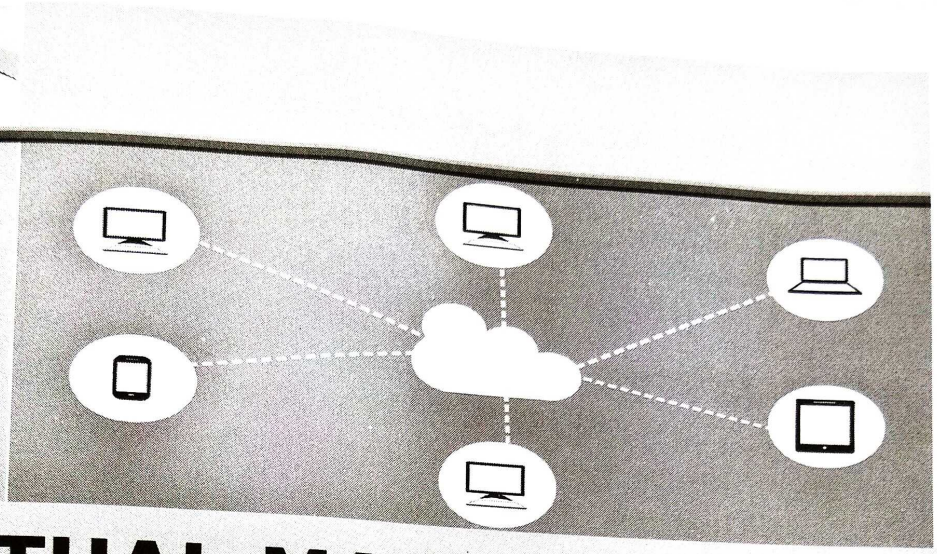
Chapter

7

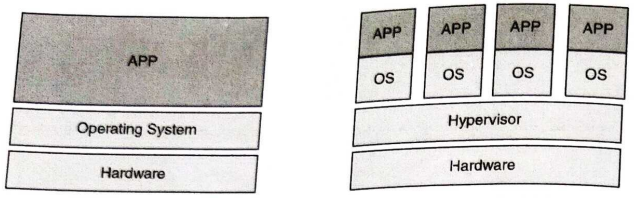


VIRTUAL MACHINES AND VIRTUALIZATION OF CLUSTERS AND DATA CENTERS

7.1 VIRTUALIZATION-AN INTRODUCTION

The foundation of cloud computing is *virtualization*. It is a process that allows for more efficient utilization of physical computer hardware and is the foundation of cloud computing. Virtualization is a technology that enables the creation of a virtual (rather than actual) version of computing resources, such as operating systems, servers, storage devices, or networks.

The process of virtualization uses software to create an abstraction layer over computer hardware that divides a single computer's hardware components—processors, memory, storage, and other components—into several virtual computers, or virtual machines (VMs). Each VM runs its own operating system (OS) and behaves like an independent computer, even though it is running on just a portion of the actual underlying computer hardware.



(a) Traditional Architecture (b) Virtualization Architecture

Software called *hypervisors* separate the physical resources from the virtual environments. A hypervisor creates and manages virtual machines (VMs), facilitating the abstraction and sharing of physical hardware resources. It sits between the hardware and VMs, allocating resources, enabling multiple operating systems to run simultaneously on a single physical machine, ensuring isolation, and managing interactions. The *Xen* hypervisor is an open source software program that is responsible for managing the low-level interactions that occur between virtual machines (VMs) and the physical hardware.

The following steps are involved in the virtualization process:

1. The physical resources and their physical environments are separated by hypervisors.
2. Resources from the physical world are taken and distributed among the different virtual environments as required.
3. System users work with and perform computations within the virtual environment.
4. Once the virtual environment is running, a user or program can send an instruction that requires extra resources form the physical environment. The hypervisor responds by sending the message to the physical system and storing the modifications. This procedure will proceed at a nearly natural pace.

Although the performance of this virtual system is not as good as performance of the operating system running on actual hardware, virtualization is a viable concept because most guest operating systems and applications do not require complete access to the underlying hardware. By doing this, the reliance on a specific hardware platform is eliminated, allowing for increased flexibility, control, and isolation.

7.1.1 Types of Virtualization

Virtualization can be applied in a variety of ways to meet your needs and transform your company. Here are just a few instances of how virtualization can help your company:

- **Hardware virtualization:** It refers to the process of creating virtual machines (VMs) that simulate the functionality of physical hardware components using a hypervisor. It abstracts and partitions the underlying physical hardware resources—like CPU, memory, storage, and networking—enabling multiple VMs to run different operating systems and applications independently on a single physical machine.
- **Server virtualization:** It involves creating multiple virtual instances (virtual machines - VMs) on a single physical server using a hypervisor. It abstracts the server's hardware resources, allowing various operating systems and applications to run independently in isolated environments. This consolidation optimizes resource utilization, enhances scalability, simplifies management, and reduces hardware costs by replacing several physical servers with a single server hosting multiple virtual instances.
- **Operating system (OS) virtualization:** It is also known as OS-level virtualization or containerization, enables multiple isolated user-space instances, called containers, to share a single OS kernel. Containers provide lightweight, efficient virtualization by leveraging the host OS's resources directly. They encapsulate applications and their dependencies, allowing portability, rapid deployment, and scalability. OS virtualization ensures agility, quick provisioning, and resource efficiency for various software development and deployment scenarios.
- **Storage virtualization:** It is the process of grouping the physical storage from multiple network storage devices so that it looks like a single storage device. Storage virtualization is also implemented by using software applications. To users, virtual storage appears like a standard read or write to a physical drive. It hides the complexity of the storage system, which allows users and administrators to perform tasks such as backup, archiving and recovery in an easier, less time-consuming manner.

