Unit 7
Virtualization

Structure of the Unit

1. Unit Outcomes
2. Definition,Need and Advantages
3. Types of Virtualization
4. Building blocks
5. Example
6. Self-Assessment Questions
7. Self-Assessment Activities
8. Multiple Choice Questions
9. Keys to Multiple Choice Questions
10. Summary of the Unit
11. Keywords
12. Recommended Resources for Further Reading

7.1 Unit Outcomes

After the successful completion of this unit, the student will be able to:
1. Define virtualization and articulate its significance in modern computing.
2. Identify the key advantages of virtualization technologies.
3. Differentiate between Type 0, Type 1, and Type 2 hypervisors.

7.2 Virtualization - Definition, Need and Advantages

Virtualization is a collection of diverse technologies designed to efficiently handle computing resources. It achieves this by introducing a software translation layer, commonly called an abstraction layer, which acts as a mediator between the software and the actual physical hardware. Think of it as a helpful bridge that makes communication between software and hardware smoother. The main job of virtualization is to transform tangible, physical resources into a more user-friendly and manageable form known as logical or virtual resources. This transformation simplifies the way users, applications, and management software interact with and utilize these resources. When virtualization is in play, users and applications can work with these virtual resources without needing to understand all the intricate details of the physical hardware underneath. It's like having a translator that converts the complex language of hardware into something everyone, including users and applications, can easily understand. This technology enables a seamless management and utilization of resources for users, applications, and management software situated above the abstraction layer. The beauty of virtualization lies in its ability to shield users and applications from the complexities of the physical details, offering a more straightforward and efficient way to make the most out of computing resources.

7.2.1 Definition of Virtualization:

Virtualization is a method that enables the sharing of a single physical instance of an application or resource among multiple organizations or tenants (customers). This sharing is achieved by giving a logical name to a physical resource and providing a reference or pointer to that specific resource when needed.

7.2.2 Need for Virtualization

1. Legacy Hardware Obsolescence:
   Challenges: Many organizations struggle with outdated hardware, making it difficult to run modern applications.
   Need for Virtualization: Virtualization provides a solution by emulating legacy hardware, allowing the continued use of applications designed for older systems.

2. Inefficiencies in Deployment Speed:
   Challenges: Traditional infrastructure faces slow deployment times, with new server setups taking weeks or longer.
   Need for Virtualization: Virtual Machines (VMs) enable rapid deployment within minutes, offering agility and responsiveness in dynamic business environments.

3. Resource Utilization and Versatility:
   Challenges: Hardware resources are often underutilized, and accommodating diverse applications on a single machine is limited.
   Need for Virtualization: Virtualization optimizes hardware usage, consolidates various application types, and enhances versatility for more efficient resource utilization.

4. Resource Consolidation:
   Challenges: High-capacity resources, like servers, may be underutilized when dedicated to a single application.
   Need for Virtualization: Virtualization addresses this by sharing a single resource among multiple applications, achieving heightened efficiency and cost-effectiveness.

5. Aggregating Resources:
   Challenges: Managing disparate resources independently can lead to inefficiencies and operational complexities.
   Need for Virtualization: Virtualization simplifies resource management by aggregating multiple resources into unified virtual entities, such as in storage virtualization.

6. Dynamic Resource Allocation:
   Challenges: Static allocation of hardware resources poses challenges in adapting to fluctuating

demands.
Need for Virtualization: Virtualization enables dynamic resource allocation, facilitating swift adjustments in response to changing workloads, improving load balancing, and ensuring fault tolerance.

7. Ease of Management:
   Challenges: Traditional infrastructure involves complex deployment and testing processes.
   Need for Virtualization: Virtual machines streamline the deployment and testing of software, providing a more straightforward and efficient management approach.

8. Enhanced Availability:
   Challenges: Ensuring high availability for physical servers involves significant costs and complexity.
   Need for Virtualization: Virtualization improves availability by clustering virtual machine hosts, offering higher availability at a reduced cost and complexity.

7.2.3 Advantages of Virtualization

Efficient Hardware Utilization:
The transition to virtualization reduces the need for physical hardware, optimizing hardware utilization for efficient operations. This leads to decreased maintenance costs, lower power consumption, and more flexibility in resource allocation, liberating organizations from dependency on specific hardware vendors.

Increased Availability:
Enhanced availability is a hallmark of virtualization platforms, providing features like live migration, fault tolerance, and distributed scheduling. These capabilities contribute to increased uptime, ensuring continuous operations even in the face of unplanned outages, ultimately boosting overall system reliability.

Disaster Recovery Simplified:
Disaster recovery becomes a streamlined process with virtualized servers, offering quick restoration through up-to-date snapshots. The ability to relocate virtual machines to alternative locations, including cloud providers, enhances the organization's ability to swiftly recover from disasters, ensuring a high success rate for the disaster recovery plan.

