CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | | |

Internal Assessment Test 2– Feb. 2024

Sub:	Cloud Computing							Sub Code:	22MCA332
Date:	16/2/2024	Duration:	90 min's	Max Marks:	50	Sem:	III	Branch:	MCA

Note : Answer FIVE FULL Questions, choosing ONE full question from each Module

		MARKS	OBE	
			CO	RBT
	PART I			
1	Different Phenomena responsible for renewed interest in vitalization	[10]	CO3	L2
	OR			
2	Discuss in detail about characteristics of virtualized solutions with the required diagrams.	[10]	CO3	L2
	PART II			
3	Discuss the machine reference model of execution virtualization	[10]	CO3	L3
	OR			
4	List and discuss different types of hardware virtualization techniques	[10]	CO3	L3
	PART III			
5	What are the problems of virtualization (disadvantages) and how to solve them?	[10]	CO3	L3
	OR			
6	Discuss and clarify in detail about need for a VMware full virtualization reference model	[10]	CO3	L2
	PART IV			
7	Explain in detail the Popek and Goldberg theorem in virtualization	[10]	CO3	L3
	OR			
8	Explain Programming language level virtualization	[10]	CO3	L2
	PARTV			
9	With a neat diagram, explain Xen-Paravirtualization.	[10]	CO3	L3
	OR			
10	What is Hypervisor? With a neat diagram explain the types of hypervisor.	[10]	CO3	L2

Q1) Different Phenomena responsible for renewed interest in vitalization

Virtualization technologies have gained renewed interested recently due to the confluence of severalphenomena:

(a) Increased performance and computing capacity.

The high-end side of the PC market, where supercomputers can provide immense compute power that can accommodate the execution of hundreds or thousands of virtual machines.

(b) Underutilized hardware and software resources.

Hardware and software underutilization is occurring due to (1) increased performance and computing capacity, and (2) the effect of limited or sporadic use of resources.

Computers today are so powerful that in most cases only a fraction of their capacity is used by an application or the system. Using these resources for other purposes after hours could improve the efficiency of the IT infrastructure.

(c) Lack of space.

Companies such as Google and Microsoft expand their infrastructures by building data centers as large as football fields that are able to host thousands of nodes. Although this is viable for IT giants, in most cases enterprises cannot afford to build another data center to accommodate additional resource capacity. This condition, along with hardware underutilization, has led to the diffusion of a technique called server consolidation

(d) Greening initiatives.

Maintaining a data center operation not only involves keeping servers on, but a great deal of energy is also consumed in keeping them cool. Infrastructures for cooling have a significant impact on the carbon footprint of a data center. Hence, reducing the number of servers through server consolidation will definitely reduce the impact of cooling and power consumption of a data center. Virtualization technologies can provide an efficient way of consolidating servers.

(e) Rise of administrative costs.

The increased demand for additional capacity, which translates into more servers in a data

center, is also responsible for a significant increment in administrative costs. Computers—in particular, servers—do not operate all on their own, but they require care and feeding from system administrators.

These are labor-intensive operations, and the higher the number of servers that have to be managed, the higher the administrative costs. Virtualization can help reduce the number of required servers for a given workload, thus reducing the cost of the administrative personnel.

Q2) Discuss in detail about characteristics of virtualized solutions with the required diagrams.

The characteristics of virtualized solutions are:

1 Increased security

2 Managed executions

3 Portability

1. Increased security

The virtual machine represents an emulated environment in which the guest is executed. All the operations of the guest are generally performed against the virtual machine, which then translates and applies them to the host. This level of indirection allows the virtual machine manager to control and filter the activity of the guest, thus preventing some harmful operations from being performed. For example, applets downloaded from the Internet run in a sandboxed 3 version of the Java Virtual Machine (JVM), which provides them with limited access to the hosting operating system resources. Both the JVM and the .NET runtime provide extensive security policies for customizing the execution environment of applications.