7.1.2 Benefits of Virtualization

Virtualization offers a range of benefits across various domains:

- **Resource Optimization:** Efficiently utilizes hardware by running multiple virtual machines (VMs) on a single physical server, reducing hardware costs and increasing resource utilization.
- **Cost Savings:** Lowers expenses by minimizing the number of physical servers needed, reducing power consumption, cooling costs, and physical space requirements.
- **Scalability:** Allows easy scaling of computing resources by provisioning or removing VMs according to demand, providing flexibility in managing workloads.
- **Isolation and Security:** Ensures that VMs operate independently, enhancing security by containing issues within isolated environments, thus preventing potential threats from spreading across systems.
- **Disaster Recovery:** Simplifies backup and recovery processes by encapsulating entire VMs, making it easier to replicate, back up, and restore in case of failures or disasters.
- **Testing and Development:** Facilitates software development and testing by creating isolated environments to test applications across different operating systems and configurations.
- **High Availability:** Offers features like live migration, allowing VMs to move seamlessly between physical servers without interrupting services, ensuring continuous availability.
- **Management Simplification:** Centralizes the management of computing resources, making it easier to administer, monitor, and automate tasks like provisioning and resource allocation.
- **Consolidation and Green Initiatives:** Reduces the number of physical servers, leading to a smaller data center footprint, decreased power consumption, and contributing to environmental sustainability.
- **Business Continuity:** Enhances business resilience by enabling quicker recovery times and minimizing downtime in case of hardware failures or system issues.

Overall, virtualization provides a robust framework for optimizing resources, reducing costs, enhancing security, and improving efficiency across IT infrastructures, making it a fundamental technology in modern computing environments.

7.2 LEVEL OF VIRTUALIZATION

Virtualization setup is a difficult task because your computer runs operating systems that are designed to run on specific kinds of hardware. It is therefore challenging to run different operating system types on the same hardware.

In order to enable smoother operation, we need a hypervisor, which serves as a bridge between your hardware and virtual operating system.

But, for implementation levels you need to undergo five levels of virtualization in cloud computing. Let's look at them:

7.2.1 Instruction Set Architecture (ISA) level

Virtualization at the Instruction Set Architecture (ISA) level involves abstracting and emulating the hardware's instruction set to enable multiple operating systems (OS) to run concurrently on the same physical machine. It involves emulating hardware instructions of a particular architecture to enable the execution of operating systems (OS) designed for that ISA on a different underlying architecture. This emulation facilitates running unmodified guest OSes on a virtual machine (VM) with a distinct ISA.

For instance, an x86-based hypervisor emulating ARM instructions allows an ARM-based guest OS, designed for mobile devices or embedded systems, to run on x86 hardware. The hypervisor intercepts and translates ARM instructions into equivalent x86 instructions to ensure compatibility and execution on the underlying x86 processor.

Another technique is *dynamic binary translation* where software translates machine code instructions from one architecture to another at runtime. It intercepts instructions meant for the guest operating system, translates them into instructions compatible with the host system's architecture, and then executes them.

7.2.2 Hardware abstraction level of Abstraction

Hardware abstraction level in cloud computing virtualization involves creating a layer of abstraction between physical hardware resources and the software running on top, facilitating efficient resource utilization, scalability, and flexibility. This abstraction allows cloud providers to pool and allocate hardware resources dynamically to multiple users or applications without requiring detailed knowledge of the underlying hardware specifics.

At this level, a hypervisor or virtual machine monitor (VMM) plays a crucial role by abstracting and managing physical hardware resources, such as CPU, memory, storage, and networking, to create virtualized environments. It enables the deployment of multiple virtual machines (VMs) or containers with varied operating systems and applications on a single physical server or across a cluster of servers.

This abstraction hides hardware complexities, enabling easy migration, scalability, and isolation of workloads. It facilitates resource optimization by dynamically allocating or reallocating hardware resources as per demand, enhancing the overall efficiency and agility of cloud computing infrastructures.

7.2.3 Operating System Level of Abstraction

At the level of the operating system, the virtualization model is capable of creating a layer that is abstract between the operating system and the application.

It involves creating isolated user-space instances called containers within a single host operating system. Unlike traditional virtualization methods that run multiple operating systems on a hypervisor, OS-level virtualization shares a single OS kernel across containers while keeping them isolated from each other.

Containers encapsulate applications, dependencies, and their runtime environment, allowing them to run consistently across various computing environments. Technologies like *Docker* and *Kubernetes* are widely used for managing and orchestrating containerized applications.

This approach offers lightweight, rapid provisioning, and efficient utilization of resources as containers share the host OS kernel, eliminating the need for a separate OS for each container. OS-level virtualization ensures quick startup times, high scalability, and easier management, making it well-suited for deploying microservices, DevOps practices, and scalable cloud-native applications. However, containers are limited to running applications compatible with the host OS kernel, unlike hypervisor-based virtualization allowing different OSes to run simultaneously.

7.2.4 Library Support Level

Virtualization at the user-library level involves abstracting and managing libraries or software dependencies to provide users with consistent access to required resources across different environments. This method allows users to access specific libraries or software

components without worrying about underlying infrastructure differences or compatibility issues.

Most applications use APIs exported by user-level libraries rather than using lengthy system calls by the OS. Application Programming Interfaces (APIs) enable users to access specific functionalities of libraries without directly interacting with underlying hardware or infrastructure. APIs act as intermediary interfaces, allowing seamless communication between applications and libraries. By utilizing APIs, users can access and leverage the functionalities provided by libraries or software components across different environments, ensuring consistent behavior and ease of integration while abstracting complexities associated with underlying systems.

7.2.5 User-Application Level of Abstraction

Virtualization at the user-application level involves creating an abstraction layer between the user and applications, allowing multiple applications to run concurrently without interference or conflict. It involves encapsulating entire software applications, including their configurations, dependencies, and runtime environments, into deployable units.

Technologies like containerization (e.g., Docker) and application virtualization achieve this by packaging applications into portable containers or images. These containers contain everything an application needs to run independently, such as code, libraries, settings, and dependencies, ensuring a consistent runtime environment. Application-level virtualization facilitates easy deployment, scaling, and management of applications within cloud environments. It enhances agility, allowing developers to create, package, and deploy applications more efficiently while ensuring consistent performance and behavior across various cloud platforms. This approach optimizes resource utilization, simplifies application deployment, and supports modern DevOps practices by streamlining the development and deployment lifecycle in cloud computing.