Energy Savings:
The shift from physical to virtual servers not only reduces hardware requirements but also leads to significant energy savings, aligning with sustainability goals. Lowering power consumption and cooling costs contributes to a more environmentally friendly and cost-effective data center operation.

Rapid Server Deployment:

Virtualization expedites server deployment through image cloning and virtual machine replication. This eliminates the need for time-consuming procurement processes, enabling organizations to respond quickly to changing requirements and reducing downtime for end-users.

Space Efficiency:

The adoption of virtualization results in substantial space efficiency gains, exemplified by the consolidation of servers in a data center. This not only optimizes physical space but also contributes to streamlined data center management and resource allocation.

Testing and Lab Environments:

Virtualization facilitates efficient testing and lab environments by providing the capability to revert to previous snapshots, minimizing the impact of errors. This ensures a seamless development process, allowing organizations to test and refine solutions before deploying them in live environments.

Transition to Cloud:

Virtualization eases the transition to cloud-based infrastructure, offering tools for exporting virtual servers. This flexibility enables organizations to seamlessly migrate their virtualized environments to cloud providers, aligning with the dynamic nature of modern IT infrastructure.

Service Division Possibilities:

The ability to segregate applications within a virtualized environment reduces the risk of service conflicts, enhancing overall server stability. Virtualization enables the creation of isolated environments, ensuring a higher level of reliability by preventing the impact of one application on another.

Increased Performance and Computing Capacity:

Virtualization, as a core technology in cloud-based infrastructures, enhances performance and computing capacity. It allows quick access to extensive computing resources, surpassing the capabilities of individual data centers and eliminating the need for substantial investments. This results in heightened performance, speed, and cost-effectiveness.

Underutilized Hardware and Software Resources:

Virtualization rectifies the underutilization of hardware and software by creating a separate environment that transparently utilizes these resources. It optimizes the efficiency of IT infrastructure by ensuring that computing capabilities are fully harnessed, leading to improved overall performance.

Lack of Space:

Virtualization, particularly in the context of server consolidation, alleviates space constraints in expanding data centers. This technique optimizes hardware resources, enabling organizations to manage the increasing demand for computing power and storage without the need for extensive physical space.

Greening Initiatives:
Virtualization plays a pivotal role in green initiatives by enabling the simultaneous operation of multiple operating system images on a single physical server. This consolidation reduces the physical server size, leading to environmental benefits through decreased energy consumption, lower e-waste footprint, and enhanced resource efficiency.

Rise of Administrative Costs:
Virtualization addresses the escalating costs associated with power consumption, cooling, and IT device expenses, along with the administrative challenges of managing an increasing number of servers. By optimizing resource utilization, virtualization contributes to a reduction in administrative staff costs, offering a more cost-effective and streamlined approach to IT management.

7.3 Types of Virtualization

The term "virtualization" encompasses various concepts, each contributing to enhanced efficiency and flexibility in computing environments. The different types of virtualization include:

1. Server Virtualization:
Server virtualization involves creating virtual instances of servers, allowing multiple operating systems to run on a single physical server. This eliminates the need for dedicated physical servers for different purposes.

2. Client & Desktop Virtualization:
Also known as desktop virtualization, this approach occurs on the user's end. It entails replacing traditional desktops with thin clients and leveraging datacenter resources to enhance manageability and accessibility.

3. Services and Application Virtualization:
This form of virtualization isolates applications and services from the underlying operating system, increasing compatibility and manageability. Technologies like Docker exemplify this approach by encapsulating applications independently.

4. Network Virtualization:
A crucial component of the virtualization infrastructure, network virtualization is particularly beneficial for visualizing servers. It facilitates the creation of multiple switching environments, VLANs (Virtual Local Area Networks), and NAT (Network Address Translation).

5. Storage Virtualization:
Predominantly used in data centers, storage virtualization enables the efficient allocation, creation, and deletion of storage resources across different hardware. This allocation occurs through network connections, with Storage Area Networks (SAN) being a leading technology in this domain.

6. Operating System Level Virtualization:

This form of virtualization introduces an abstraction layer between the traditional operating system (OS) and user applications. OS-level virtualization creates isolated containers on a single physical server, allowing for the efficient utilization of hardware and software resources in data centers.

In summary, virtualization spans various domains, from servers and desktops to networks and storage. It optimizes resource utilization, streamlines management, and enhances the overall flexibility of computing environments.

7.3.1 Concept of Virtualization:

The idea of virtualization involves the creation of a virtual machine on existing operating systems and hardware, a concept known as Hardware Virtualization. Virtual Machines (VMs) offer an environment that is logically separated from the actual underlying hardware. The implementation of virtual machines encompasses several key components. At the foundation is the host, which represents the underlying hardware system responsible for executing the virtual machines. Overlying this infrastructure is the Virtual Machine Manager (VMM), also known as a hypervisor. The VMM plays a pivotal role in creating and managing virtual machines by offering an interface that mirrors that of the host, with the exception of cases involving paravirtualization.