2 Managed executions

Virtualization of the execution environment not only allows increased security, but a wider range of features also can be implemented. In particular, sharing, aggregation, emulation, and isolation

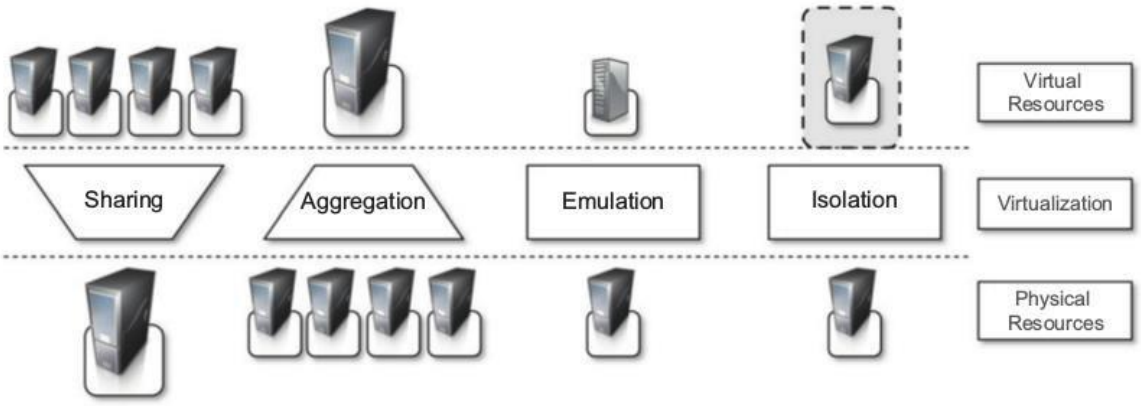

FIGURE 3.2

Functions enabled by managed execution.

are the most relevant features (see Figure 3.2).

(a) Sharing - Virtualization allows the creation of a separate computing environments within the same host. In this way it is possible to fully exploit the capabilities of a powerful guest, which would otherwise be underutilized.

(b)Aggregation - Not only is it possible to share physical resource among several guests, but virtualization also allows aggregation, which is the opposite process. A group of separate hosts can be tied together and represented to guests as a single virtual host.

(c)Emulation - Guest programs are executed within an environment that is controlled by the virtualization layer, which ultimately is a program. This allows for controlling and tuning the environment that is exposed to guests. For instance, a completely different environment with respect to the host can be emulated, thus allowing the execution of guest programs requiring specific characteristics that are not present in the physical host.

(d) Isolation - Virtualization allows providing guests—whether they are operating systems, applications, or other entities—with a completely separate environment, in which they are executed. The guest program performs its activity by interacting with an abstraction layer, which provides access to the underlying resources.

3 Portability

The concept of portability applies in different ways according to the specific type of virtualization considered. In the case of a hardware virtualization solution, the guest is packaged into a virtual image that, in most cases, can be safely moved and executed on top of different virtual machines.

In the case of programming-level virtualization, as implemented by the JVM or the .NET runtime, the binary code representing application components (jars or assemblies) can be run without any recompilation on any implementation of the corresponding virtual machine. This makes the application development cycle more flexible and application deployment very straightforward: One version of the application, in most cases, is able to run on different platforms with no changes.

Q3) Discuss the machine reference model of execution virtualization

Execution virtualization includes all techniques that aim to emulate an execution environment thatis separate from the one hosting the virtualization layer.

All these techniques concentrate their interest on providing support for the execution of programs, whether these are the operating system, a binary specification of a program compiled against an abstract machine model, or an application.

Modern computing systems can be expressed in terms of the reference model described in Figure3.4.

