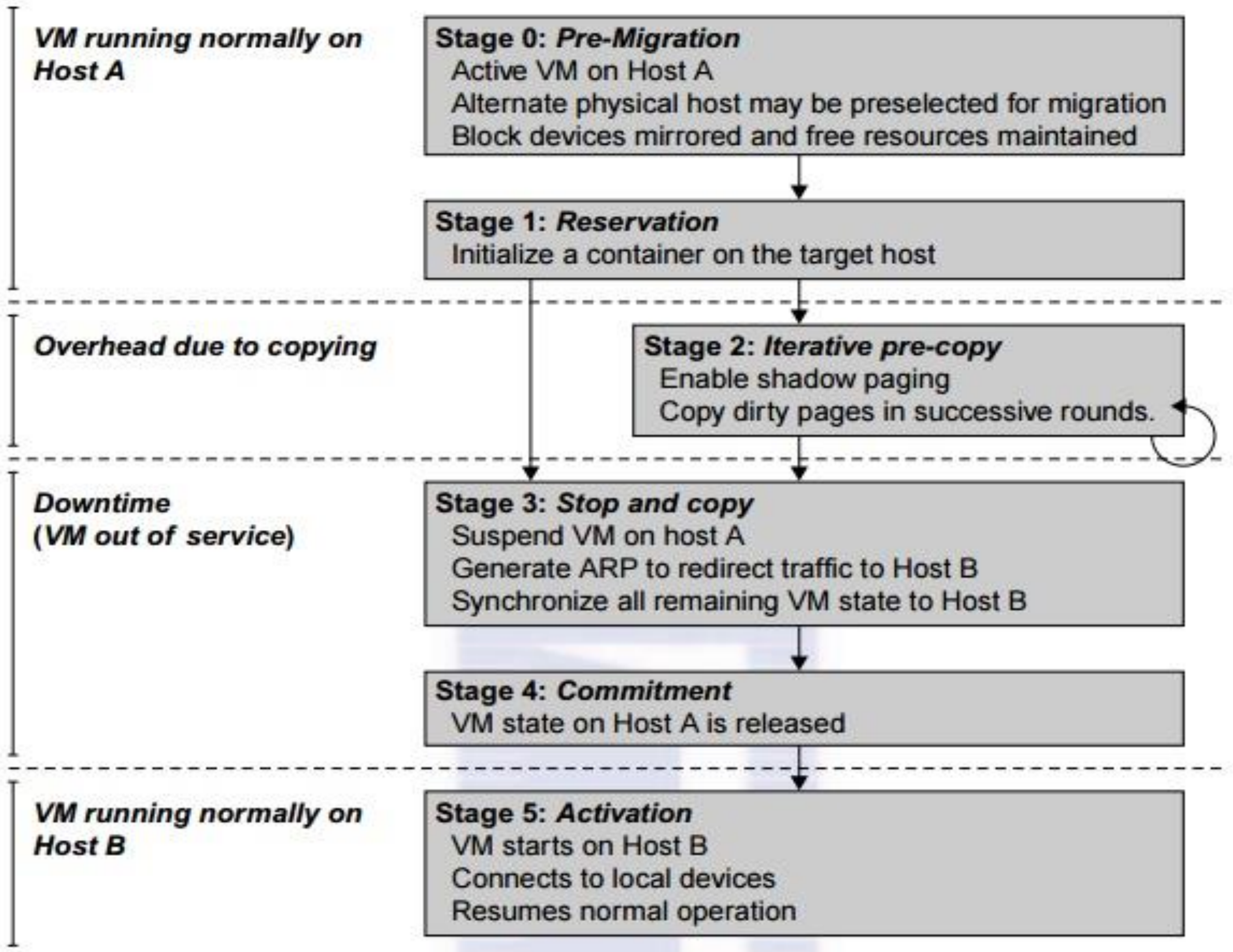
we should ensure that the migration will not disrupt other active services residing in the same host through resource contention (e.g., CPU, network bandwidth)

Live migration process

VM can be in any one of four states

1) Inactive state – VM is not enabled
2) Active state – VM instantiated at the virtualization platform
3) Paused state- VM instantiated but is disabled to process a task or waiting
4) Suspended state – if its machine fail and virtual resources are stored back to the disk

Dirty page: A logical write occurs when data is modified in a **page** in the buffer cache. A physical write occurs when the **page** is written from the buffer cache to disk. When a **page** is modified in the buffer cache, it is not immediately written back to disk; instead, the **page** is marked as **dirty**.



VM running normally on Host A
Stage 0: Pre-Migration
Active VM on Host A
Alternate physical host may be preselected for migration
Block devices mirrored and free resources maintained
Stage 1: Reservation
Initialize a container on the target host
Overhead due to copying
Stage 2: Iterative pre-copy
Enable shadow paging
Copy dirty pages in successive rounds.
Downtime (VM out of service)
Stage 3: Stop and copy
Suspend VM on host A
Generate ARP to redirect traffic to Host B
Synchronize all remaining VM state to Host B
Stage 4: Commitment
VM state on Host A is released
VM running normally on Host B
Stage 5: Activation
VM starts on Host B
Connects to local devices
Resumes normal operation

FIGURE 3.20

Live migration process of a VM from one host to another.