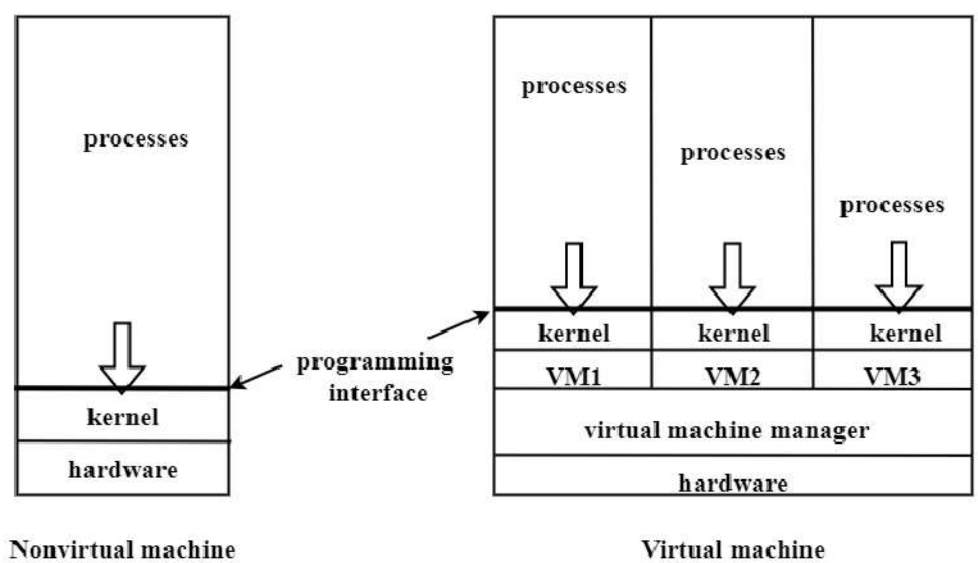

Figure 7.1:System models for virtual and not virtual machine.

The VMM facilitates the creation and operation of virtual machines, presenting each guest process with a virtual replica of the host system, as illustrated in Figure 7.1. Typically, these guest processes are actual operating systems. Consequently, a single physical machine has the capability to concurrently run multiple operating systems, each residing within its dedicated virtual machine. This architectural setup allows for enhanced flexibility and resource utilization, enabling diverse operating systems to coexist on the same hardware platform.

In essence, virtualization allows for the efficient sharing of computing resources by creating a virtual layer that abstracts the underlying physical hardware. This separation provides flexibility, enabling multiple users or organizations to utilize the same physical resources without interfering with one another. The hypervisor acts as a manager, overseeing and coordinating the various virtual machines on the host system. This approach enhances resource utilization, simplifies management, and contributes to the overall efficiency of the computing environment.

7.3.2 Hypervisors: Abstracting Hardware for Virtual Machines

Virtualization, functions as a form of abstraction, separating the physical hardware from virtual machines (VMs). The software responsible for this abstraction is the virtual machine monitor or hypervisor, which acts as a mediator between the operating system and the hardware. The hypervisor serves as a broker, managing and allocating resources on the physical host as requested by the VMs. A virtual machine, resembling a physical server, is configured with processors, RAM, storage, and network connectivity. Unlike physical servers, virtual servers only perceive the resources they are configured with, allowing multiple VMs to run concurrently on a single physical host without interference. The hypervisor facilitates communication between the VM and the physical server, translating and managing input/output. VM instances are defined by files, including a configuration file detailing attributes such as virtual processors, allocated RAM, I/O devices, and network interface cards. Storage, presented as virtual disks, is also defined in these files. The use of files simplifies functions like backup, as copying VM files creates a backup of the entire server, including the OS, applications, and hardware configuration. The rapid deployment of new VMs is often achieved through templates, providing standardized hardware and software settings. Templates streamline the creation of new VMs, allowing for quick provisioning with unique identifiers and configuration changes.Following are the important hypervisor functions:

1. Execution Management of VMs:
   - Scheduling VMs, managing virtual memory, and context switching.
   - Ensuring VM isolation and emulation of timer and interrupt mechanisms.
2. Devices Emulation and Access Control:
   - Emulating network and storage devices expected by native drivers in VMs.
   - Mediating access to physical devices for different VMs.
3. Execution of Privileged Operations:
   - Handling privileged operations invoked by guest OSs on behalf of the host hardware.
4. Management of VMs (Lifecycle Management):
   - Configuring and controlling VM states (Start, Pause, Stop).
5. Administration of Hypervisor Platform and Software:
   - Setting parameters for user interactions with the hypervisor host and software.

There are different types of hypervisor, let us now look at how the the types of hypervisor's differ.

