

⊗ vmcs → per VM per vCPU
 ↳ updated by VMM or CPU (h/w) (root+ring0)

⊗ memory virtualization. for VMs

address space abstraction.

○ starting, linear, isolation, per process
 - h/w. assistance: mmu, privilege modes, cr3, cr0, cr2,
 pte flags, page fault/trap, ...

via paging + segm. support.

- the same/^{OS-}consistent view is required for virtualization?

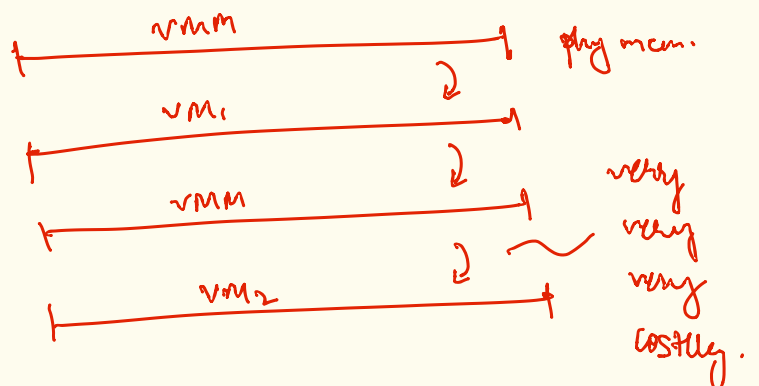
memory virtualization/abstraction mechanism ↳ paging swapping
 memory mgmt. policy ~ LRU, LFU, ARC ~

⊗ how to provide this view/support for VMs?

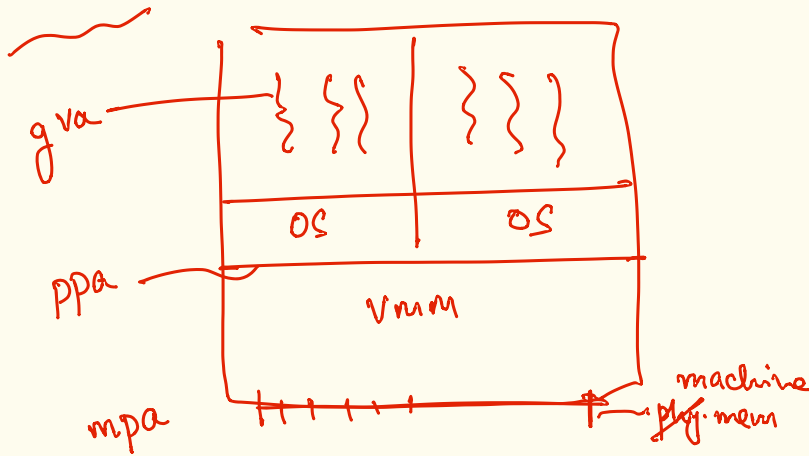
↳ multiplexing VMs! ~ VM memory!
 ↳ all physical memory is not available to a OS VM.

VM : ~ 4 vcpus
 ↳ 32 GB memory

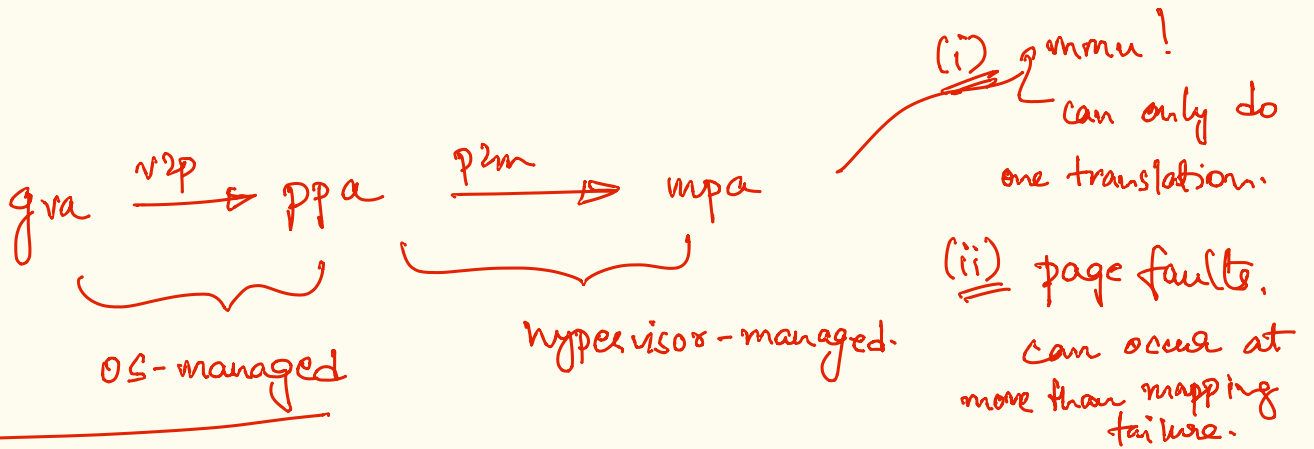
time design 1:
~~space~~ multiplexing of memory.



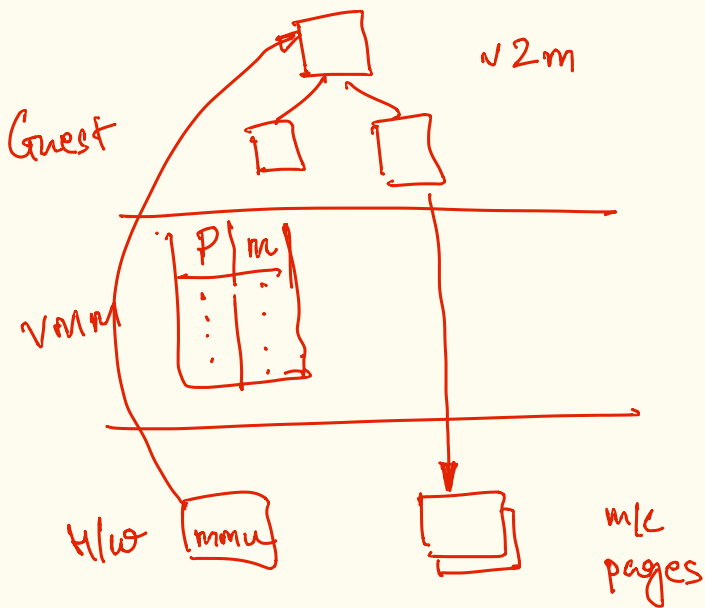
design 2
spatial multiplexing.



gva: guest virtual address
ppa - pseudo physical address
mpa: machine physical address



Design 1: Direct Mapping (Para Virtualization approach)



v2p } v2m
p2m }
→ w/ a single mmu.

⊗ how to build the v2m mapping?

- all guest page tables are write-protected. ⊗
- guest update to page tables via the VMM.

