Virtualization

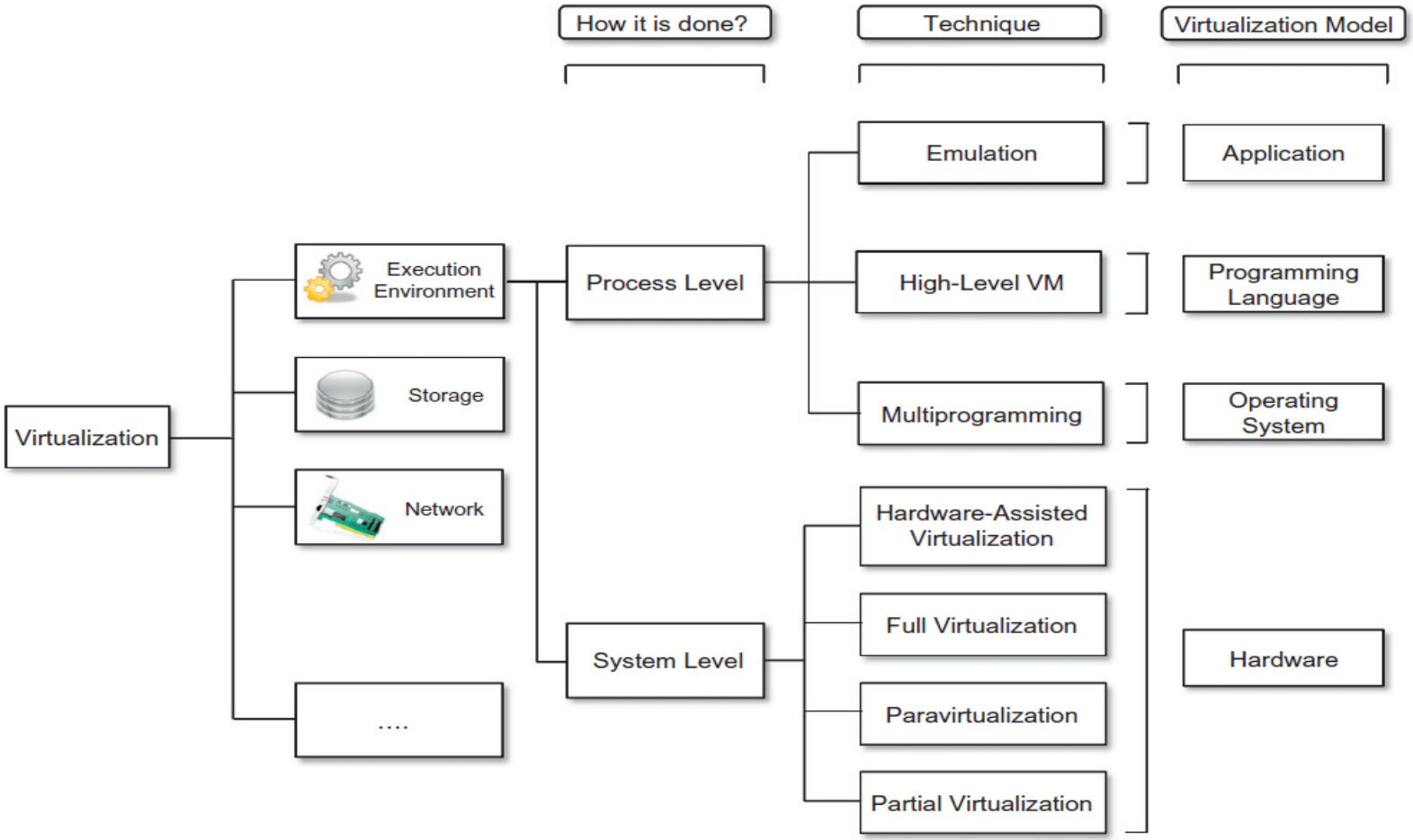
Dr. Prasenjit Chanak
Assistant Professor

Department of Computer Science and Engineering
Indian Institute of Technology (BHU), Varanasi-221005