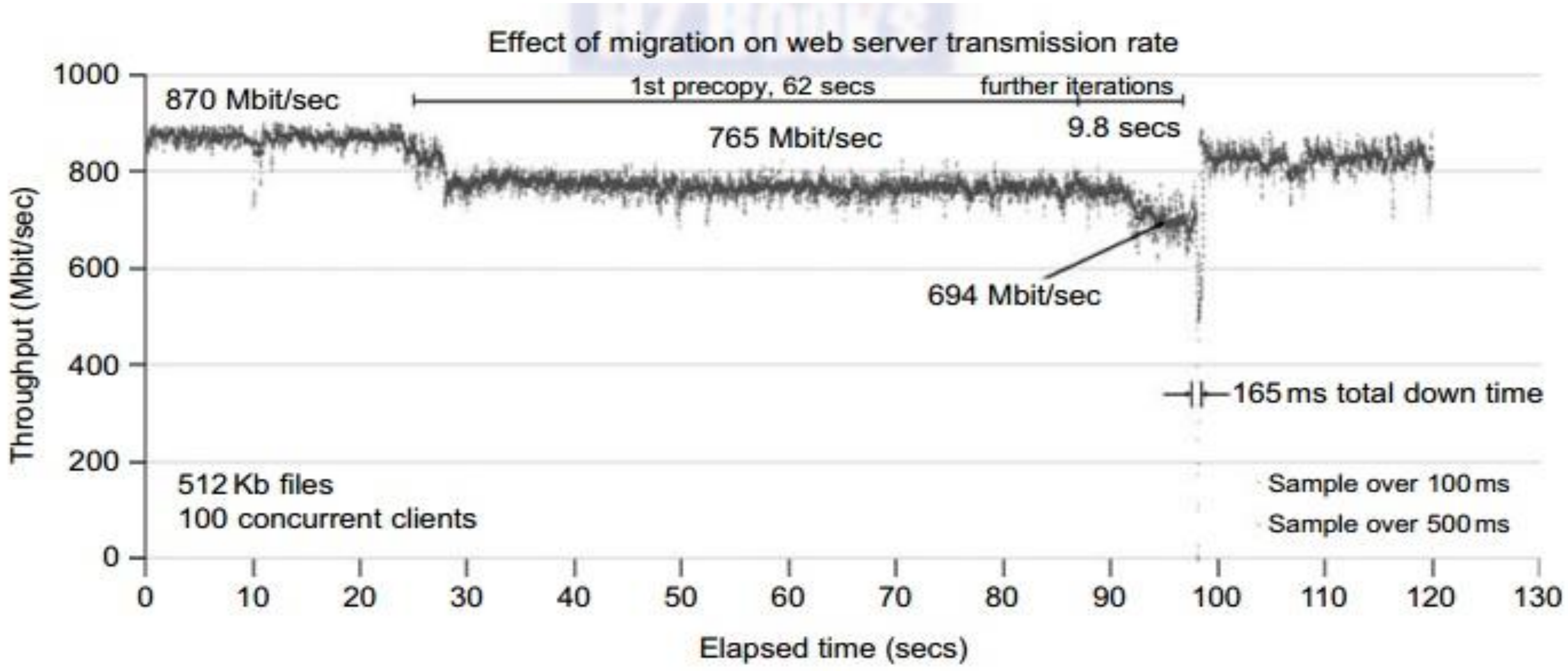


FIGURE 3.21

Effect on data transmission rate of a VM migrated from one failing web server to another.

Migration of memory, files and network resources

- Clusters have a high initial cost of ownership, including space, power conditioning, and cooling equipment, leasing or sharing access to a common cluster is an attractive solution when demands vary over time.
- Shared clusters offer economies of scale and more effective utilization of resources by multiplexing.

Memory migration

- Moving the memory instance of a VM from one physical host to another can be approached in any number of ways
- The Internet Suspend-Resume (ISR) technique exploits **temporal locality** as memory states are likely to have considerable overlap in the suspended and the resumed instances of a VM.
- Temporal locality refers to the fact that the memory states differ only by the amount of work done since a VM was last suspended before being initiated for migration.
- To exploit temporal locality, each file in the file system is represented as a tree of small subfiles. A copy of this tree exists in both the suspended and resumed VM instances.
- The advantage of using a tree-based representation of files is that the caching ensures the transmission of only those files which have been changed

Migration of memory, files and network resources

File system migration

- To support VM migration, a system must provide each VM with a consistent, location-independent view of the file system that is available on all hosts.
- A simple way to achieve this is to provide each VM with its own virtual disk which the file system is mapped to and transport the contents of this virtual disk along with the other states of the VM.
- However, due to the current trend of high-capacity disks, migration of the contents of an entire disk over a network is not a viable solution.
- Another way is to have a global file system across all machines where a VM could be located. This way removes the need to copy files from one machine to another because all files are network-accessible.

File system migration - cont

- A distributed file system is used in ISR serving as a transport mechanism for propagating a suspended VM state.
- The actual file systems themselves are not mapped onto the distributed file system. Instead, the VMM only accesses its local file system.
- The relevant VM files are explicitly copied into the local file system for a resume operation and taken out of the local file system for a suspend operation.
- This approach relieves developers from the complexities of implementing several different file system calls for different distributed file systems.
- It also essentially disassociates the VMM from any particular distributed file system semantics. However, this decoupling means that the VMM has to store the contents of each VM's virtual disks in its local files, which have to be moved around with the other state information of that VM.

