SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)

Approvedy by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER APPLICATIONS

COURSE

23CAE717
Cloud Computing

UNIT III

Cloud Infrastructure

TOPIC

Virtualization of CPU, Memory and I/O devices

Semester

II Semester / I MCA

CPU Virtualization

- Modern OS & processors permit multiple processes to run simultaneously
- All processors have at least two modes, user mode and supervisor mode
- Instructions running in supervisor mode are called privileged instructions
- Other instructions are unprivileged instructions
- VMware Workstation is a VM software suite for x86 and x86-64 computers
- KVM (Kernel-based Virtual Machine) is a Linux kernel virtualization infrastructure

CPU Virtualization

- unprivileged instructions of VMs run directly on the host machine
- critical instructions should be handled carefully
- Three categories of critical instructions: privileged instructions, control-sensitive instructions, and behavior-sensitive instructions
- Privileged instructions execute in a privileged mode and will be trapped if executed outside this mode.

CPU Virtualization

- Control-sensitive instructions attempt to change the configuration of resources used.
- Behavior-sensitive instructions have different behaviors depending on the configuration of resources, including the load/store operations over the virtual memory
- RISC CPU architectures can be naturally virtualized
- x86 CPU architectures are not primarily designed to support virtualization, because 10 sensitive instructions, are not privileged instructions

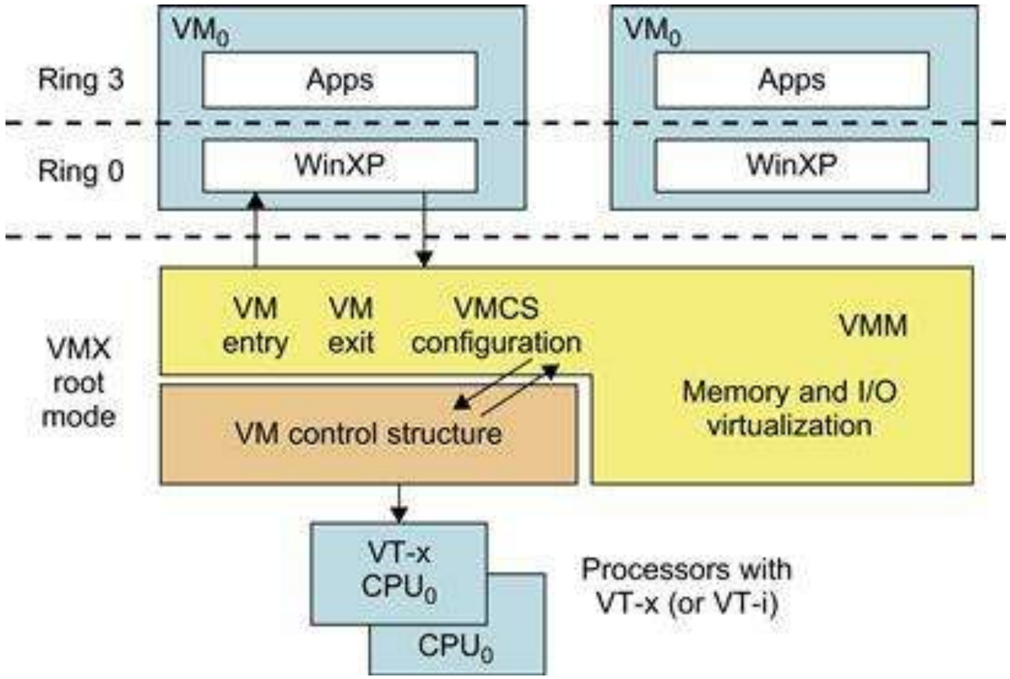
Hardware assisted CPU Virtualization

- Intel and AMD add an additional mode called privilege mode level (some people call it Ring-1) to x86 processors. Therefore, operating systems can still run at Ring 0 and the hypervisor can run at Ring -1.
- All the privileged and sensitive instructions are trapped in the hypervisor automatically.
- This technique removes the difficulty of implementing binary translation of full virtualization

CPU Virtualization

- A CPU architecture is virtualizable if it supports the ability to run the VM's privileged and unprivileged instructions in the CPU's user mode while the VMM runs in supervisor mode.
- When the privileged instructions including control- and behavior-sensitive instructions of a VM are executed, they are trapped in the VMM