Taxonomy of Virtualization Techniques

- Virtualization covers a wide range of emulation techniques that are applied to different areas of computing
- The first classification discriminates against the service or entity that is being emulated
- Virtualization is mainly used to emulate execution environments, storage, and networks
- Among these categories, execution virtualization constitutes the oldest, most popular, and most developed area
- Execution virtualization techniques can be categorized into two major categories:
  - Process-level techniques are implemented on top of an existing operating system, which has full control of the hardware
  - System-level techniques are implemented directly on hardware and do not require or require a minimum of support from an existing operating system

Taxonomy of Virtualization Techniques



Execution Virtualization

- Execution virtualization includes all techniques that aim to emulate an execution environment that is separate from the one hosting the virtualization layer
- Execution virtualization can be implemented directly on top of the hardware by the operating system, an application, or libraries dynamically or statically linked to an application image

Machine Reference Model

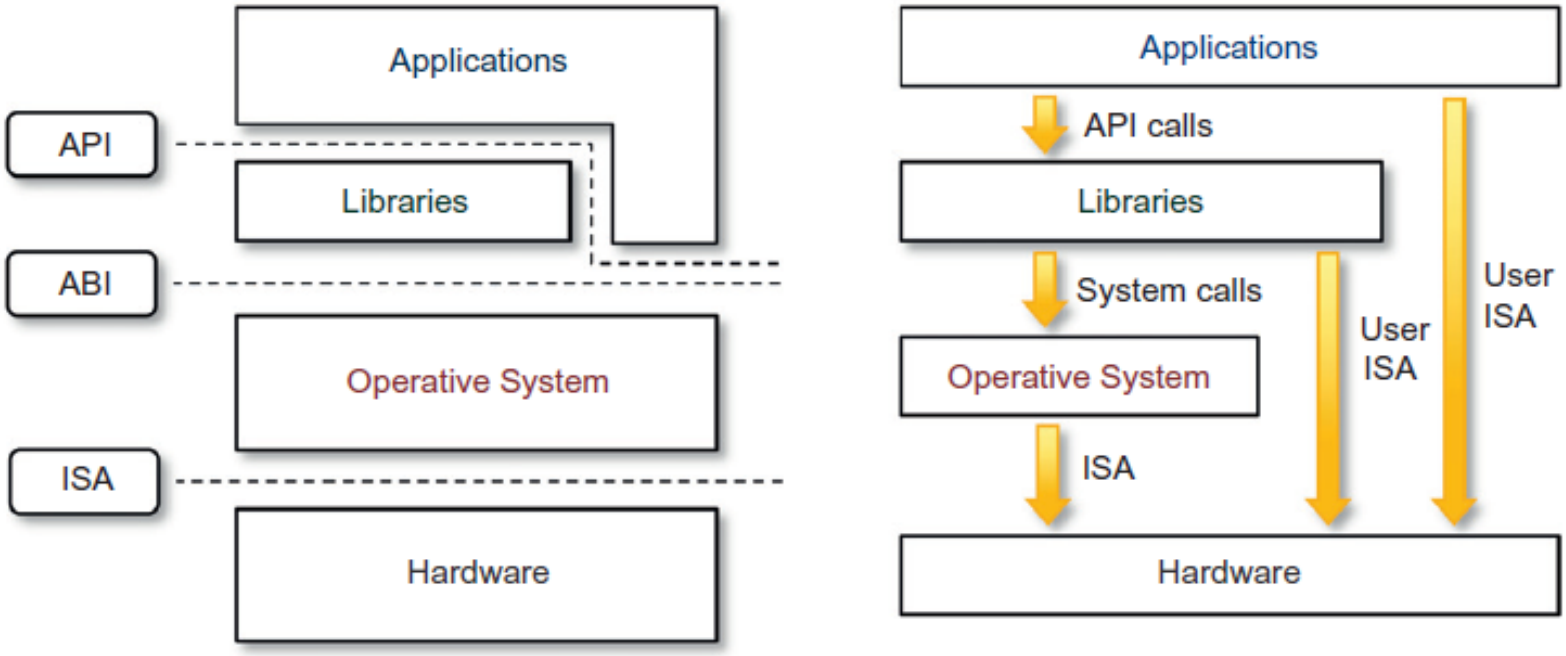


FIGURE 3.4

A machine reference model.