File system migration - cont

- In smart copying, the VMM exploits spatial locality.
- Typically, people often move between the same small number of locations, such as their home and office.
- In these conditions, it is possible to transmit only the difference between the two file systems at suspending and resuming locations.
- This technique significantly reduces the amount of actual physical data that has to be moved.
- In situations where there is no locality to exploit, a different approach is to synthesize much of the state at the resuming site.
- On many systems, user files only form a small fraction of the actual data on disk.
- Operating system and application software account for the majority of storage space

Network migration

- To enable remote systems to locate and communicate with a VM, each VM must be assigned a virtual IP address known to other entities.
- This address can be distinct from the IP address of the host machine where the VM is currently located.
- Each VM can also have its own distinct virtual MAC address. (M**edia Access Control address)**
- The VMM maintains a mapping of the virtual IP and MAC addresses to their corresponding VMs. In general, a migrating VM includes all the protocol states and carries its IP address with it.
- If the source and destination machines of a VM migration are typically connected to a single switched LAN, an unsolicited ARP (Address Resolution Protocol - is a protocol for mapping an Internet Protocol address (IP address) to a physical machine address (MAC) that is recognized in the local network) reply from the migrating host is provided advertising that the IP has moved to a new location.
- This solves the open network connection problem by reconfiguring all the peers to send future packets to a new location.
- Although a few packets that have already been transmitted might be lost, there are no other problems with this mechanism.
- Alternatively, on a switched network, the migrating OS can keep its original Ethernet MAC address and rely on the network switch to detect its move to a new port.

Network migration- cont..

- In live migration Only memory and CPU status needs to be transferred from the source node to the target node.
- Live migration techniques mainly use the <u>pre-copy approach</u>, which first transfers all memory pages, and then only copies modified pages during the last round iteratively.
- The VM service downtime is expected to be minimal by using iterative copy operations.
- When applications' writable working set becomes small, the VM is suspended and only the CPU state and dirty pages in the last round are sent out to the destination.
- In the pre-copy phase, much performance degradation will occur because the migration daemon continually <u>consumes network bandwidth to transfer dirty pages in each round.</u>
- An adaptive rate limiting approach is employed to mitigate this issue, but total migration time is prolonged by nearly 10 times. Moreover, the maximum number of iterations must be set because not all applications' dirty pages are ensured to converge to a small writable working set over multiple rounds.

Network migration- cont..

- In fact, these issues with the pre-copy approach are caused by the large amount of transferred data during the whole migration process.(memory-to-memory)
- A checkpointing/recovery and trace/replay approach (CR/ TR-Motion) is proposed to provide fast VM migration.
- This approach transfers the execution trace file in iterations rather than dirty pages, which is logged by a trace daemon.
- Apparently, the total size of all log files is much less than that of dirty pages.
- So, total migration time and downtime of migration are drastically reduced.
- However, CR/TR-Motion is valid only when the log replay rate is larger than the log growth rate. The inequality between source and target nodes limits the application scope of live migration in clusters.

Network migration- cont..

- Another strategy is – post-copy
- In this all memory pages are transferred only once during the whole migration process and the baseline total migration time is reduced.
- But the downtime is much higher than that of pre-copy due to the latency of fetching pages from the source node before the VM can be resumed on the target.
- With the advent of multicore or many-core machines, abundant CPU resources are available.
- Even if several VMs reside on a same multicore machine, CPU resources are still rich because physical CPUs are frequently amenable to multi-plexing.
- We can exploit these copious CPU resources to compress page frames and the amount of transferred data can be significantly reduced.
- Memory compression algorithms typically have little memory overhead.
- Decompression is simple and very fast and requires no memory for decompression.

Virtualization for data-center automation

- Many big companies like google, apple, amazon etc invested billions of dollars in data-center construction and automation. The aim is allocation of resources dynamically to all internet users with guaranteed QoS and cost effectiveness.
- This automation is possible only with virtualization.
- The virtualization highlights:
- High availability
- Backup services
- Workload balancing
- Increases in client bases(no of users)
- Automation
- Service orientation

Server consolidation in data centers

- Data centers- large number of heterogeneous workloads run on servers
- This can be divided as – chatty workloads and non-interactive workloads
- Chatty workloads- may burst at some point and may return silent state. Example –web video service- heavy use in night time and less use at day.
- Non-interactive workloads-do not require people's efforts to make progress after they are submitted. Example- high-performance computing
- It is common that most servers in data centers are underutilized. A large amount of hardware, space, power, and management cost of these servers is wasted.
- Server consolidation is an approach to improve the low utility ratio of hardware resources by reducing the number of physical servers.
- Server virtualization enables smaller resource allocation than a physical machine.

Server virtualization has the following side effects

- Consolidation enhances hardware utilization. Many underutilized servers are consolidated into fewer servers to enhance resource utilization. Consolidation also facilitates backup services and disaster recovery.
- This approach enables more agile provisioning and deployment of resources. In a virtual environment, the images of the guest OSes and their applications are readily cloned and reused.
- The total cost of ownership is reduced. In this sense, server virtualization causes deferred purchases of new servers, a smaller data-center footprint, lower maintenance costs, and lower power, cooling, and cabling requirements
- This approach improves availability and business continuity. The crash of a guest OS has no effect on the host OS or any other guest OS. It becomes easier to transfer a VM from one server to another, because virtual servers are unaware of the underlying hardware

- To automate data center operations the following has to be considered
- Resource scheduling
- architectural support
- power management
- Performance of analytical models
- Automatic resource management, etc
- Dynamic CPU allocation
    - One method considers both CPU and memory flowing as well as automatically adjusting resource over-head based on varying workloads in hosted services.
    - In another method a local controller at the VM level and a global controller at the server level are designed. They implement autonomic resource allocation via the interaction of the local and global controllers
- Scheduling and reallocations has to be done carefully by fine scheduler.

