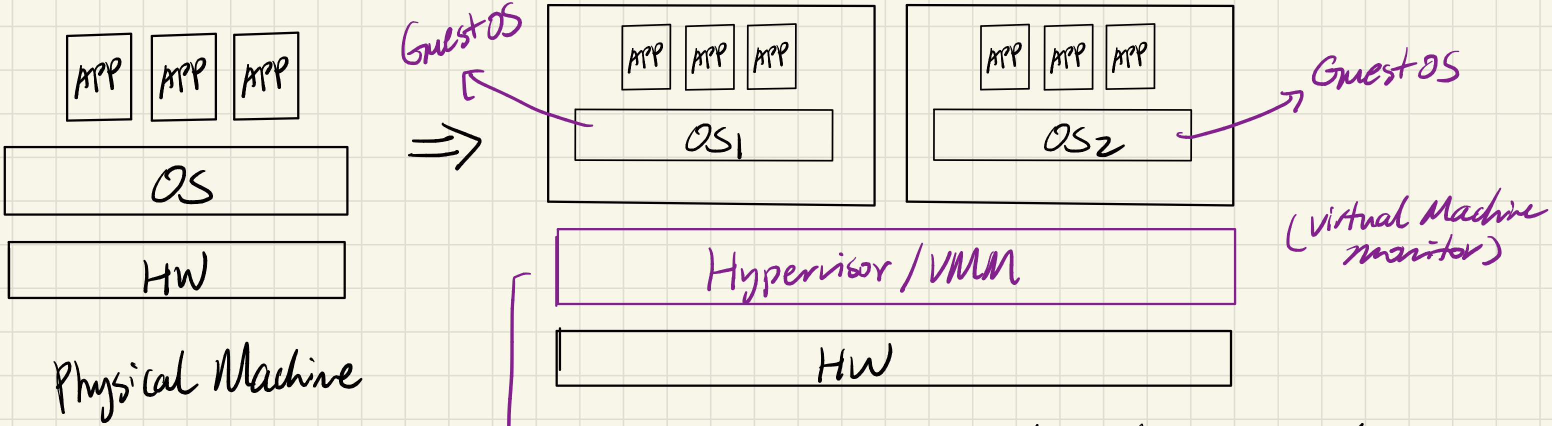
12/2 Virtualization

Virtual Machine - virtualization of a computer

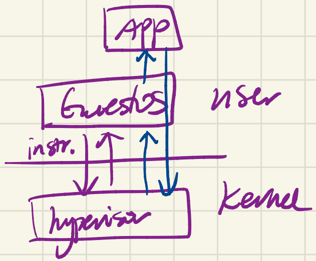


VMs running on top of a physical machine

virtualizes hardware for each VM to run on

How does the hypervisor virtualizes the hardware?

1). How to virtualize CPU states & priviledge modes?

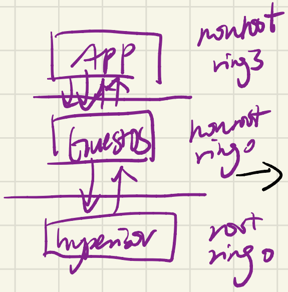


→ Guest OS can't directly run in kernel mode, why?

→ 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 w/ its own priviledge rings)
(hypervisor) (VM)

Guest Apps run in non root ring 3

Guest OS in non root ring 0

↳ new instructions: VM enter, vmexit

↳ VMCB: configures which instr. should trap into root mode

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 peform actual memory access.

-> Shadow Paging

- Guest OS maintains PT for every application (Guest virtual Addr => Guest paddr)
- hypervisor maintains a shadow PT that maps Guest Vaddr => Host Paddr, this is installed in Cr3

↳ hypervisor involved in all changes in mappings, need to 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 GVA access, walk [GVA => GPA] [GPA -> HPA] Guest PT to find GPA, for each GPA, walk the hypervisor PT to find the actual physical address (HPA)

done by HW ←