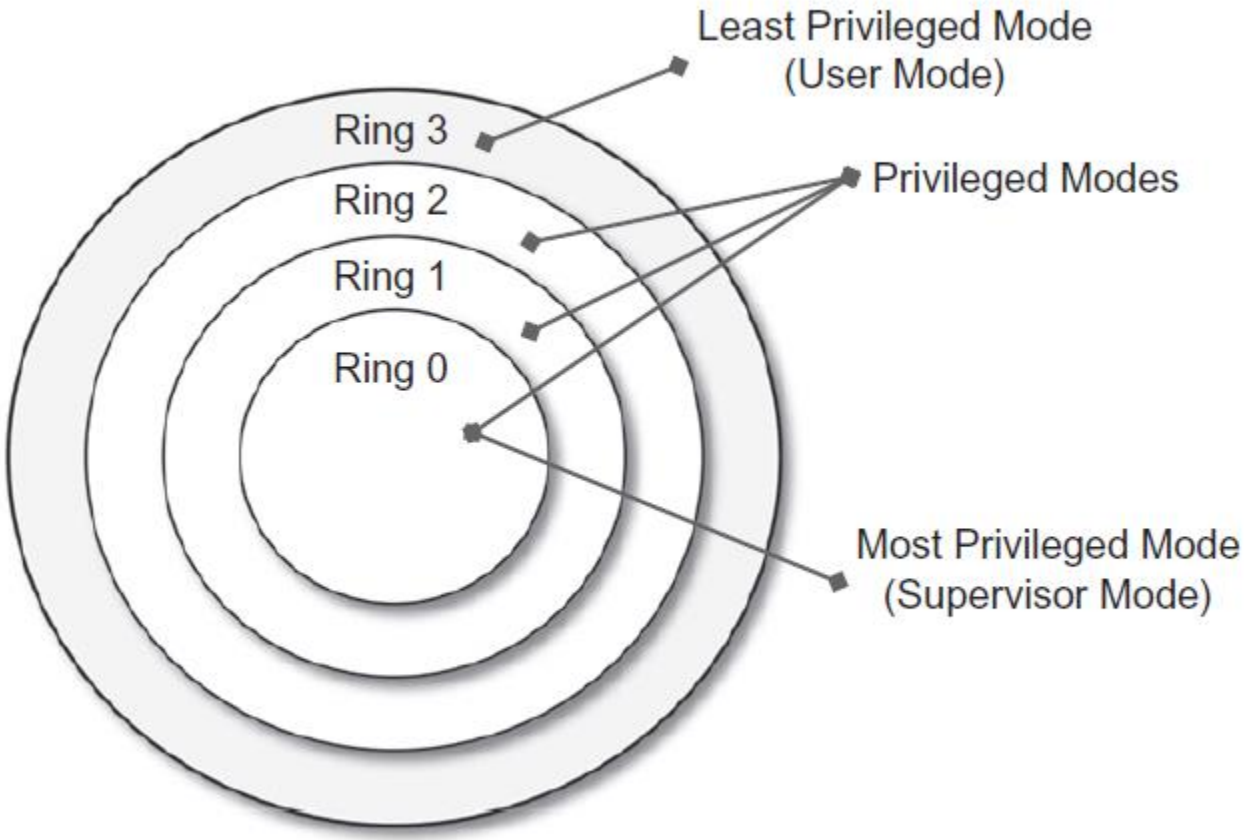
ISA ,ABI,API

- Modern computing systems can be expressed in terms of the 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
- 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. System calls are defined at this level
- This interface allows portability of applications and libraries across operating systems that implement the same ABI
- The highest level of abstraction is represented by the application programming interface(API), which interfaces applications to libraries and / or the underlying operating system