7.3 VIRTUALIZATION STRUCTURES/TOOLS AND MECHANISMS

Before virtualization, hardware is managed by the operating. After virtualization, a virtualization layer is inserted between the hardware and the operating system. In this scenario, the virtualization layer is responsible for converting some of the physical

hardware into virtual hardware. As a result, multiple operating systems, including Windows and Linux, can run concurrently on a single physical computer. Depending on the position of the virtualization layer, there are following classes of VM architectures:

1. Hypervisor
2. Full Virtualization
3. Para Virtualization

7.3.1 Hypervisor

A hypervisor, also known as a Virtual Machine Monitor (VMM), is a software layer that enables the creation and management of virtual machines (VMs) on physical hardware. It abstracts and partitions the underlying hardware resources, allowing multiple VMs to run independently with their own operating systems and applications. There are two types:

- Type-1 virtualization, also known as *bare-metal* or *native* virtualization, involves a hypervisor that runs directly on the physical hardware without the need for a host operating system. This hypervisor manages and allocates resources, facilitating the creation and operation of multiple virtual machines (VMs) independently. Examples include VMware vSphere/ESXi, Microsoft Hyper-V Server, and Xen.
- Type-2 virtualization, referred to as *hosted* virtualization, utilizes a hypervisor installed on top of a host operating system. This hypervisor creates and manages VMs within the host OS environment, allowing users to run multiple operating systems concurrently on their personal computers or workstations. Popular examples include VMware Workstation, Oracle VirtualBox, and Parallels Desktop.

Hypervisors facilitate efficient resource allocation, isolation, and control over VMs, enabling consolidation of multiple workloads, enhanced flexibility, and the optimization of hardware utilization in virtualized environments.

A micro-kernel hypervisor is a lightweight type of hypervisor that employs a minimalistic design, separating core functionalities into small modules or services. It runs essential functions like memory management, scheduling, and inter-process communication in a micro-kernel, while delegating device drivers and other services to user-space components. This architecture enhances security and flexibility by reducing the trusted computing base, enabling efficient and modular virtualization solutions.

7.3.1.1 The Xen Architecture

Xen is an open-source hypervisor that facilitates the creation and management of virtual machines (VMs) on hardware platforms as shown in fig. 7.1.

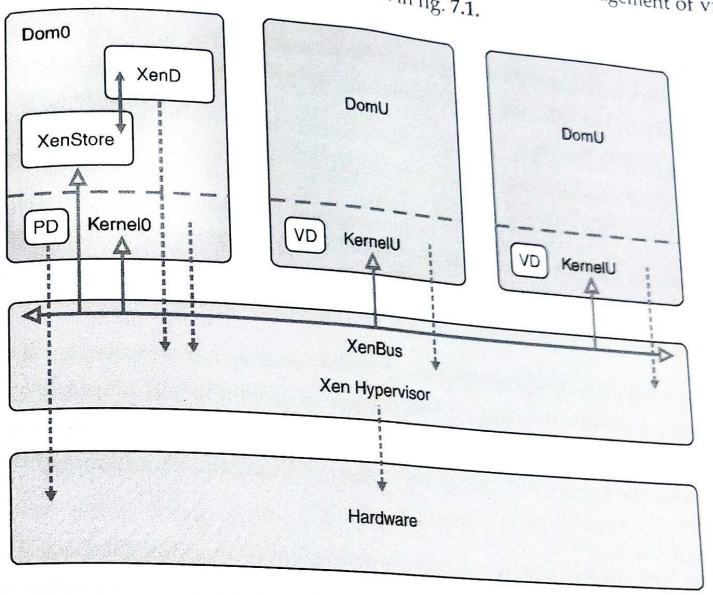


Fig. 7.1: Xen Architecture

Xen architecture comprises several components that work together to enable virtualization:

- **Xen Hypervisor:** The core component, often referred to as the Xen hypervisor, runs directly on the hardware (Type 1 hypervisor). It manages and controls the allocation of physical resources such as CPU, memory, and I/O devices to multiple VMs. Xen provides the fundamental functions for virtualization, including scheduling, memory management, and hardware access.
- **Domain 0 (Dom0):** Dom0 is a privileged VM that serves as the control domain for managing other VMs and device drivers. It has direct access to hardware and performs administrative tasks such as creating, starting, stopping, and managing other VMs (also known as DomUs or unprivileged domains). Dom0 runs a modified operating system, often a lightweight version of Linux.

- **Domain U (DomU):** DomUs are unprivileged VMs that run guest operating systems and applications. These VMs rely on Dom0 for resource management and access to physical devices.
- **XenStore:** It's a repository for sharing configuration and state information among different domains. XenStore helps in communication and coordination between Dom0 and DomUs, storing configuration details and runtime properties.
- **XenBus:** XenBus provides a communication interface between different domains within the Xen hypervisor. It enables inter-domain communication and facilitates the exchange of control and data messages.
- **Backend and Frontend Drivers:** Backend drivers run in Dom0 and communicate with the physical hardware. Frontend drivers reside in DomUs and interact with the backend drivers, providing access to virtualized resources.

Xen's architecture enables efficient resource sharing and isolation among VMs, contributing to better security, performance, and flexibility in deploying virtualized environments. It supports paravirtualization (where guest OSes are modified to work efficiently with the hypervisor) and hardware-assisted virtualization (using CPU extensions like Intel VT-x or AMD-V) for improved performance.

Xen has been widely used in cloud computing platforms and data centers due to its stability, performance, and open-source nature, providing a robust foundation for building virtualized infrastructures.

7.3.2 Full virtualization

Full virtualization allows multiple virtual machines (VMs) to run simultaneously on a single physical host, emulating complete hardware environments for each VM. A hypervisor abstracts and manages physical resources, presenting them as virtual equivalents to VMs. Guest operating systems run unmodified, unaware they operate in a virtual environment. The hypervisor translates privileged instructions and manages access to physical resources, ensuring isolation and security between VMs. This approach enables efficient resource sharing, hardware independence, and seamless migration of VMs across hosts. Although it may incur performance overhead due to emulation, full virtualization supports diverse operating systems and facilitates robust, isolated environments for various applications. Full virtualization often involves techniques like *binary translation* or *host-based virtualization*.

7.3.2.1 Binary Translation with Full Virtualization

Binary translation is a technique used in full virtualization where the hypervisor dynamically intercepts and translates sensitive or privileged instructions from guest VMs into equivalent instructions compatible with the underlying hardware. It allows execution of instructions that would normally require direct hardware access. The hypervisor converts these instructions at runtime, ensuring compatibility between the guest's instruction set and the host's architecture. This process enables guest operating systems, originally designed for different hardware, to run efficiently on the virtualized environment without modifying their code, enhancing compatibility and facilitating the execution of diverse operating systems on the same physical hardware.