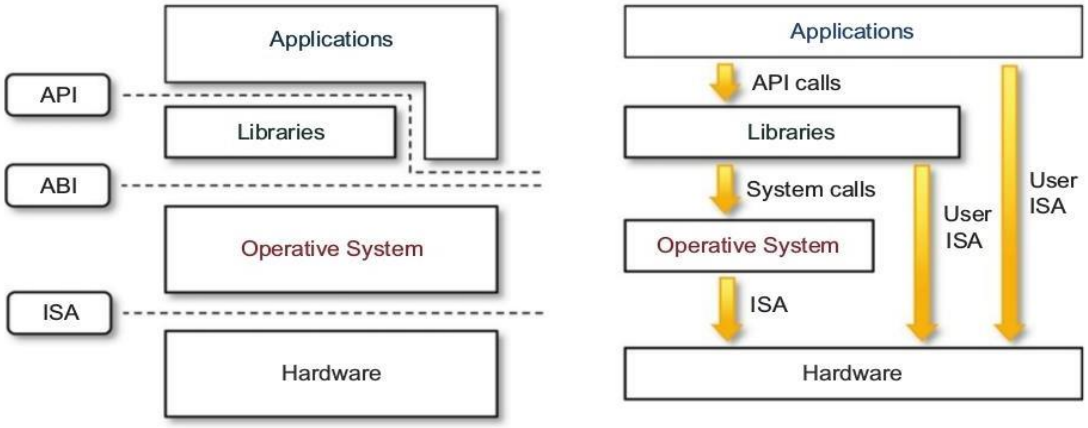

FIGURE 3.4

A machine reference model.

- At the bottom layer, the model for the hardware is expressed in terms of the **Instruction Set Architecture (ISA)**, which defines the instruction set for the processor, registers, memory, and interrupt management.
- **ISA** is the interface between hardware and software, and it is important to the operating system (OS) developer (System ISA) and developers of applications that directly manage theunderlying hardware (User ISA).
- The **application binary interface (ABI)** separates the operating system layer from the applications and libraries, which are managed by the OS.
- ABI covers details such as low-level data types, alignment, and call conventions and defines a format for executable programs.
- The highest level of abstraction is represented by the **application programming interface (API)**, which interfaces applications to libraries and/or the underlying operating system.

The **instruction set (ISA)** exposed by the hardware has been divided into two different parts as **privileged** and **nonprivileged** instructions.

- **Privileged instructions** are those that are executed under specific **restrictions** and are mostly used for **sensitive operations**, which expose (behavior-sensitive) or modify (control-

sensitive) the privileged state.

- **Nonprivileged instructions** are those instructions that can be used **without interfering** with other tasks because they **do not access shared resources**.

A possible implementation features a hierarchy of privileges (see Figure 3.5) in the form of ring-based security: Ring 0, Ring 1, Ring 2, and Ring 3;

- Ring 0 is in the most privileged level and Ring 3 in the least privileged level.
- **Ring 0** is used by the **kernel of the OS** (Supervisor)**.**
- **Rings 1 and 2** are used by the **OS-level services** (hypervisor) and
- **Ring 3** is used by the user.

Recent systems support only two levels, with Ring 0 for supervisor mode and Ring 3 for user mode.

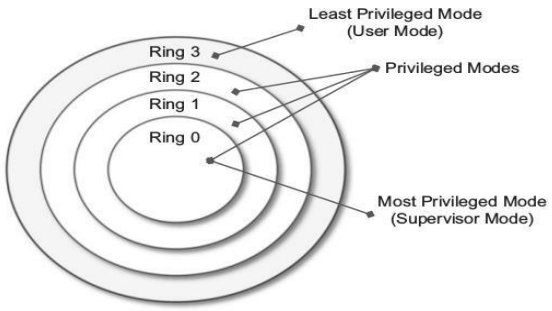

FIGURE 3.5
Security rings and privilege modes.

Q4) List and discuss different types of hardware virtualization techniques

Hardware virtualization techniques (System Level techniques)

- Hardware-assisted virtualization
- Full virtualization
- Paravirtualization
- Partial virtualization

- **Hardware-assisted virtualization (Hardware – Processors)**

  This term refers to a scenario in which the hardware provides architectural support for building a virtual machine manager able to run a guest operating system in complete isolation.

- **Full virtualization (Running Operating System – no modified OS)**

  Full virtualization refers to the ability to run a program, most likely an operating system, directly on top of a virtual machine and without any modification, as though it were run on the raw hardware.