Privileged and Non-Privileged Instructions

- The high-level abstraction is converted into machine-level instructions to perform the actual operations supported by the processor
- The instruction set exposed by the hardware has been divided into different security classes that define who can operate with them
- The first distinction can be made between privileged and non privileged instructions
- Non-privileged instructions are those instructions that can be used without interfering with other tasks because they do not access shared resources
- Privileged instructions are those that are executed under specific restrictions and are mostly used for sensitive operations

Security Rings and Privilege Modes



Hardware-level Virtualization

- Hardware-level virtualization is a virtualization technique that provides an abstract execution environment in terms of computer hardware on top of which a guest operating system can be run
- In this model, the guest is represented by the operating system, the host by the physical computer hardware, the virtual machine by its emulation, and the virtual machine manager by the hypervisor
- The hypervisor is generally a program or a combination of software and hardware that allows the abstraction of the underlying physical hardware
- Hardware-level virtualization is also called system virtualization, since it provides ISA to virtual machines, which is the representation of the hardware interface of a system. This is to differentiate it from process virtual machines, which expose ABI to virtual machines

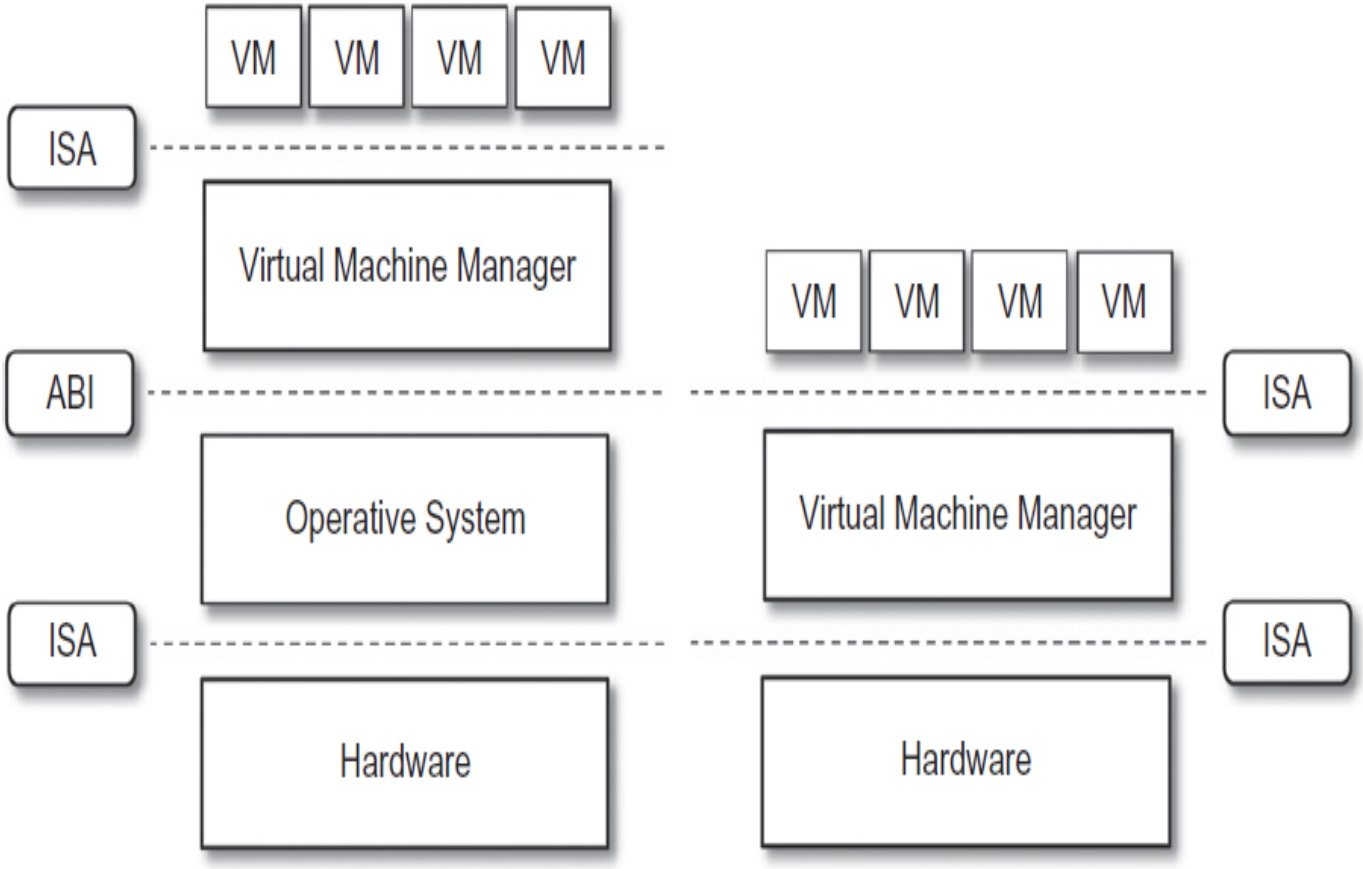
Hypervisors

- 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 hypervisor: Type I and Type II
- Type I hypervisors:
  - 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 in order 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

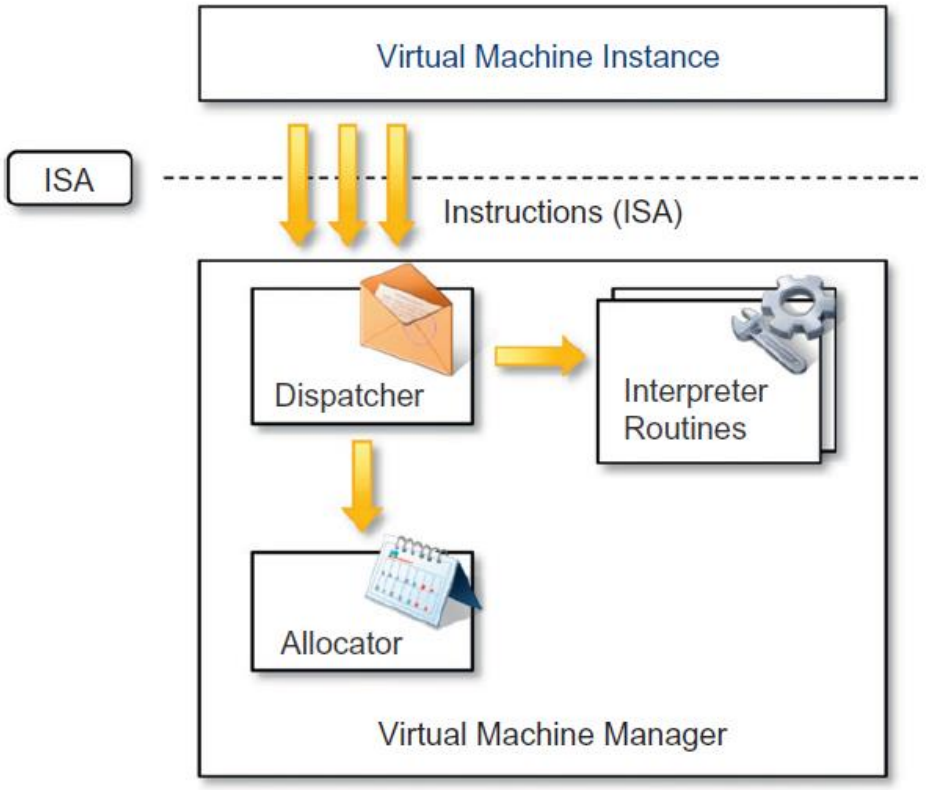
Hypervisors

- Type II hypervisors
  - Require 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

Hosted (left)and Native (right) Virtual Machines types of Hypervisors



A Hypervisor Reference Architecture



A Hypervisor Reference Architecture

- Three main modules, dispatcher, allocator, and interpreter, coordinate their activity in order to emulate the underlying hardware
- The dispatcher constitutes the entry point of the monitor and reroutes the instructions issued 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 allocator is invoked by the dispatcher
- The interpreter module consists of interpreter routines. These are executed whenever a virtual machine executes a privileged instruction: a trap is triggered and the corresponding routine is executed

Properties of VMM

- Three properties have to be satisfied by VMM:
  - Equivalence
  - Resource control
  - Efficiency