Virtual storage management

- Generally, the data stored in this environment can be classified into two categories:
- VM images and application data.
- The VM images are special to the virtual environment, while application data includes all other data which is the same as the data in traditional OS environments
- Important aspect of system virtualization are **encapsulation and isolation.**
- Isolation: while VM can share the physical resources of a physical machine, they remain completely isolated from each other as if they were separate physical machines.
- Encapsulation: this makes virtual machine portable and easy to manage. For eg a VM can be moved and copied from one location to another just like a file
- The storage management of guest OS performs as though it is operating in a real hard disk while the guest OSes cannot access the hard disk directly. On the other hand, many guest OSes contest the hard disk when many VMs are running on a single physical machine. Therefore, storage management of the underlying VMM is much more complex than that of guest OSes

- Parallax is a distributed storage system customized for virtualization environments.
- Parallax runs as a user-level application in the storage appliance VM.
- It provides virtual disk images (VDIs) to VMs.
- A VDI is a single-writer virtual disk which may be accessed in a location-transparent manner from any of the physical hosts in the Parallax cluster. The VDIs are the core abstraction provided by Parallax.

- Cloud OS for Virtualized data centers
- In the table Nimbus, Eucalyptus,OpenNebula are open sources.

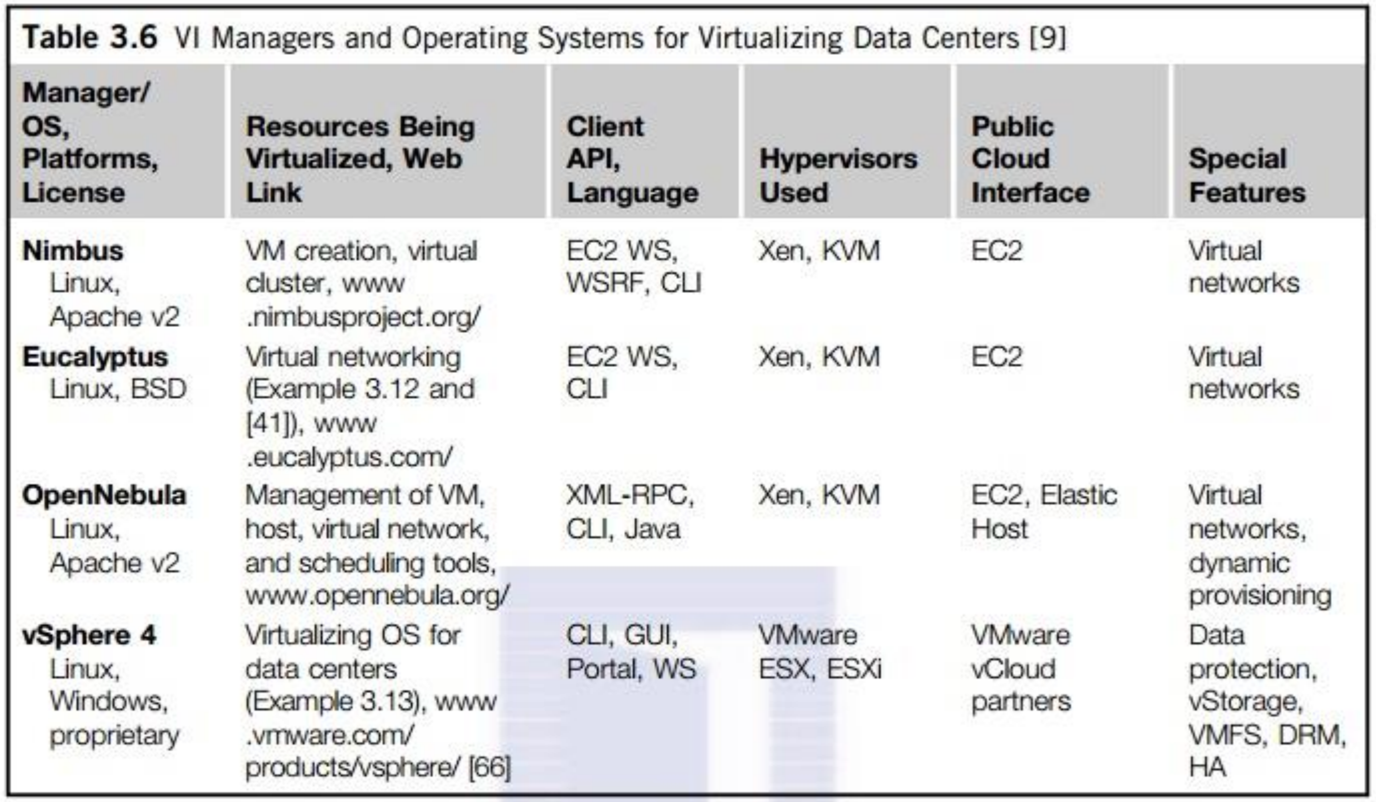


Table 3.6 VI Managers and Operating Systems for Virtualizing Data Centers [9]