7.3.2.2 Host-based virtualization

Host-based virtualization, also known as Type 2 virtualization, refers to virtualization that occurs within a host operating system. Unlike Type 1 virtualization where the hypervisor runs directly on the hardware, Type 2 virtualization involves a hypervisor that operates as an application within a host operating system.

A hypervisor application, residing on top of the host OS, creates and manages virtual machines (VMs). It presents virtualized hardware to VMs, enabling multiple guest operating systems to run concurrently. The hypervisor mediates access to physical resources, such as CPU, memory, and storage, sharing them among VMs. Each VM runs its own isolated environment within the host OS, allowing users to run different operating systems or test software without affecting their primary system.

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. However, this method incurs some performance overhead due to the hypervisor running as an application within the host OS.

7.3.3 Para-Virtualization

Para-virtualization modifies guest operating systems to collaborate closely with the hypervisor, enhancing performance by replacing certain non-virtualizable instructions with hypercalls, optimized for virtual environments. Unlike full virtualization, para-virtualized

guest operating systems are aware of the virtual environment and communicate directly with the hypervisor. This collaboration reduces overhead, improving efficiency in resource management, I/O operations, and memory access. However, it requires OS modifications to support the hypervisor interface, limiting compatibility but providing higher performance gains compared to full virtualization in environments where OS customization is feasible and desirable. Examples include Xen and some Linux distributions supporting para-virtualized kernels.

7.3.3.1 Para-Virtualization Architecture

Para-virtualization architecture involves modifications made to guest operating systems (OSes) to collaborate more efficiently with the hypervisor or virtual machine monitor (VMM). This collaboration optimizes interactions between the OS and the hypervisor, reducing the overhead typically associated with full virtualization.

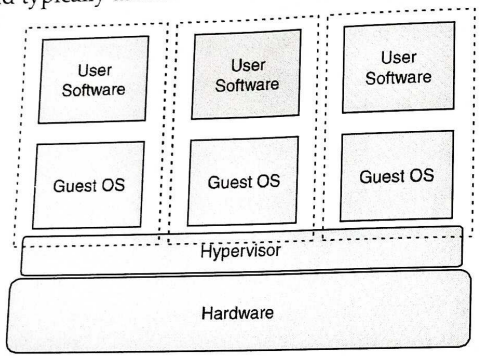


Fig. 7.2. The Architecture of Para Virtualization

Key elements of para-virtualization architecture include:

- **Hypervisor Interface:** The hypervisor provides an interface through which the modified guest OS communicates with it. This interface facilitates efficient interactions, allowing the guest OS to perform tasks directly on the underlying hardware with the assistance of the hypervisor.
- **Hypercalls:** In para-virtualization, certain instructions that cannot be directly executed in a virtualized environment are replaced with hypercalls. Hypercalls are specific calls made by the guest OS to the hypervisor, allowing it to perform tasks that require direct hardware access, such as memory management, I/O operations, and scheduling.

- **OS Modifications:** Guest OSes need to be modified or adapted to support the hypervisor interface and hypercalls. These modifications involve changes to the OS kernel, making it aware of the virtual environment and enabling efficient communication and resource management with the hypervisor.

By avoiding the overhead associated with emulating certain instructions and enabling direct communication between the guest OS and the hypervisor, para-virtualization architecture improves overall system performance, particularly in scenarios involving intensive I/O operations or memory access.

Xen is a prominent example of a hypervisor that employs para-virtualization. It requires guest OSes to be modified with Xen-aware drivers to take advantage of its para-virtualized architecture, achieving higher performance and efficiency compared to full virtualization in certain scenarios.

7.4 VIRTUALIZATION OF CPU, MEMORY, AND I/O DEVICES

Virtualization partitions a physical CPU, memory, and I/O devices into multiple virtual instances, mimicking independent systems. The CPU's resources are allocated to virtual machines (VMs), allowing simultaneous execution of multiple operating systems. Memory and I/O devices are also abstracted, enhancing system flexibility, resource utilization, and scalability in computing environments.

7.4.1 CPU Virtualization

The objective of CPU virtualization is to make a CPU run in the same way that two separate CPUs would run. In effect, this is like running two separate computers on a single physical machine. Perhaps the most common reason for doing this is to run two different operating systems on one machine.

CPU virtualization distributes hosting resources among all virtual machines as if they were different virtual processors, and each virtual machine functions as a physical machine. When all hosting services receive the request, each virtual machine shares physical resources. Finally, the virtual machines get a share of the single CPU allocated to them being a single-processor acting as a dual-processor.

Using CPU virtualization, it is possible to run two separate operating systems simultaneously. For instance, a computer running Linux could use virtualization to

Windows®. It's also possible to use CPU virtualization to run Windows® on a Mac® or Linux PC, or to run Mac OS® and Linux at the same time.

The ability for multiple people to use a single computer at once is another advantage of virtualization. This would function by connecting a single CPU-powered computer running virtualization software to several "desks," each of which has a keyboard, mouse, and monitor. The operating system would then be running on separate copies of each user's CPU. An emulator manages everything, directing software to operate in accordance with it. That being said, CPU Virtualization is not an emulator. The emulator operates in the same manner as a typical computer system. Just like a physical machine, it duplicates the same copy of data and produces the same output. The emulator performs the same way as a normal computer machine does. It replicates the same copy or data and generates the same output just like a physical machine does. The emulation function offers great portability and facilitates working on a single platform, acting like working on multiple platforms. The main two types of CPU virtualization are:

1. Software-Based CPU Virtualization

In this approach, a hypervisor or virtual machine monitor (VMM) emulates the hardware environment required for each virtual machine (VM). *Binary translation* is a technique used by hypervisors to enable virtualization on hardware. The hypervisor dynamically translates machine-level code from the guest operating system into an equivalent code understandable by the underlying hardware. This translation occurs during runtime, allowing the guest OS to run on the virtualized environment seamlessly. By intercepting and converting instructions on the fly, binary translation facilitates the execution of guest code on a virtual machine, enabling multiple operating systems to run concurrently on hardware that may not inherently support virtualization, thereby expanding the scope of virtualization possibilities.

2. Hardware-Assisted CPU Virtualization

Hardware-assisted CPU virtualization utilizes specialized features within modern processors, such as Intel VT-x or AMD-V, to optimize and streamline virtualization processes. These hardware extensions introduce new instructions and functionalities that enhance the efficiency and security of virtualized environments. They enable the hypervisor to more effectively manage virtual machines (VMs) by offloading certain tasks from

software to dedicated hardware. The best part in hardware-assisted CPU Virtualization is that there is no requirement for translation while using it for hardware assistance.