- **Paravirtualization (Thin virtual machine - modified OS)**

  Paravirtualization techniques expose a software interface to the virtual machine that is slightly modified from the host and, therefore, guests need to be modified. The aim of paravirtualization is to provide the capability to demand the execution of performance-critical operations directly on the host

- **Partial virtualization**

  Partial virtualization provides a partial emulation of the underlying hard- ware, thus not allowing the complete execution of the guest operating system in complete isolation. Partial virtualization allows many applications to run transparently, but not all the features of the operating system can be supported.

Q5) What are the problems of virtualization (disadvantages) and how to solve them?

(a) Performance degradation

Performance is one of the major concerns in using virtualization technology. Since virtualization interposes an abstraction layer between the guest and the host, the guest can experience increased latencies (delays).

For instance, in the case of hardware virtualization, where the intermediate emulates a bare machine on top of which an entire system can be installed, the causes of performance

degradation can be traced back to the overhead introduced by the following activities:

- Maintaining the status of virtual processors
- Support of privileged instructions (trap and simulate privileged instructions)
- Support of paging within VM
- Console functions

(b) Inefficiency and degraded user experience

Virtualization can sometime lead to an inefficient use of the host. Some of the specific features of the host cannot be exposed by the abstraction layer and then become inaccessible. In the case of hardware virtualization, this could happen for device drivers: The virtual machine can sometime simply provide a default graphic card that maps only a subset of the features available in the host. In the case of programming-level virtual machines, some of the features of the underlying operating systems may become inaccessible unless specific librariesare used.

(c) Security holes and new threats

Virtualization opens the door to a new and unexpected form of phishing. The capability of emulating a host in a completely transparent manner led the way to malicious programs that are designed to extract sensitive information from the guest. The same considerations can be made for programming-level virtual machines: Modified versions of the runtime environment can access sensitive information or monitor the memory locations utilized by guest applications while these are executed.

Q6) Discuss and clarify in detail about need for a VMware full virtualization reference model

VMware: Full Virtualization

In full virtualization primary hardware is replicated and made available to the guest operating system, which executes unaware of such abstraction and no requirements to modify. Technology of VMware is based on the key concept of Full Virtualization. Either in desktop

environment, with the help of type-II hypervisor, or in server environment, through type-I hypervisor, VMware implements full virtualization. In both the cases, full virtualization is possible through the direct execution for non-sensitive instructions and binary translation for sensitive instructions or hardware traps, thus enabling the virtualization of architecture like x86.

(a) Full Virtualization and Binary Translation

VMware is widely used as it tends to virtualize x86 architectures, which executes unmodified on-top of their hypervisors. With the introduction of hardware-assisted virtualization, full virtualization is possible to achieve by support of hardware. But earlier, x86 guest operating systems unmodified in a virtualized environment could be executed only with the use of dynamic binary translation.

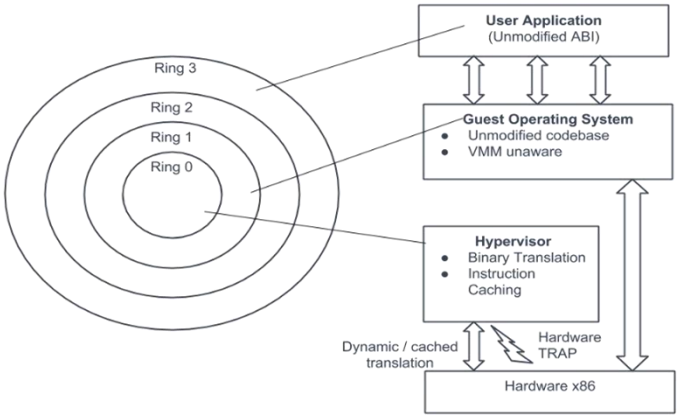

Figure – Full Virtualization Reference Model

The major benefit of this approach is that guests can run unmodified in a virtualized environment, which is an important feature for operating system whose source code does not exist. Binary translation is portable for full virtualization. As well as translation of instructions at runtime presents an additional overhead that is not existed in other methods like

paravirtualization or hardware-assisted virtualization. Contradict, binary translation is only implemented to a subset of the instruction set, while the others are managed through direct execution on the primary hardware. This depletes somehow the impact on performance of binary translation.