Type 0 Hypervisor:

Type 0 hypervisors, known by various names such as "partitions" or "domains," have been present for many years as a hardware feature. This characteristic brings both advantages and disadvantages.

Operating systems do not require special modifications to leverage their features. The Virtual Machine Monitor (VMM) is encoded in firmware and loaded during boot time, subsequently loading guest images for each partition. The feature set of a type 0 hypervisor is typically smaller since it is implemented in hardware. For example, a system may be divided into four virtual systems, each with dedicated CPUs, memory, and I/O devices, simplifying implementation details for each guest.
However, challenges arise in I/O management, especially when there are insufficient I/O devices for all guests. In such cases, the hypervisor must manage shared access or allocate all devices to a control partition. Some advanced type 0 hypervisors can dynamically move physical CPUs and memory between running guests, requiring paravirtualization where guests are aware of virtualization and assist in its execution.

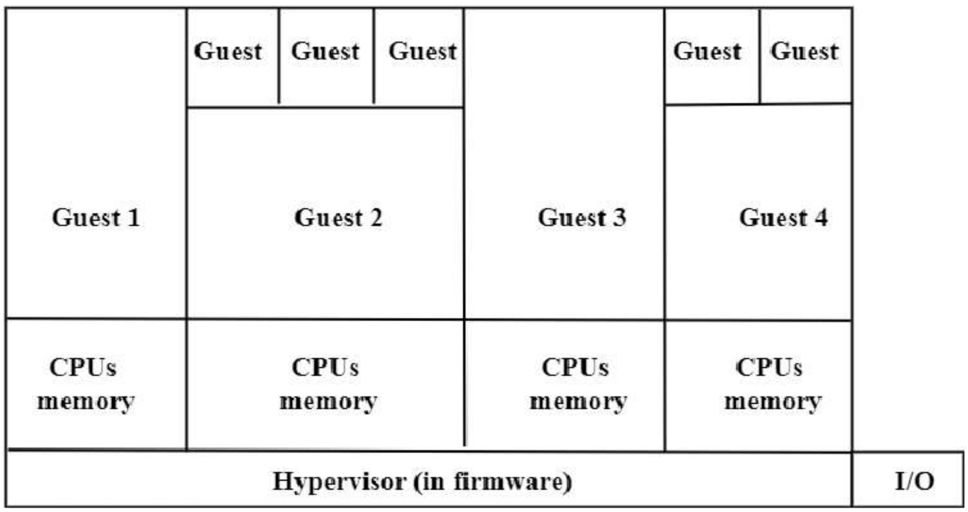

Figure 7.2: Type 0 hypervisor

Figure 7.2 shows Type 0 virtualization. Type 0 virtualization, being closely tied to raw hardware execution, should be considered separately from other methods. It allows for the running of multiple guest operating systems, each in its hardware partition, and these guests, running on raw hardware, can function as VMMs themselves, offering a unique virtualization-within-virtualization functionality not present in other types.

Type 1 Hypervisor:

Type 1 hypervisors are prevalent in company data centers, often referred to as the "data-center operating system." These are specialized operating systems running directly on hardware, managing guest operating systems. While they can run on standard hardware or type 0 hypervisors, they cannot operate on other type 1 hypervisors. Guests generally remain unaware that they are not running on native hardware. Figure 7.3 shows the type 1 hypervisor.

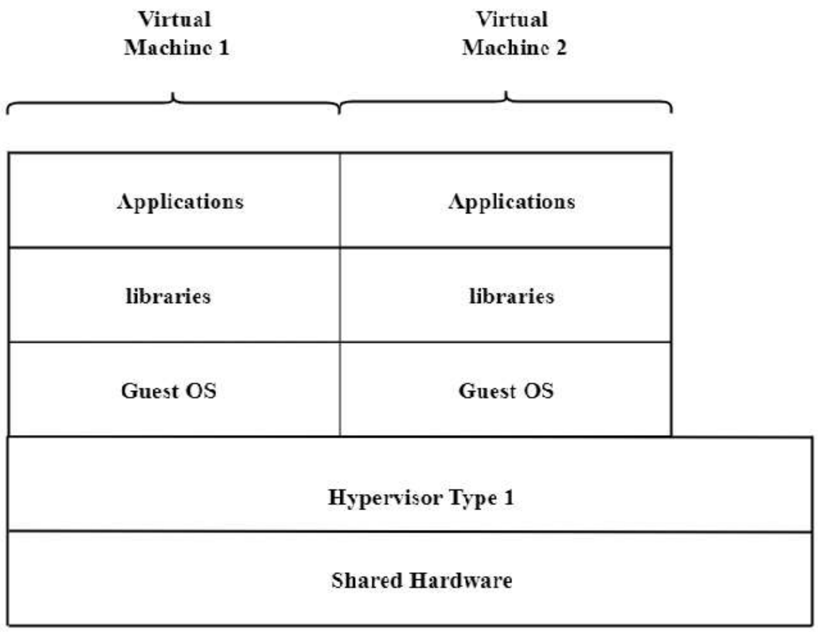

Figure 7.3 :Type 1 hypervisor

Running in kernel mode, type 1 hypervisors leverage hardware protection, utilizing multiple modes to enhance guest operating system control and performance. They provide essential operating system functionalities, including device drivers for the underlying hardware, CPU scheduling, memory management, I/O management, and security. Many type 1 hypervisors are commercial offerings, such as VMware ESX, or open source and hybrid variants, like Citrix XenServer.