Manager/ OS, Platforms, License	Resources Being Virtualized, Web Link	Client API, Language	Hypervisors Used	Public Cloud Interface	Special Features
Nimbus Linux, Apache v2	VM creation, virtual cluster, www .nimbusproject.org/	EC2 WS, WSRF, CLI	Xen, KVM	EC2	Virtual networks
Eucalyptus Linux, BSD	Virtual networking (Example 3.12 and [41]), www .eucalyptus.com/	EC2 WS, CLI	Xen, KVM	EC2	Virtual networks
OpenNebula Linux, Apache v2	Management of VM, host, virtual network, and scheduling tools, www.opennebula.org/	XML-RPC, CLI, Java	Xen, KVM	EC2, Elastic Host	Virtual networks, dynamic provisioning
vSphere 4 Linux, Windows, proprietary	Virtualizing OS for data centers (Example 3.13), www .vmware.com/ products/vsphere/ [66]	CLI, GUI, Portal, WS	VMware ESX, ESXi	VMware vCloud partners	Data protection, vStorage, VMFS, DRM, HA

Eucalyptus for Virtual Networking of Private Cloud

- It is an open source s/w system mainly for supporting IaaS clouds.
- Supports virtual networking and mgt of VMs. It doesn’t support virtual storage
- It supports private cloud and interaction with public clouds.
- High-level system components are designed as stand-alone webservices.
- Three resource managers in the design are

Instance manager- controls the execution, inspection, and terminating of VM instances on the host where it runs.
Group manager- gathers information about and schedules VM execution on specific instance mgrs., as well as manages virtual instance network
Cloud manager- is the entry-point into the cloud for users and administrators.
It queries node managers for information about resources, makes scheduling decisions, and implements them by making requests to group managers