Advantages of Binary Translation –

1. This kind of virtualization delivers the best isolation and security for Virtual Machine.
2. Truly isolated numerous guest OS can execute concurrently on the same hardware.
3. It is only implementation that needs no hardware assist or operating system assist tovirtualize sensitive instruction as well as privileged instruction.

Disadvantages of Binary Translation –

1. It is time consuming at run-time.
2. It acquires a large performance overhead.
3. It employs a code cache to stock the translated most used instructions to enhance theperformance, but it increases memory utilization along with the hardware cost.
4. The performance of full virtualization on the x86 architecture is 80 to 95 percent that ofthe Host machines.

Q7) Explain in detail the Popek and Goldberg theorem in virtualization

Popek and Goldberg provided a classification of the instruction set and proposed three theorems that define the properties that hardware instructions need to satisfy in order to efficiently supportvirtualization.

THEOREM 1

For any conventional third-generation computer, a VMM may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged

This theorem establishes that all the instructions that change the configuration of the system resources should generate a trap in user mode and be executed under the control of the virtual machine manager.

THEOREM 2

A conventional third-generation computer is recursively virtualizable if:

- It is virtualizable and
- A VMM without any timing dependencies can be constructed for it.

Recursive virtualization is the ability to run a virtual machine manager on top of another virtual machine manager. This allows nesting hypervisors as long as the capacity of the underlying resources can accommodate that. Virtualizable hardware is a prerequisite to recursive virtualization.

THEOREM 3

A hybrid VMM may be constructed for any conventional third-generation machine in which the set of user-sensitive instructions is a subset of the set of privileged instructions.

There is another term, hybrid virtual machine (HVM), which is less efficient than the virtual machine system. In the case of an HVM, more instructions are interpreted rather than being executed directly. All instructions in virtual supervisor mode are interpreted. Whenever there is an attempt to execute a behavior-sensitive or control-sensitive instruction, HVM controls the execution directly or gains the control via a trap. Here all sensitive instructions are caught by HVM that are simulated.

Q8) Explain Programming language level virtualization

- Programming language-level virtualization is mostly used to achieve ease of deployment of applications, managed execution, and portability across different platforms and operating systems. It consists of a virtual machine executing the byte code of a program, which is the result of the compilation process. Compilers implemented and used this technology to produce a binary format representing the machine code for an abstract architecture.
- The characteristics of this architecture vary from implementation to implementation. Generally these virtual machines constitute a simplification of the underlying hardware instruction set and provide some high-level instructions that map some of the features of the languages compiled for them. At runtime, the byte code can be either interpreted or compiled on the fly or jitted against the underlying hardware instruction set.
- Programming language-level virtualization has a long trail in computer science history and originally was used in 1966 for the implementation of Basic Combined Programming Language (BCPL), a

language for writing compilers and one of the ancestors of the C programming language. Other important examples of the use of this technology have been the UCSD Pascal and Smalltalk.

- The Java virtual machine was originally designed for the execution of programs written in the Java language, but other languages such as Python, Pascal, Groovy, and Ruby were made available. The ability to support multiple programming languages has been one of the key elements of the Common Language Infrastructure (CLI), which is the specification behind .NET Framework.
- Currently, the Java platform and .NET Framework represent the most popular technologies for enterprise application development. Both Java and the CLI are stack-based virtual machines: The reference model of the abstract architecture is based on an execution stack that is used to perform operations.
- Both Java and the CLI are stack-based virtual machines: The reference model of the abstract architecture is based on an execution stack that is used to perform operations. The byte code generated by compilers for these architectures contains a set of instructions that load operands on the stack, perform some operations with them, and put the result on the stack.
- The main advantage of programming-level virtual machines, also called process virtual machines, is the ability to provide a uniform execution environment across different platforms. Programs compiled into byte code can be executed on any operating system and platform for which a virtual machine able to execute that code has been provided.