Type 1 hypervisors enable data-center managers to control and manage operating systems and applications more effectively, facilitating consolidation of multiple systems onto fewer physical servers. This consolidation allows for dynamic adjustments and movement of guests and their applications, contributing to increased efficiency.

Another variant of type 1 hypervisors involves general-purpose operating systems with built-in VMM functionality. These operating systems, such as RedHat Enterprise Linux or Windows, perform regular duties while also providing a VMM for running other operating systems as guests, albeit with fewer virtualization features compared to dedicated type 1 hypervisors.

Type 2 Hypervisor:

Type 2 hypervisors are less relevant to those exploring operating systems, as they involve minimal operating system interaction. Operating at the application level, these virtual machine managers (VMMs) are treated as processes run and managed by the host. Type 2 hypervisors lack visibility into the virtualization happening within the VMM. Figure 7.4 shows the type 2 hypervisor.

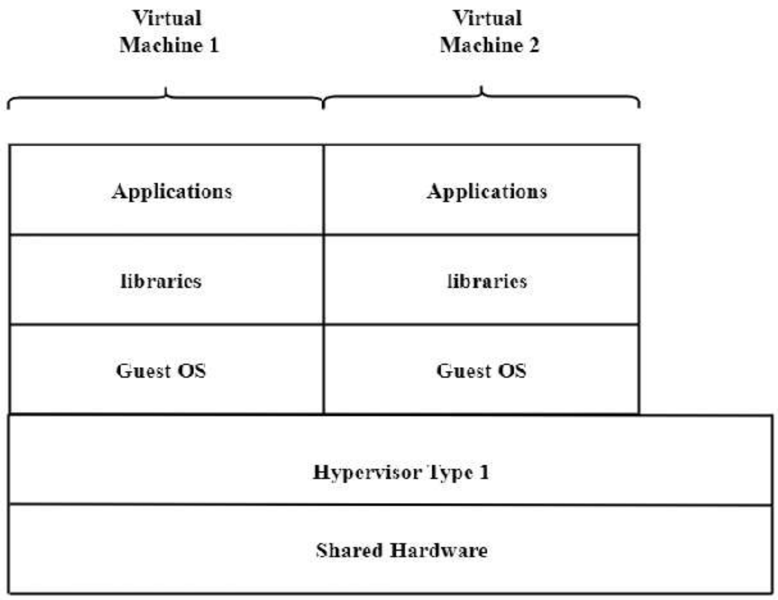

Figure 7.4 :Type 2 hypervisor

Despite running on various general-purpose operating systems without requiring modifications to the host OS, type 2 hypervisors have limitations. Users need administrative privileges to access hardware assistance features, and their overall performance tends to be poorer than type 0 or 1 hypervisors due to the added overhead of running a general-purpose OS alongside guest operating systems.
While limited in performance, type 2 hypervisors offer the advantage of running on different operating systems without necessitating changes to the host. This makes them suitable for educational purposes, enabling students to experiment with various operating systems on a single machine. For example, an Apple laptop can host multiple operating systems like Windows, Linux, Unix, and others concurrently for learning and experimentation.

Apart from the above methods there are some other ways of virtualization as listed below.

- **Paravirtualization** is a technique wherein the guest operating system undergoes modifications to collaboratively optimize performance with the Virtual Machine Monitor (VMM).
- Another approach is **programming-environment virtualization**, where VMMs refrain from virtualizing actual hardware and instead construct an optimized virtual system. This method is employed by platforms like Oracle Java and Microsoft.Net.
- **Emulators** represent a different strategy, enabling applications developed for one hardware environment to operate on a substantially different hardware setup, such as a distinct CPU type.
- Distinct from conventional virtualization, **application containment** focuses on providing virtualization-like features by isolating applications from the operating system. Examples include Oracle Solaris Zones, BSD Jails, and IBM AIX WPARs, which "contain" applications, enhancing security and manageability.