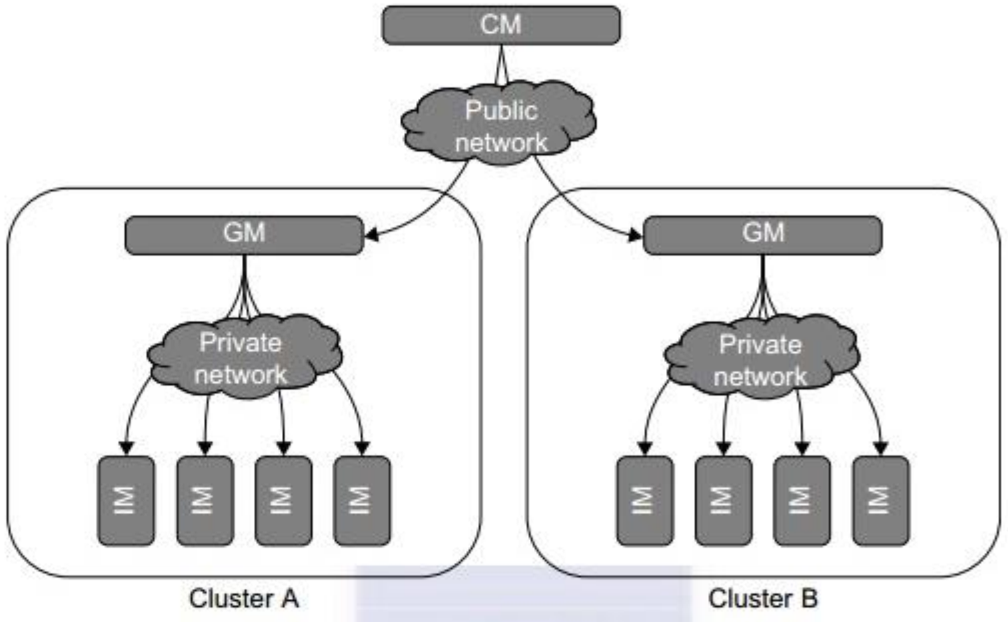


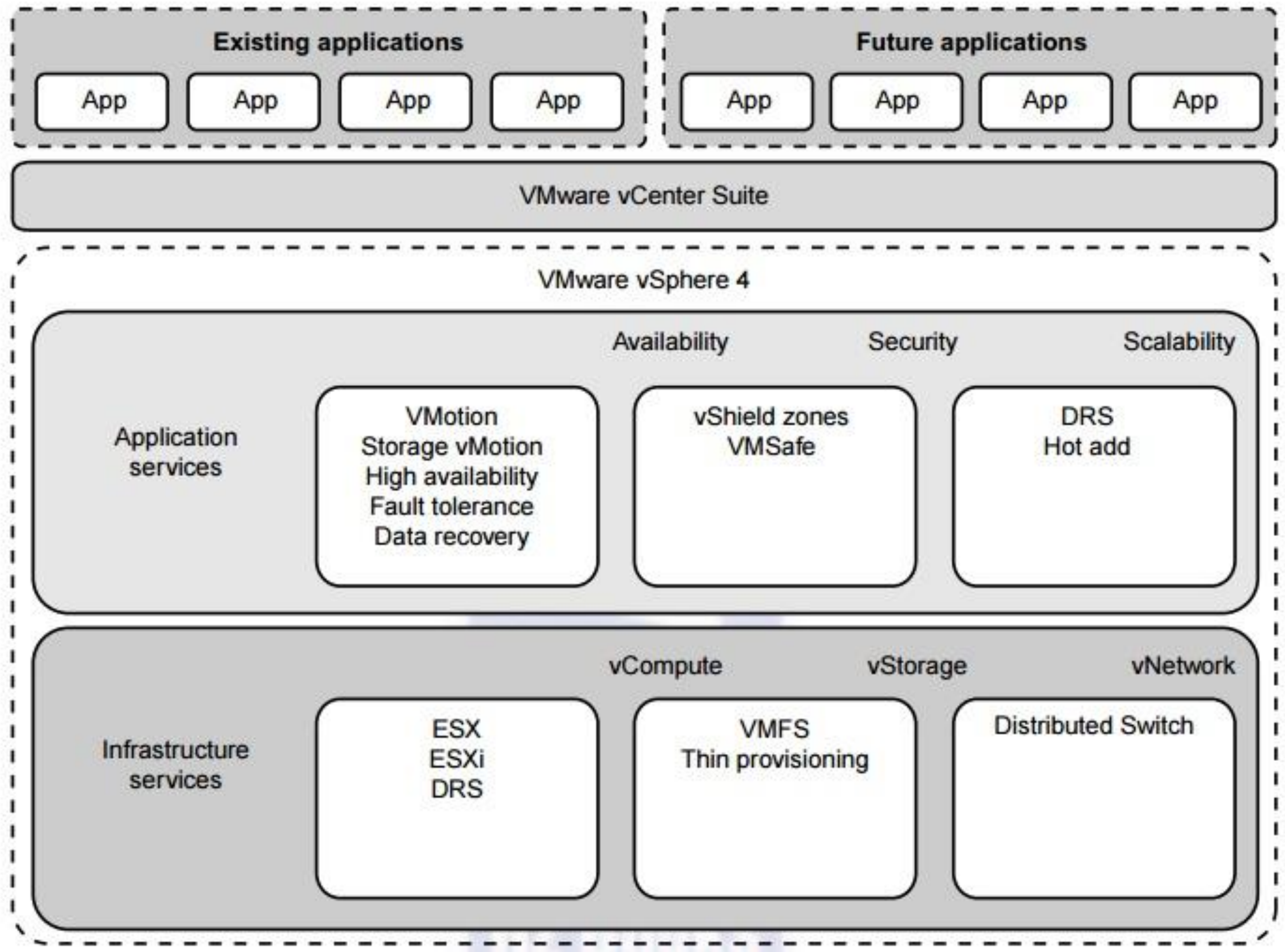
FIGURE 3.27

Eucalyptus for building private clouds by establishing virtual networks over the VMs linking through Ethernet and the Internet.

VMware vSphere 4 as a Commercial Cloud OS

- The vSphere 4 offers a hardware and software ecosystem developed by VMware and released in April 2009.
- vSphere extends earlier virtualization software products by VMware, namely the VMware Workstation, ESX for server virtualization, and Virtual Infrastructure for server clusters

vSphere4 is built with two functional s/w suites: infrastructure services and Application services
Infrastructure services:
vCompute- supported by ESX, ESXi, DRS
vStorage- supported by VMS and thin pro-
Visioning libraries
vNetwork- offers distributed switching.
These packages interact with h/w servers,
Disks and networks in data center.
Application services:
Availability- Vmotion, StoragevMotion etc
Security- vShield, Vmsafe etc
Scalability- DRS, Hot Add



Trust management in virtualized data centers

- A VM entirely encapsulates the state of the guest operating system running inside it.
- Encapsulated machine state can be copied and shared over the network and removed like a normal file, which proposes a challenge to VM security.
- In general, a VMM can provide secure isolation and a VM accesses hard-ware resources through the control of the VMM, so the VMM is the base of the security of a virtual system.
- Once a hacker successfully enters the VMM or management VM, the whole system is in danger.

VM-based Intrusion Detection

- Intrusions are unauthorized access to a certain computer from local or network users and intrusion detection is used to recognize the unauthorized access.
- An intrusion detection system (IDS) is built on operating systems, and is based on the characteristics of intrusion actions.
- A typical IDS can be classified as a host-based IDS (HIDS) or a network-based IDS (NIDS), depending on the data source.
- A HIDS can be implemented on the monitored system. When the monitored system is attacked by hackers, the HIDS also faces the risk of being attacked.
- A NIDS is based on the flow of network traffic which can't detect fake actions.
- The VM-based IDS contains a policy engine and a policy module. The policy framework can monitor events in different guest VMs by operating system interface library and PTrace indicates trace to secure policy of monitored host

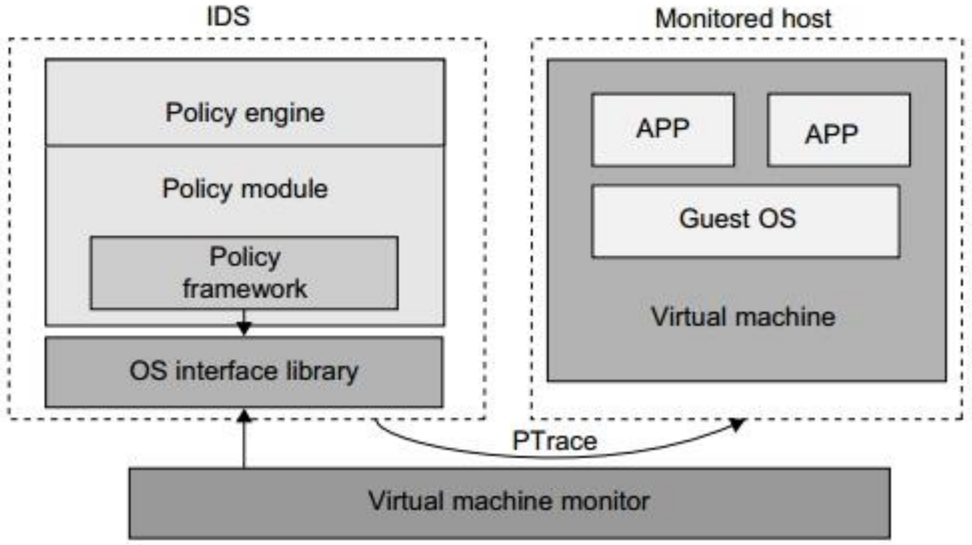


FIGURE 3.29

The architecture of livewire for intrusion detection using a dedicated VM.