Q9) With a neat diagram, explain Xen-Paravirtualization.

Xen is an open-source **hypervisor** based on **paravirtualization**. It is the most popular application of paravirtualization. Xen has been extended to compatible with full virtualization using hardware-assisted virtualization. It enables high performance to execute **guest operating system**.

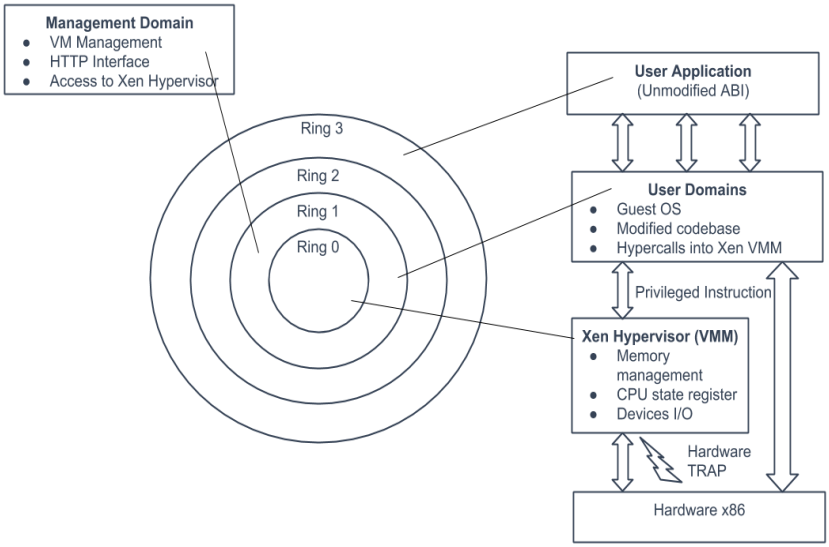

Figure – Xen Architecture and Guest OS Management

Above figure describes the Xen Architecture and its mapping onto a **classic x86 privilege model**. A Xen based system is handled by Xen hypervisor, which is executed in the most privileged mode and maintains the access of guest operating system to the basic hardware. Guest operating systems are run between domains, which represents virtual machine instances.

In addition, **particular control software**, which has **privileged access** to the host and handles all other guest OS, runs in a special domain called Domain 0. This the only one loaded once the virtual machine manager has fully booted and hosts an HTTP server that delivers requests for virtual machine creation, configuration, and termination.

This component establishes the primary version of a shared virtual machine manager (VMM), which is a necessary part of Cloud computing system delivering Infrastructure-as-a-Service (IaaS) solution

Almost all the frequently used Operating system uses only two levels i.e., Ring0 for the Kernel code and Ring 3 for user application and non-privilege OS program. This provides a chance to the Xen to implement paravirtualization. This enables Xen to control unchanged the Application Binary Interface (ABI) thus allowing a simple shift to Xen- virtualized solutions, from an application perspective.

Paravirtualization demands the OS codebase be changed, and hence all operating systems

cannot be referred to as guest OS in a Xen-based environment. This condition holds where hardware-assisted virtualization cannot be free, which enables to run the hypervisor in Ring 1 and the guest OS in Ring 0. Hence, Xen shows some limitations in terms of legacy hardware and in terms of legacy OS.

Pros:

a) Xen server is developed over open-source Xen hypervisor and it uses a combination of hardware-based virtualization and paravirtualization. This tightly coupled collaboration between the operating system and virtualized platform enables the system to develop lighter and flexible hypervisor that delivers their functionalities in an optimized manner.

b) Xen supports balancing of large workload efficiently that capture CPU, Memory, disk input-output and network input-output of data. It offers two modes to handle this workload: Performance enhancement, and For handling data density.

c) It also comes equipped with a special storage feature that we call Citrix storage link. Which allows a system administrator to uses the features of arrays from Giant companies- Hp, Netapp,Dell Equal logic etc.

d) It also supports multiple processors, live migration one machine to another, physical server to virtual machine or virtual server to virtual machine conversion tools, centralized multi server management, real time performance monitoring over window and Linux.