Key components include the Virtual Machine Control Structure (VMCS) and enhanced memory management, allowing the hypervisor to efficiently manage and switch between VMs while ensuring secure isolation. Hardware-based I/O virtualization features improve input/output operations' performance within virtualized setups. Additionally, security measures integrated into hardware-assisted virtualization bolster isolation between VMs, reducing vulnerabilities.

By leveraging these hardware capabilities, hardware-assisted CPU virtualization significantly minimizes performance overhead associated with traditional software-only virtualization. It enables more efficient resource allocation, heightened security, and smoother operation of multiple VMs concurrently on a single physical system, benefiting various computing environments like data centers and cloud platforms.

7.4.2 Memory Virtualization

Memory Virtualization can be understood as a concept where multiple physical memories across different servers are put together as one to form a singular virtual memory. This allows you an access to a bigger memory to work on.

Memory virtualization in cloud computing is implemented through various techniques and mechanisms that facilitate efficient management, allocation, and utilization of memory resources across multiple virtual machines (VMs) or instances within cloud environments. Here are key methods used for memory virtualization in cloud computing:

- **Hypervisor-Based Memory Management:** Hypervisors, such as VMware ESXi, Microsoft Hyper-V, KVM, or Xen, manage memory virtualization in cloud environments. They abstract physical memory into virtual memory pools that can be allocated to different VMs. The hypervisor handles memory address translation, mapping virtual memory addresses used by each VM to physical memory addresses on the host system.
- **Page Sharing:** Page sharing, a key technique in memory virtualization identifies duplicate memory pages across multiple virtual machines (VMs). It consolidates identical memory pages into a single physical page, reducing memory consumption. Through this process, common pages among VMs are shared, minimizing redundant data storage. Page sharing occurs transparently to VMs, optimizing memory usage

in virtualized environments. By efficiently utilizing memory resources and reducing redundancy, page sharing enhances the scalability and performance of virtualized systems, enabling more VMs to operate on the same physical infrastructure without compromising system stability or responsiveness.

- **Memory Over commitment:** Cloud platforms often utilize memory over commitment, allowing the allocation of more virtual memory than the actual physical memory available. This technique relies on memory management systems to handle situations where demand exceeds physical capacity.
- **Dynamic Memory Ballooning:** Memory ballooning is a memory management technique in virtualized environments where a hypervisor dynamically adjusts memory allocations across virtual machines (VMs). It involves a balloon driver installed within VMs, allowing the hypervisor to request memory from a VM when needed. The balloon driver within the VM inflates, consuming memory, which the hypervisor then reallocates to other VMs or the host system. This method helps optimize memory utilization by redistributing resources based on VM requirements. When memory demand decreases, the balloon driver deflates, returning the allocated memory to the VM.

7.4.3 I/O Virtualization

I/O virtualization, also known as input/output virtualization, is a fundamental aspect of virtualization technology that aims to maximize the allocation and management of input and output resources in a virtualized environment, including network interfaces and storage devices. I/O virtualization enhances resource utilization by enabling efficient sharing of physical I/O resources among multiple virtual machines (VMs) while preserving isolation and performance. Thanks to this technology, administrators can now allocate, manage, and control I/O resources dynamically in modern data centers and cloud environments.

Here's an overview of how I/O virtualization works:

- **Abstraction of Physical Resources:** The process begins by abstracting physical I/O resources such as network interfaces, storage devices (disks, SSDs), GPUs, and other peripherals. This abstraction creates virtual representations of these resources that can be assigned to and utilized by VMs or containers.

- **Hypervisor or Virtualization Layer:** In a virtualized environment, a hypervisor or virtualization layer manages the allocation of these virtualized I/O resources to different VMs or containers. The hypervisor sits between the hardware and the virtual machines, controlling access to physical resources.
- **Virtual I/O Devices:** Virtual instances of I/O devices are created for the VMs or containers to use. These devices appear and function like their physical counterparts but are actually managed by the hypervisor, which oversees their interactions with the physical hardware.
- **I/O Multiplexing and Sharing:** I/O virtualization techniques enable efficient sharing of physical I/O resources among multiple VMs or containers. Technologies like I/O multiplexing allow several VMs to share the same physical device while maintaining isolation and security.
- **SR-IOV and Direct Assignment:** Single Root I/O Virtualization (SR-IOV) allows a physical device to be divided into multiple virtual functions, each assigned directly to a VM. This bypasses the hypervisor for certain I/O operations, reducing latency and enhancing performance for those specific VMs.

7.5 VIRTUAL CLUSTERS AND RESOURCE MANAGEMENT

In cloud computing, virtual clusters and resource management play crucial roles in optimizing resource utilization and providing efficient and scalable services. Here's an overview of virtual clusters and resource management in the context of cloud computing:

- **Virtual Clusters:** Virtual clusters are logical groupings of virtual machines (VMs) or containers within a cloud environment. They provide a way to create isolated and dedicated environments for specific applications or workloads. Virtual clusters allow organizations to achieve better resource utilization by consolidating multiple workloads on a shared infrastructure. They enable flexibility and scalability as virtual clusters can be easily provisioned, scaled up or down, and managed independently.
- **Resource Management:** Resource management involves efficiently allocating and managing the computing resources (e.g., CPU, memory, storage) in a cloud environment. It aims to optimize resource utilization, performance, and cost-effectiveness while meeting the demands of various applications and users. Resource management techniques and tools are employed to ensure fair resource allocation, load balancing, and scalability.