- Besides IDS, honeypots and honeynets are also prevalent in intrusion detection.
- They attract and provide a fake system view to attackers in order to protect the real system.
- In addition, the attack action can be analysed, and a secure IDS can be built.
- A honeypot is a purposely defective system that simulates an operating system to cheat and monitor the actions of an attacker
- The arrowed boxes on the left and the brief description between the arrows and the zoning boxes are security functions and actions taken at the four levels from the users to the providers.
- The small circles between the four boxes refer to interactions between users and providers and among the users themselves.
- The arrowed boxes on the right are those functions and actions applied between the tenant environments, the provider, and the global communities..

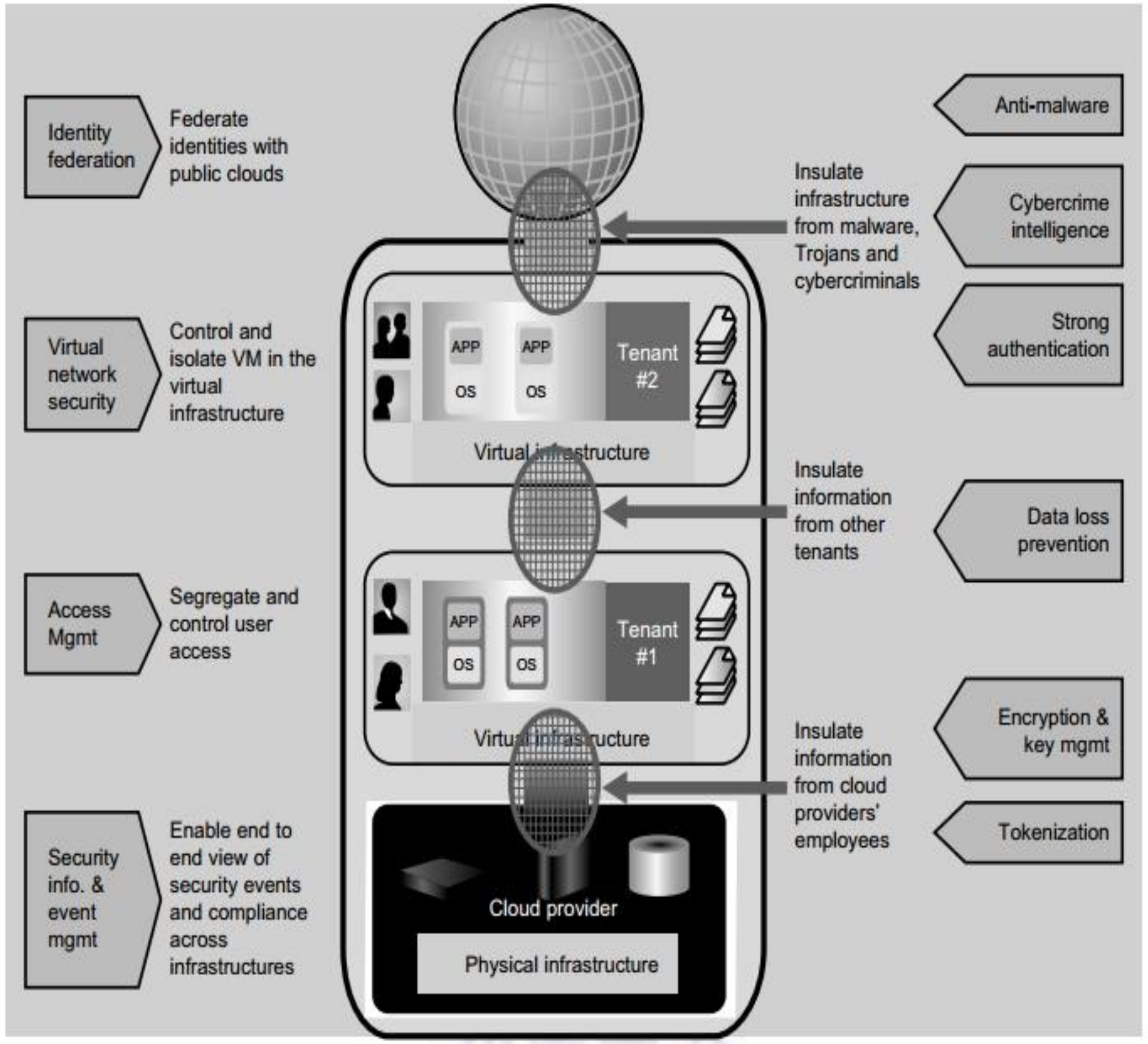

Identity federation
Federate identities with public clouds
Virtual network security
Control and isolate VM in the virtual infrastructure
Access Mgmt
Segregate and control user access
Security info. & event mgmt
Enable end to end view of security events and compliance across infrastructures
APP
OS
Tenant #2
Virtual infrastructure
Tenant #1
Cloud provider
Physical infrastructure
Insulate infrastructure from malware, Trojans and cybercriminals
Insulate information from other tenants
Insulate information from cloud providers' employees
Anti-malware
Cybercrime intelligence
Strong authentication
Data loss prevention
Encryption & key mgmt
Tokenization

FIGURE 3.30

Techniques for establishing trusted zones for virtual cluster insulation and VM isolation.