7.4 Building Blocks

The implementation of the virtual machine (VM) concept, while valuable, poses significant challenges. Creating an exact duplicate of the underlying machine requires substantial effort, particularly on dual-mode systems where only user mode and kernel mode are available. In this context, we explore the essential building blocks necessary for efficient virtualization, acknowledging that these building blocks are not mandatory for type 0 hypervisors. The feasibility of virtualization hinges on the capabilities offered by the CPU. If the CPU features are adequate, a Virtual Machine Monitor (VMM) can be developed to establish a guest environment. Otherwise, virtualization becomes unattainable. VMMs employ various techniques, such as trap-and-emulate and binary translation, to implement virtualization. These techniques, along with the required hardware support, are discussed in this section. A pivotal concept in virtualization involves the implementation of a virtual CPU (VCPU). Unlike a traditional CPU executing code, a VCPU represents the CPU state as perceived by the guest machine. Each guest is associated with a VCPU, maintaining the guest's current CPU state. During a guest context switch by the VMM, information from the VCPU is utilized to load the appropriate context, akin to a general-purpose operating system using a Process Control Block (PCB).

In dual-mode systems, where the virtual machine guest operates solely in user mode, the kernel runs in kernel mode. To address this, a virtual user mode and a virtual kernel mode are introduced within the virtual machine, both running in physical user mode. Transitions from virtual user mode to virtual kernel mode are essential, mirroring transitions from user mode to kernel mode on a physical machine. This transition is achieved through the trap-and-emulate method, where the VMM intercepts privileged instructions executed by the guest kernel and emulates the corresponding actions. Privileged instructions present a challenge in terms of performance, causing overhead and potentially slowing down the virtual machines. Various approaches have been adopted to mitigate this issue, such as allowing normal instructions to execute directly on the hardware and emulating only privileged instructions. For CPUs lacking a clear separation of privileged and nonprivileged instructions, like the Intel x86 CPU line, the binary translation technique proves essential. Binary translation dynamically converts instructions executed in virtual kernel mode into equivalent tasks, overcoming the limitations posed by special instructions that complicate trap-and-emulate methods.

The process of binary translation involves running guest instructions natively if the VCPU is in user mode and translating special instructions into a new set of instructions when the VCPU is in kernel mode. Caching is employed to enhance performance by storing translated instructions for subsequent executions, minimizing the need for repetitive translation. Memory management, specifically page tables, is a critical aspect of virtualization. Nested page tables (NPTs) are commonly used to represent the guest's page-table state, allowing the VMM to mirror changes made by the guest. However, the use of NPTs can introduce challenges such as increased TLB misses. Despite potential overhead, binary translation has proven effective in virtualizing Intel x86-based systems. Hardware assistance has played a crucial role in virtualization, with CPUs introducing features like Intel VT-x instructions for improved support. Memory management enhancements, such as AMD's RVI and Intel's EPT, eliminate the need for software NPTs, offering better control over paging and faster address translation.

Hardware assistance extends to I/O and interrupts, ensuring secure and efficient operation. CPUs with virtualization support feature interrupt remapping and DMA controllers with indirection to prevent interference between guests and enhance overall system stability and security.

7.5 Example

Despite the considerable advantages offered by virtual machines, they initially received limited attention for several years following their inception. However, in contemporary times, virtual machines have gained popularity as effective solutions for addressing system compatibility issues. In this section, we delve into two widely used virtual machines of today: VMware Workstation and the Java virtual machine. It becomes evident that these virtual machines can operate seamlessly on diverse operating systems, embracing the design principles discussed in previous chapters - ranging from simple layers and microkernels to modules and virtual machines. Importantly, these operating-system design methods are not mutually exclusive.

7.5.1 VMware

VMware Workstation, a prominent commercial application, serves as a noteworthy example of a Type 2 hypervisor. Functioning as an application on a host operating system, such as Windows or Linux, it facilitates the simultaneous execution of multiple independent virtual machines, each running its own guest operating system. The architecture of this system is illustrated in Figure 7.5, where Linux acts as the host operating system, while FreeBSD, Windows NT, and Windows XP function as guest operating systems. At the core of VMware lies the virtualization layer, which abstracts the physical hardware, creating isolated virtual machines with dedicated virtual CPUs, memory, disk drives, network interfaces, and more.