Key aspects related to virtual clusters and resource management

Here are some key aspects related to virtual clusters and resource management in cloud computing:

- **Dynamic Resource Allocation:** Virtual clusters allow for dynamic allocation of resources based on the changing demands of applications. This resource management involves provisioning and de-provisioning resources such as VMs, CPU, memory, and storage as needed.
- **Resource Scheduling:** Efficiently scheduling workloads and tasks across the available resources is crucial for maximizing resource utilization. Various scheduling algorithms and policies are employed to allocate resources effectively, considering factors like workload priorities, resource constraints, and performance objectives.
- **Load Balancing:** Distributing workloads evenly across the virtual cluster nodes helps prevent resource bottlenecks and ensures optimal resource utilization. Load balancing techniques manage and balance the computing load across multiple nodes or VMs within the cluster.
- **Monitoring and Performance Optimization:** Continuous monitoring of resource usage, performance metrics, and application behavior helps in identifying bottlenecks or underutilized resources. Optimization strategies may involve scaling resources up or down, implementing caching mechanisms, or adjusting configurations to enhance performance.
- **Fault Tolerance and Resilience:** Virtual clusters need mechanisms for fault tolerance to ensure continuous operation in the event of hardware failures or other issues. This involves techniques like data replication, redundancy, and failover mechanisms to maintain high availability.
- **Elasticity and Scalability:** Virtual clusters should be designed to scale both vertically (increasing resources within a node) and horizontally (adding more nodes) to accommodate changing workloads and resource demands.
- **Cost Management:** Resource management also involves cost optimization by rightsizing resources, using reserved instances, or employing auto-scaling features to match resource usage with the actual requirements, thereby minimizing unnecessary expenses.

Migration of Memory, Files, and Network Resources in Virtualization

Since 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. Early configuration and management systems focus on expressive and scalable mechanisms for defining clusters for specific types of service, and physically partition cluster nodes among those types. When one system migrates to another physical node, we should consider the following issues.

- **Memory Migration:** Memory migration refers to the process of transferring the active memory contents of a virtual machine (VM) from one physical host to another without disrupting the VM's operation. This process is known as live or hot migration. It involves copying the memory pages of the running VM to the destination host while the VM continues to execute. Live migration minimizes downtime, allowing for workload balancing, hardware maintenance, or resource optimization.
- **File Migration:** File migration involves moving the virtual disks, configurations, and associated files of a VM from one storage location to another. This could include transferring virtual hard disks (VHDs) or virtual machine disk files (VMDKs) to different storage systems or locations. File migration helps in load balancing, upgrading storage, or relocating VMs without significant downtime.
- **Network Resource Migration:** Network resource migration involves transferring networking configurations, settings, and connections associated with a VM from one physical network or subnet to another. This process ensures that a VM maintains its network connectivity and settings when it is moved between hosts or data centers. It includes updating IP addresses, MAC addresses, and network settings to ensure seamless communication post-migration.

Migration techniques in virtualization environments use specialized software and protocols such as VMotion (VMware), Live Migration (Hyper-V), or OpenStack's live migration capabilities. These technologies coordinate the transfer of memory, files, and network resources while maintaining system integrity, ensuring data consistency, and minimizing downtime for the VMs being migrated.

Overall, the migration of memory, files, and network resources in virtualization enables workload mobility, resource optimization, and seamless management of virtualized environments.

7.6 VIRTUALIZATION FOR DATA-CENTER AUTOMATION

A data center is basically a physical location that holds all operations and machinery in charge of storing, processing, and distributing data for an organization. Over the last decade, new technologies have emerged that are physically transforming the traditional data center due to advancements in cloud computing and a rise in the need for adaptable IT solutions. Since server virtualization has gained popularity, many companies have switched from having physical data centers on-site to virtualized data center solutions. Data center virtualization is the process of creating a virtual data center that still performs all of the functions a data center does, but freeing it from physical servers and other equipment.

In a virtualized data center, a virtual server, also known as software-defined data center (SDDC) is created from traditional, physical servers. With the use of a hypervisor, this procedure abstracts physical hardware by imitating its CPU, operating system, and other resources. A hypervisor (or virtual machine monitor, VMM, virtualizer) is a software that creates and manages a virtual machine. It views hardware resources like CPU, memory, and storage as a pool that may be readily redistributed to other virtual machines or between already running ones.

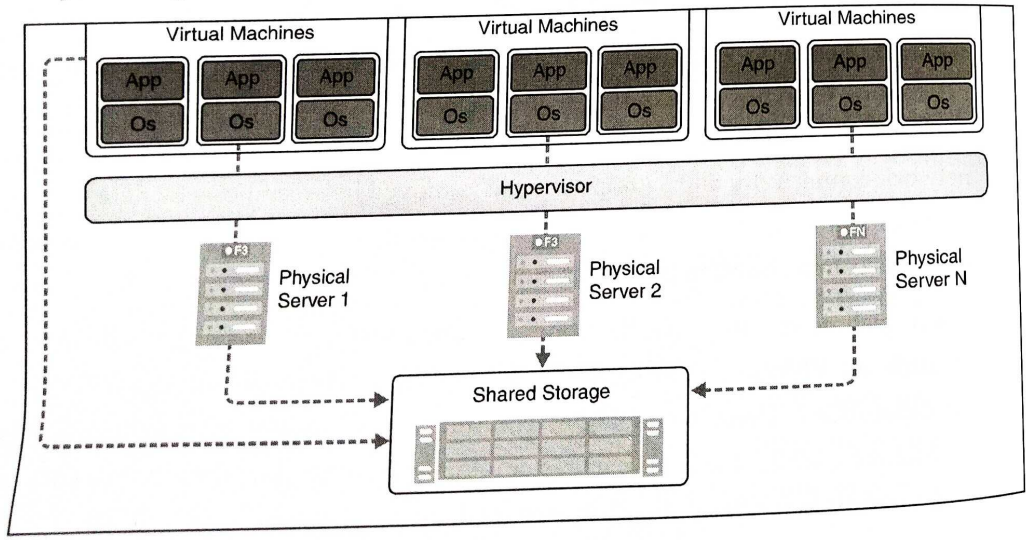


Fig. 7.3: Virtualization Architecture for Data Center

7.6.1 Server Consolidation in Data Centers

Numerous heterogeneous workloads can run on servers at different times in data centers. Therefore, it is common that most servers in data centers are underutilized. Let's take an example where you have a file server, a backup server, and a web server. The web server is always running but doesn't use a lot of resources. During business hours, the file server is operational, however it is largely idle at night. The backup server uses the most resources and runs nightly, not during the day or in the evening. The file server and the backup server aren't maxing out at the same time, meaning they are idle most of the time. A large amount of hardware, space, power, and management cost of these servers is wasted.

The cost of increasing the power of each server is reduced by one-third if all three of them are virtual machines running on the same system. This is because there is only one machine that needs to be upgraded.

Server consolidation is a strategy to improve the low utility ratio of hardware resources by reducing the number of physical servers. In cloud computing, server consolidation is the act of merging several servers into a single, more powerful server or server cluster.