Why we need virtualization

- Save energy, go green
- Help move things to cloud
- Snapshots
- Live migration
- Easier management from single console
- Easier high availability(HA) deployment

Virtualization features

Snapshots:

- A storage snapshot is a set of refence markets or pointers to data stored on a disk drive, on a tape, or in a storage area network(SAN)
- Snapshot is something like a detailed table of contents, but it is treated by the computer as a complete data backup.
- creates a snapshot or image of a set of files, an entire data repository, system or drive. This image can be stored on a backup medium with minimum overheads

Two types of snapshot:

- Copy-on-write:-The copy-on-write snapshot stores only the metadata about where the original data is located, but doesn't copy the actual data at the initial creation. This makes snapshot creation virtually instantaneous, with little impact on the system taking the snapshot. The snapshot then tracks the original volume paying attention to changed blocks as writes are performed. As the blocks change, the original data is copied into the reserved storage capacity set aside for the snapshot prior to the original data being overwritten. The original data blocks snapped are copied just once at the first write request. This process ensures snapshot data is consistent with the exact time the snapshot was taken, and it's why the process is called "copy-on-write."
- Split-mirror - A clone or split-mirror snapshot creates an identical copy of the data. It requires as much storage capacity as the original data

Virtualization features -continued

- High Availability

High Availability refers to a system or component that is continuously operational for a desirably long length of time.

HA clusters or **failover clusters** are groups of computers that support server applications that can be reliably utilized with a minimum of down-time. They operate by using high availability software to harness redundant computers in groups or clusters that provide continued service

Why we need virtualization

- Save energy, go green
- Help move things to cloud
- Snapshots
- Live migration
- Easier management from single console
- Easier high availability(HA) deployment

Virtualization features

Snapshots:

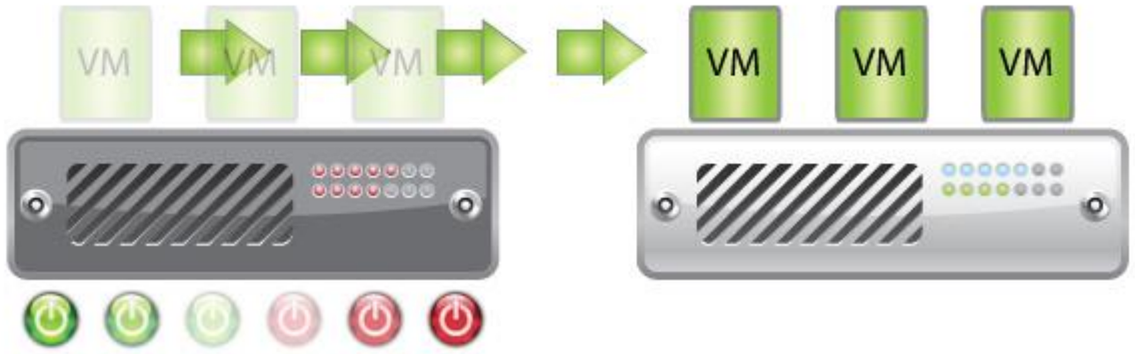
- A storage snapshot is a set of refence markets or pointers to data stored on a disk drive, on a tape, or in a storage area network(SAN)
- Snapshot is something like a detailed table of contents, but it is treated by the computer as a complete data backup.
- creates a snapshot or image of a set of files, an entire data repository, system or drive. This image can be stored on a backup medium with minimum overheads

Two types of snapshot:

- Copy-on-write:-The copy-on-write snapshot stores only the metadata about where the original data is located, but doesn't copy the actual data at the initial creation. This makes snapshot creation virtually instantaneous, with little impact on the system taking the snapshot. The snapshot then tracks the original volume paying attention to changed blocks as writes are performed. As the blocks change, the original data is copied into the reserved storage capacity set aside for the snapshot prior to the original data being overwritten. The original data blocks snapped are copied just once at the first write request. This process ensures snapshot data is consistent with the exact time the snapshot was taken, and it's why the process is called "copy-on-write."
- Split-mirror - A clone or split-mirror snapshot creates an identical copy of the data. It requires as much storage capacity as the original data

Virtualization features -continued

- Live Migration - Live migration refers to the process of moving a running virtual machine or application between different physical machines without disconnecting the client or application. Memory, storage, and network connectivity of the virtual machine are transferred from the original host machine to the destination. When properly carried out, this process takes place without any noticeable effect from the point of view of the end user
- Live migration allows an administrator to take a virtual machine offline for maintenance or upgrading without subjecting the system's users to downtime.
- Requires a shared storage



Virtualization features -continued

- High Availability

High Availability refers to a system or component that is continuously operational for a desirably long length of time.

HA clusters or **failover clusters** are groups of computers that support server applications that can be reliably utilized with a minimum of down-time. They operate by using high availability software to harness redundant computers in groups or clusters that provide continued service