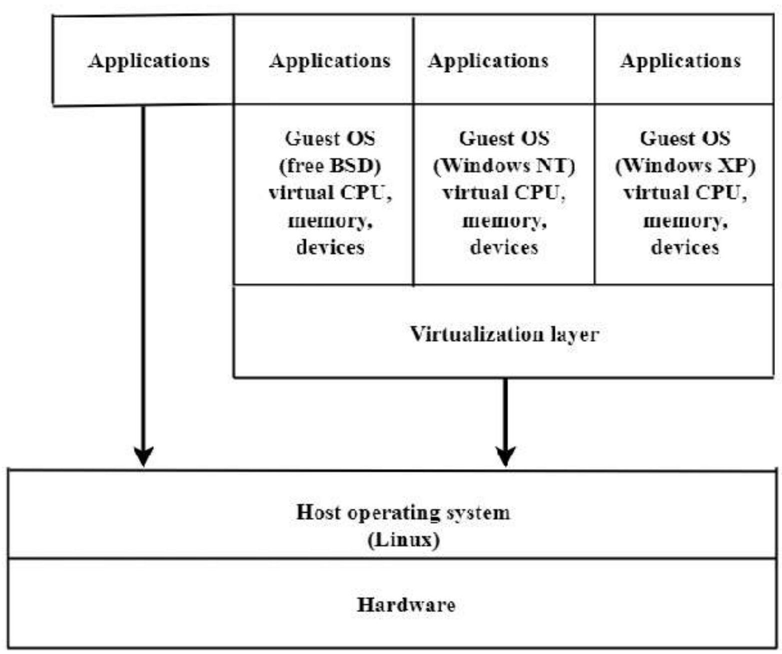

Figure 7.5: VMware Workstation architecture.

Notably, the physical disk managed by the guest is essentially a file within the host operating system's file system. This allows for the straightforward replication of an identical guest by simply copying the corresponding file. Such file manipulation provides essential flexibility, ensuring the guest's resilience against disasters at the original site and enabling the relocation of the entire guest system to another location. These scenarios exemplify how virtualization enhances the efficiency of system administration and optimizes the utilization of system resources.

7.6 Self-Assessment Questions

Q1. How does virtualization shield users and applications from the complexities of the underlying hardware? Explain [4 marks, L2]

Q2. Identify and elaborate on one specific challenge for which virtualization provides a solution, emphasizing the need for its adoption. [4 marks, L2]

Q3. Explain how virtualization enhances availability, and mention specific features that contribute to increased uptime. [5 marks, L2]

Q4. Explain the concept of space efficiency in virtualization, particularly in the context of server consolidation. [4 marks, L2]

Q5. How does virtualization contribute to efficient hardware utilization? Provide examples of benefits associated with optimizing hardware usage. [5 marks, L2]

Q6. Discuss the role of virtualization in easing the transition to cloud-based infrastructure, emphasizing its flexibility. [5 marks, L3]

Q7. Explain the relationship between virtualization and increased performance and computing capacity, especially in cloud-based infrastructures. [4 marks, L2]

Q8. What is the significance of hypervisors in virtualization, and how do they abstract hardware for virtual machines? [5 marks, L2]

Q9. What is storage virtualization. [4 marks, L2]

Q10. Explain the concept of server virtualization and how it eliminates the need for dedicated physical servers. [5 marks, L2]

Q11. Describe services and application virtualization, highlighting its impact on compatibility and manageability. [5 marks, L2]

Q12. What is paravirtualization, and how does it optimize performance by modifying the guest operating system? . [3 marks, L2]

7.7 Self-Assessment Activities

A1. Provide a labeled diagram of the virtualization components (e.g., host, VMM, virtual machines). Ask your friends to identify and explain each component's role in the virtualization process.

A2. A company is considering implementing desktop virtualization to enhance manageability and accessibility for its employees.Explain how desktop virtualization can improve manageability and accessibility for end-users, and what challenges might the company face during implementation.

A3. A large data center is dealing with storage allocation challenges and wants to efficiently allocate, create, and delete storage resources across various hardware.Outline how storage virtualization can address the challenges of storage allocation in a data center and provide efficient solutions.

A4. Create a comparative chart highlighting the differences between Type 0, Type 1, and Type 2 hypervisors. Include factors such as implementation, performance, and use cases.

A5. Explore VMware Workstation or a similar Type 2 hypervisor. Install a virtual machine, configure settings, and explain how virtualization is implemented in such software.

7.8 Multiple-Choice Questions

Q1. What is the primary purpose of virtualization in computing?[1 mark, L1]

A. Enhancing hardware complexity
B. Streamlining communication between software and hardware
C. Introducing more intricate details
D. Increasing dependencies on physical resources

Q2. What is the primary role of a hypervisor in virtualization? [1 mark, L1]

A. Managing physical hardware
B. Creating virtual machines
C. Handling network configurations
D. Running user applications

Q3. What type of virtualization involves creating virtual instances of servers on a single physical server? [1 mark, L1]

A. Server Virtualization
B. Client & Desktop Virtualization
C. Network Virtualization
D. Storage Virtualization

Q4. Which type of hypervisor runs directly on hardware and is often referred to as the "data-center operating system"? [1 mark, L1]

A. Type 0 Hypervisor
B. Type 1 Hypervisor
C. Type 2 Hypervisor
D. Type 3 Hypervisor

Q5. Which form of virtualization involves isolating applications and services from the underlying operating system? [1 mark, L1]

A. Server Virtualization
B. Client & Desktop Virtualization
C. Services and Application Virtualization
D. Storage Virtualization