Types of Server Consolidation

1. **Logical Consolidation**: In logical server consolidation, multiple virtual servers are consolidated onto a single physical server. Each virtual server is isolated from the others and has its own operating system and applications, but shares the same physical resources such as CPU, RAM, and storage. This enables businesses to operate several virtual servers on a single physical server, which can result in significant financial savings and enhanced functionality. Organizations can more readily adapt to changing business needs thanks to the ease with which virtual servers can be added or removed as needed.
2. **Physical Consolidation**: Physical consolidation refers to the consolidation of multiple physical servers into a single, more powerful server or cluster of servers. This can be accomplished by replacing multiple older servers with newer, more powerful servers, or by adding additional resources to existing servers such as memory and storage. Organizations can benefit from physical consolidation to improve the performance and efficiency of their cloud computing environment.

Benefits of Server Consolidation

- **Cost Savings:** Reduces hardware expenses, power consumption, cooling costs, and physical space requirements.
- **Enhanced Resource Utilization:** Maximizes server capacity by consolidating workloads onto fewer physical machines, reducing underutilized resources.
- **Simplified Management:** Centralizes administration tasks, streamlines updates, backups, and maintenance activities, reducing administrative overhead.
- **Improved Scalability:** Allows for dynamic allocation of resources to meet changing demands, ensuring efficient use of computing power.
- **Increased Flexibility:** Facilitates easier provisioning and deployment of virtual machines, adapting swiftly to varying workloads.
- **Enhanced Performance:** Optimizes system performance by efficiently allocating resources and minimizing resource contention.
- **Better Disaster Recovery:** Simplifies backup and recovery processes, making it easier to restore data and applications in case of failures or disasters.
- **Environmental Impact:** Reduces carbon footprint through decreased energy consumption and fewer physical servers, contributing to eco-friendliness.

QUESTIONS

Short Answer Questions

Q1. What is meant by virtualization?

Ans: Virtualization refers to the process of creating a virtual, rather than actual, version of something, such as an operating system, server, storage device, or network resources. It enables the efficient utilization of physical resources by abstracting them, allowing multiple virtual instances to run independently on a single physical machine. This technology enhances flexibility, scalability, and cost-effectiveness in IT environments by isolating resources, optimizing performance, and simplifying management through the use of software-driven emulation.

Q2. What are the implementation levels of virtualization?

Ans: Virtualization can be implemented at various levels:

- **Instruction Set Architecture (ISA) Level:** It involves abstracting and emulating the hardware's instruction set to enable multiple operating systems (OS) to run concurrently on the same physical machine.
- **Hardware Level:** It involves creating virtual machines (VMs) on physical hardware using hypervisors like VMware.
- **Operating System Level:** It involves creating isolated user-space instances called containers within a single host operating system.
- **Library Support Level:** It involves virtualizing at the software library level, allowing multiple applications to share libraries efficiently.
- **User-application level:** It involves creating an abstraction layer between the user and applications, allowing multiple applications to run concurrently without interference or conflict.

Q3. What is OS-level virtualization?

Ans: OS-level virtualization, also known as containerization, enables multiple isolated user-space instances, known as containers, to share a single host OS kernel. Containers utilize the same OS kernel but remain independent in their user space, allowing efficient resource utilization, rapid deployment, and scalability. Popular containerization platforms like Docker, LXC, and Kubernetes use this method.

Q4. What is ISA level virtualization?

Ans: Instruction Set Architecture (ISA) level virtualization involves emulating a different instruction set on a host processor, allowing software designed for one architecture to run on another. This method often uses binary translation or interpretation to convert instructions, enabling compatibility between diverse hardware architectures, like running x86 software on ARM processors or vice versa, enhancing cross-platform functionality.

Q5. What is User-Application Level virtualization?

Ans: User-Application Level virtualization operates by encapsulating individual applications and their dependencies, allowing them to run independently without affecting the host system. This method isolates applications, ensuring portability,

security, and compatibility across different environments. Technologies like application virtualization, containerization, and sandboxing implement this approach, facilitating efficient deployment and management of specific software components.

Q6. What is hardware level virtualization?

Ans: Hardware level virtualization involves using a hypervisor to create and manage multiple *virtual machines* (VMs) on a single physical hardware platform. This method allows simultaneous execution of various operating systems and applications, providing isolation, resource allocation, and hardware abstraction. Hypervisors like *VMware* and *KVM* facilitate hardware-level virtualization, enabling efficient utilization of computing resources.

Q7. What is library level virtualization?

Ans: Library-level virtualization refers to a technique where multiple applications share common libraries, allowing efficient resource usage and reducing redundancy. It involves abstracting and managing software libraries separately from applications, enabling them to access shared resources dynamically. This method enhances compatibility, simplifies updates, and optimizes memory usage by eliminating the need for redundant copies of shared libraries across applications.

Q8. What are the types of VMM Design requirements?

Ans: Virtual Machine Monitor (VMM) design requirements include:

- *Isolation*: Ensuring VMs remain independent and secure from each other.
- *Performance*: Optimizing resource allocation for efficient VM operations.
- *Compatibility*: Supporting diverse operating systems and hardware configurations.
- *Resource Control*: Managing and allocating resources effectively among VMs.
- *Reliability:* Ensuring stable operation and fault tolerance within the virtualized environment.

Q9. What are the advantages of OS level Virtualization?

Ans: OS-level virtualization, such as containerization, offers advantages like:

- *Efficiency:* Minimal overhead as containers share the host OS kernel.

- *Rapid Deployment:* Quick start-up and shutdown times for containers.
- *Resource Efficiency:* Lightweight, utilizing fewer resources than VMs.
- *Scalability:* Easily manage multiple containers on a single host.
- *Portability:* Consistent behavior across various environments due to shared OS components.

Q10. Write short notes on Xen architecture?

Ans: Xen, an open-source hypervisor, employs a microkernel architecture, separating the hypervisor (Domain 0) from guest domains (DomU). It facilitates hardware resource management, memory allocation, and I/O handling. Domain 0 manages hardware and serves as a control domain, while various guest domains run individual operating systems. Xen utilizes paravirtualization or hardware-assisted virtualization for efficient resource allocation and guest OS execution. It offers robust isolation, performance optimization, and supports multiple OS types within its virtualized environment.

Q11. What are the types of hardware virtualization?