Cons:

a) Xen is more reliable over linux rather than on window.

b) Xen relies on 3rd-party component to manage the resources like drivers, storage, backup,recovery & fault tolerance.

c) Xen deployment could be a burden some on your Linux kernal system as time passes.

d) Xen sometimes may cause increase in load on your resources by high input-output rate and may cause starvation of other Vm's.

Q10) What is Hypervisor? With a neat diagram explain the types of hypervisor

Hypervisors

A fundamental element of hardware virtualization is the **hypervisor**, or virtual machine manager (**VMM**). It recreates a hardware environment in which guest operating systems are installed. There are two major types of hypervisors: Type I and Type II (see Figure 3.7).

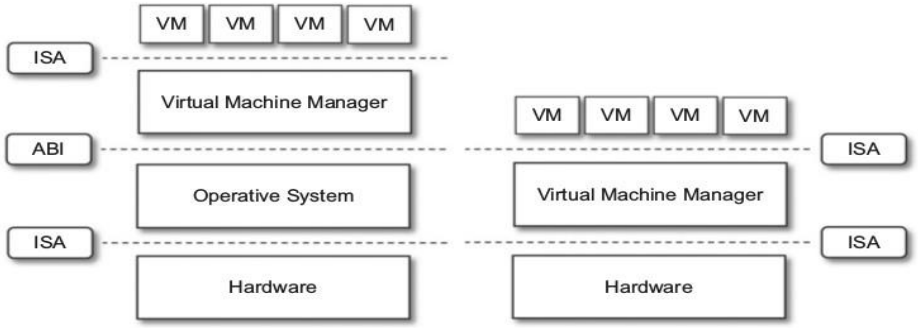

FIGURE 3.7
Hosted (left) and native (right) virtual machines. This figure provides a graphical representation of the two types of hypervisors.

- **Type I hypervisors (native virtual machine)** It run directly on top of the hardware. Therefore, they take the place of the operating systems and interact directly with the ISA interface exposed by the underlying hardware, and they emulate this interface to allow the management of guest operating systems. This type of hypervisor is also called a **native virtual machine** since it runs natively on hardware.
- **Type II hypervisors (hosted virtual machine)** It requires the support of an operating system to provide virtualization services. This means that they are programs managed by the operating system, which interact with it through the ABI and emulate the ISA of virtual hardware for guest operating systems. This type of hypervisor is also called a **hosted virtual machine** since it is hosted within an operating system.
- **Virtual machine manager (VMM)** is internally organized as described in Figure 3.8.

  Three main modules, **dispatcher**, **allocator**, and **interpreter**, coordinate their activity to emulate the underlying hardware.

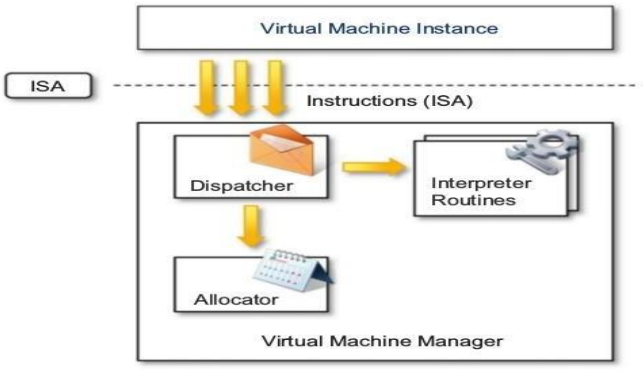

FIGURE 3.8
A hypervisor reference architecture.

- The **dispatcher** constitutes the entry point of the monitor and reroutes the

instructionsissued by the virtual machine instance to one of the two other modules.

- The **allocator** is responsible for deciding the **system resources to be provided to the VM**.
- The **interpreter module** consists of interpreter routines. These are executed whenever avirtual machine executes a privileged instruction.

The design and architecture of a virtual machine manager, together with the underlying hardware design of the host machine, determine the full realization of hardware virtualization, where a guest operating system can be transparently executed on top of a VMM as though it were run on the underlying hardware.