Memory Virtualization

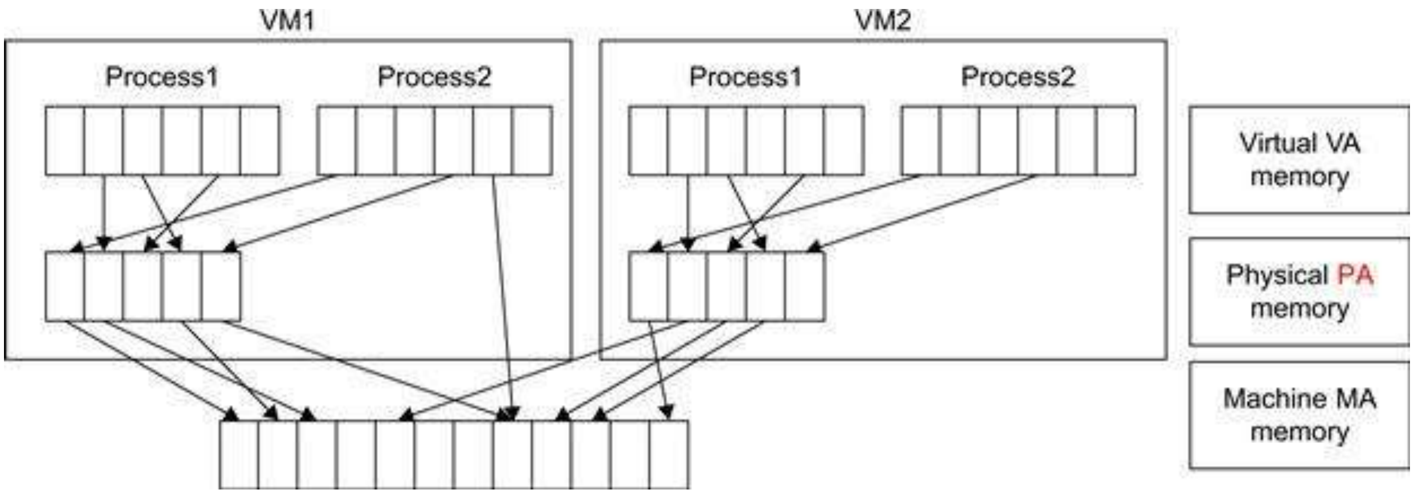


Intel hardware-assisted CPU virtualization

Memory Virtualization

- Virtual memory virtualization is similar to virtual memory supported by modern OS
- modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance.
- Two-stage mapping process should be maintained by the guest OS and the VMM, respectively: virtual memory to physical memory and physical memory to machine memory

Memory Virtualization



Two-level memory mapping procedure

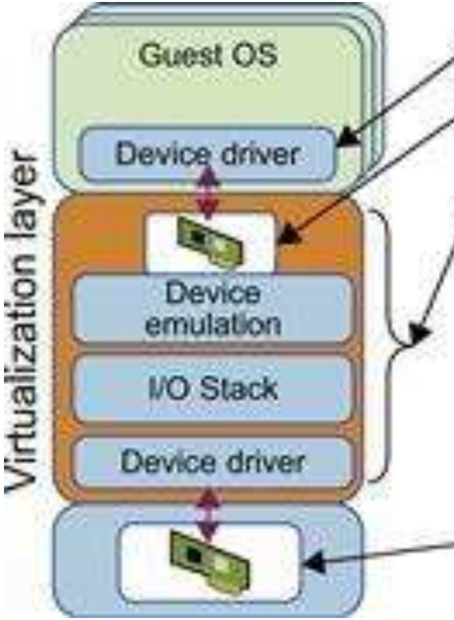
Memory Virtualization

- each page table of the guest OSes has a separate page table in the VMM corresponding to it, the VMM page table is called the shadow page table
- MMU already handles virtual-to-physical translations as defined by the OS
- VMware uses shadow page tables to perform virtual-memory-to-machine-memory address translation.
- Processors use TLB hardware to map the virtual memory directly tothe machine memory to avoid the two levels of translation on every access

I/O Virtualization

- involves managing the routing of I/O requests between virtual devices and the shared physical hardware
- three ways to implement I/O virtualization:
  - full device emulation
  - para-virtualization
  - direct I/O
- ***Full device emulation***: All the functions of a device or bus infrastructure, such as device enumeration, identification, interrupts, and DMA, are replicated in software

I/O Virtualization



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

I/O Virtualization

- ***Para-virtualization***: consisting of a frontend driver and a backend driver.
- The frontend driver is running in Domain U and the backend driver is running in Domain 0.
- The frontend driver manages the I/O requests of the guest OSes
- Backend driver is responsible for managing the real I/O devices and multiplexing the I/O data of different VM
- achieves better device performance than full device emulation

I/O Virtualization

- ***Direct I/O virtualization***: lets the VM access devices directly.
- It can achieve close-to- native performance without high CPU costs.
- self-virtualized I/O (SV-IO): All tasks associated with virtualizing an I/O device are encapsulated in SV-IO. It provides virtual devices and an associated access API to VMs and a management API to the VMM
- defines one virtual interface (VIF) for every kind of virtualized I/O device, such as virtual network interfaces, virtual block devices (disk), virtual camera devices,

References

- Kai Hwang, Geoffrey C Fox, Jack G Dongarra, “Distributed and Cloud Computing, From Parallel Processing to the Internet of Things”, Morgan Kaufmann Publishers, 2012
- James E. Smith, Ravi Nair, “Virtual Machines: Versatile Platforms for Systems and Processes”, Elsevier/Morgan Kaufmann, 2005.
- Kumar Saurabh, “Cloud Computing – insights into New-Era Infrastructure”, Wiley India,2011.
- Toby Velte, Anthony Velte, Robert Elsenpeter, “Cloud Computing, A Practical Approach”, TMH, 2009.
- John W.Rittinghouse and James F.Ransome, “Cloud Computing: Implementation, Management, and Security”, CRC Press, 201