Ans: Hardware virtualization encompasses two primary types:

- *Full Virtualization:* Allows unmodified guest operating systems to run on a virtual machine (VM) by emulating hardware components through a hypervisor.
- *Para-virtualization:* Requires guest OSs to be modified to interact with the hypervisor, enhancing performance by providing direct communication and eliminating some emulation overhead.

Q12. Write short notes about memory virtualization?

Ans: Memory virtualization involves abstracting physical memory from the underlying hardware, allowing multiple processes or virtual machines to utilize memory efficiently. Techniques like page tables, memory allocation, and address translation manage memory allocation and access. Memory virtualization ensures isolation, optimizes memory utilization, and enables dynamic allocation, enhancing overall system performance in virtualized environments.

Q13. Write short notes about CPU virtualization?

Ans: CPU virtualization involves creating multiple virtual CPU instances from a physical processor. It's managed by the hypervisor, which allocates CPU resources to virtual

machines or processes. Techniques like time slicing, scheduling, and hardware-assisted virtualization (e.g., Intel VT-x, AMD-V) enable efficient sharing of CPU resources among multiple virtual environments, ensuring fair access and optimal performance without interference between virtual machines.

Q14. What is Hardware-Assisted CPU Virtualization?

Ans: Hardware-Assisted CPU Virtualization involves CPU features, like Intel VT-x or AMD-V, enhancing virtualization performance by providing hardware-level support for virtualization tasks. These extensions allow the hypervisor to efficiently manage and allocate CPU resources among virtual machines. Hardware-assisted virtualization reduces overhead by enabling direct interaction between the hypervisor and CPU, enhancing overall virtualization performance and securit

Q15. What is IO virtualization?

Ans: IO (Input/Output) virtualization involves abstracting and managing physical I/O devices, such as network adapters, storage controllers, and other peripherals, for use by multiple virtual machines. This technique enables the efficient sharing of I/O resources among virtual environments. IO virtualization techniques include device emulation, passthrough, and direct assignment, optimizing access to I/O devices while ensuring isolation and performance for virtualized systems.

Q16. Write short notes on Full device emulation?

Ans: Full device emulation involves creating virtual representations of physical hardware to mimic the behavior of actual devices. It allows software running in a virtual environment to interact with emulated devices as if they were real. This method ensures compatibility across different hardware configurations but might incur performance overhead due to the translation of device instructions. Full device emulation is commonly used in virtualization scenarios where direct hardware access is unavailable or impractical, ensuring broader system compatibility.

Q17. Write short notes on para-virtualization?

Ans: Para-virtualization architecture involves modifications made to guest operating systems (OSes) to collaborate more efficiently with the hypervisor or virtual machine monitor (VMM). This collaboration optimizes interactions between the OS and the hypervisor, reducing the overhead typically associated with full virtualization.

Q18. What are virtual clusters?

Ans: Virtual clusters are groups of interconnected virtual machines (VMs) or containers operating as a unified system. They simulate a cluster of physical computers or servers within a single environment. These clusters enable high availability, load balancing, and resource pooling, fostering resilience and scalability. Managed by orchestration tools like *Kubernetes* or *Docker Swarm*, virtual clusters efficiently allocate resources, ensuring seamless application deployment and management.

Q19. What is network migration?

Ans: Network migration involves transferring or transitioning an existing network infrastructure, data, configurations, or services from one environment to another. It includes moving systems, applications, or resources between different network architectures, platforms, or providers. This process aims to ensure minimal disruption, maintain functionality, and often involves planning, testing, and implementing strategies to seamlessly transition network components while preserving data integrity and system performance.

Q20. Define Virtual Machine Monitor (VMM)?

Ans: A *Virtual Machine Monitor* (VMM), also known as a hypervisor, is a software or firmware layer that creates and manages virtual machines (VMs) on a physical machine. It abstracts and partitions the underlying hardware, allowing multiple VMs to run concurrently, each with its own operating system and applications. The VMM provides isolation, resource allocation, and controls access to hardware resources for efficient virtualization.

Q21. Explain Host OS and Guest OS?

Ans: The *Host Operating System* (Host OS) is the primary operating system installed directly on the physical hardware of a computer or server. It manages system resources and provides services to support virtualization. A *Guest Operating System* (Guest OS) refers to an operating system running within a virtual machine (VM) on a host. Multiple guest OSes can run simultaneously on the same physical hardware, each isolated and independent from the others.

Q22. What is memory migration?

Ans: Memory migration involves transferring the contents of memory from one physical server or location to another without disrupting the operation of the system. This

process is commonly used in virtualized environments to move active memory pages from one host to another, enabling live migration of virtual machines (VMs) between physical servers while maintaining system uptime and minimizing downtime.

Q23. Explain the server consolidation?

Ans: Server consolidation involves combining multiple individual servers or workloads onto a single physical server or a reduced number of servers. This process optimizes resource utilization, reduces hardware footprint, and enhances efficiency by eliminating underutilized servers. By consolidating workloads, organizations can improve resource allocation, save space, reduce energy consumption, and streamline management, leading to cost savings and increased operational efficiency.

Long Answer Questions

Q1. What is virtualization? Discuss the types and benefits of virtualization.

Ans: Refer Section 7.1

Q2. Discuss the level of virtualization in cloud computing?

Ans: Refer Section 7.2

Q3. Explain in detail the virtualization Structures /tools and Mechanisms?

Ans: Refer Section 7.3

Q4. Discuss the difference between full virtualization and para-virtualization.

Ans: Refer Section 7.3

Q5. Draw and explain the Xen Architecture in detail.

Ans: Refer Section 7.3.1.1

Q6. Discuss the various techniques of full virtualization?

Ans: Refer Section 7.3.2

Q7. Explain in detail Virtualization of the followings:

(a) CPU

(b) Memory

(c) I/O Devices

Ans: Refer Section 7.4

Q8. Discuss the various types of CPU virtualization.

Ans: Refer Section 7.4.1

Q9. Explain the various methods used for memory virtualization in cloud computing.

Ans: Refer Section 7.4.2

Q10. What is meant by server consolidation? What are its types and benefits?

Ans: Refer Section 7.6.1

EXERCISE

1. Differentiate full virtualization and para-virtualization.
2. Draw and explain the architecture of para virtualization.
3. Write a short note on Binary Translation with Full Virtualization.
4. Explain the various components of Xen Architecture.
5. What is hypervisor? What is its importance in virtualization?

OOO