Virtual Machine – virtualization of a computer

Physical Machine

Guest OS

OS

Hardware

Virtual Machine

Hypervisor/VMM

OS 1

OS 2

VMMs running on top of a physical machine

Virtualizes hardware for each VM to run on
How does the hypervisor virtualizes the hardware?

1) How to virtualize CPU states & privilege modes?

→ Guest OS can't directly run in kernel mode, why?

→ Trap & Emulate

⇒ Run guest OS in user mode, when it accesses privilege instr. traps into the hypervisor & hypervisor changes virtualized registers states according to the instr. (Eflags, control registers)

⇒ What happens when application requests a kernel service?

⇒ Will trap into the hypervisor, hypervisor then forwards the trap to the guest OS to handle)

※ Large overhead (lots of control transfers & mode switches)

⇒ Intel VT-x (HW support for CPU virtualization)

⇒ Root & non-root mode (Each with its own privilege rings)

⇒ New instructions: VM enter, vmexit

⇒ VMCB: Configures which instr. should trap into root mode

Guest Apps run in non root ring 3

Guest OS in non root ring 0
2). How to virtualize memory?

- provide Guest OS a virtualized physical memory
- Guest OS manages Guest Physical Address, hypervisor translates this into Host Physical Address to perform actual memory access.

- Shadow Paging
  - Guest OS maintains PT for every application (Guest Virtual Addr) -> Guest padded
  - Hypervisor maintains a shadow PT that maps Guest Addr -> Host Padder, this is installed in CR3
    - hypervisor involved in all changes in mappings, update shadow PT.
    - Guest OS wasted work in updating the PT.

- Extended Page Table / Nested Page Table
  - HW support to walk PT in both Guest OS & hypervisor
    - For each GNA access, walk [GNA -> GPA] [GPA -> HPA]
      - Guest PT to find GPA, for each GPA, walk the hypervisor PT to find the actual physical address (HPA)