Q6. In the context of virtualization, what is paravirtualization? [1 mark, L1]

A. Creating virtual servers.
B. Emulating network and storage devices
C. Modifying the guest OS to optimize performance
D. Isolating applications from the OS

Q7. What building block is critical for virtualization and represents the CPU state as perceived by the guest machine? [1 mark, L1]

A. Memory management.
B. Virtual Machine Monitor (VMM).
C. Virtual CPU (VCPU).
D. Nested Page Tables (NPTs)

Q8. What is a key advantage of virtualization related to energy consumption? [1 mark, L1]

A. Increased power consumption.
B. Lowering hardware utilization.
C. Alignment with sustainability goals
D. Dependence on specific hardware vendors

Q9. Which challenge does virtualization address regarding server deployment speed?[1 mark, L1]

A. Hardware obsolescence.
B. Rapid deployment times
C. Resource consolidation
D. Static resource allocation

Q10. How does virtualization impact space efficiency in data centers? [1 mark, L1]

A. Increases physical server size.
B. Decreases server consolidation.
C. Optimizes physical space through consolidation
D. Requires extensive physical space

Q11. In the context of resource consolidation, what does virtualization aim to achieve? [1 mark, L1]

A. Dedicated resources for each application.
B. Increased hardware diversity.
C. Underutilization of resources
D. Sharing a single resource among multiple applications

Q12. How does virtualization simplify resource interaction for users and applications? [1 mark, L1]

A. By revealing intricate hardware details
B. By making hardware language more complex
C. By introducing hardware dependencies
D. By transforming physical resources into logical or virtual resources

Q13. What is the primary function of a Type 2 Hypervisor?

A. Running directly on hardware
B. Operating at the application level
C. Managing hardware resources
D. Running multiple guests in their own hardware partitions

Q14. What is a key advantage of VMware Workstation as a Type 2 hypervisor?

A. Direct hardware access
B. Simultaneous execution of multiple virtual machines
C. Operation as a data-center operating system
D. Running on multiple hardware types without modification

7.9 Keys to Multiple-Choice Questions

Q1. Streamlining communication between software and hardware (B)
Q2. Creating virtual machines (B)
Q3. Server Virtualization (A)
Q4. Type 1 Hypervisor (B)
Q5. Services and Application Virtualization (C)
Q6. Modifying the guest OS to optimize performance(C)
Q7. Virtual CPU (VCPU). (C)
Q8. Alignment with sustainability goals. (C)
Q9. Rapid deployment times. (B)
Q10. Optimizes physical space through consolidation. (C)
Q11. Sharing a single resource among multiple applications. (D)
Q12. By transforming physical resources into logical or virtual resources. (D)
Q13. Operating at the application level. (B)
Q14. Simultaneous execution of multiple virtual machines (B)

7.10 Summary of the Unit

Virtualization is a technique employed to furnish a guest system with a replica of the underlying hardware of a host system. This enables multiple guests to operate on a single system, with each guest perceiving itself as the native operating system exercising complete control over the entire system. Initially, virtualization emerged as a means for IBM to segregate users and allocate individual execution environments on mainframes. Over time, driven by advancements in system and CPU performance, coupled with innovative software techniques, virtualization has evolved into a ubiquitous feature in data centers and personal computers alike. The growing popularity of virtualization has prompted CPU designers to incorporate features specifically designed to enhance virtualization support, and this trend is expected to persist, leading to continuous improvements in both virtualization software and hardware support. Type 0 virtualization is implemented directly in the hardware, necessitating modifications to the operating system to ensure seamless operation. These modifications exemplify the concept of paravirtualization, where the operating system is not oblivious to virtualization but, instead, undergoes alterations with added features and modified algorithms to enhance the performance and functionality of virtualization. In the case of Type 1 virtualization, a host Virtual Machine Monitor (VMM) plays a central role in providing the environment and features required to create, run, and terminate guest virtual machines. Each guest encompasses all the software components typically associated with a full-fledged native system, including the operating system, device drivers, applications, user accounts, and more. On the other hand, Type 2 hypervisors are essentially applications that operate within other operating systems, unaware of the ongoing virtualization. These hypervisors, also known as hosted hypervisors, lack direct hardware or host support. Consequently, they are compelled to execute all virtualization activities within the confines of a process, relying solely on the capabilities of the host operating system.

7.11 Keywords

Virtualization,Hypervisor,Paravirtualization,Virtual Machine Manager,Emulators,VMware

7.12 Recommended Learning Resources

[1] Silberschatz, A., Galvin, P., & Gagne, G. (2005). Operating System Concepts, 7th ed., Hoboken.

[2] William stalling. Operating Systems: Internal and design principles, 7th edition PHI

[3] D.M. Dhamdhere. Operating Systems: A concept-based Approach, 2nd Edition, TMH