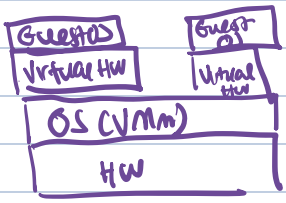
Discussion Plan:

0) Background on Virtualization
- simulation of computer
- uses
  - run multiple OS
  - kernel development
- accuracy:
  - reproduce quirks (malicous)
  - maintain isolation w/ host
  - cannot tell you are running in guest
- History:
  - 1960s, 70s: IBM
  - 90s: VMWare repopularized VMs for x86 hardware
  - 2000s: Amd/Intel built specialized CPUs for virtualization



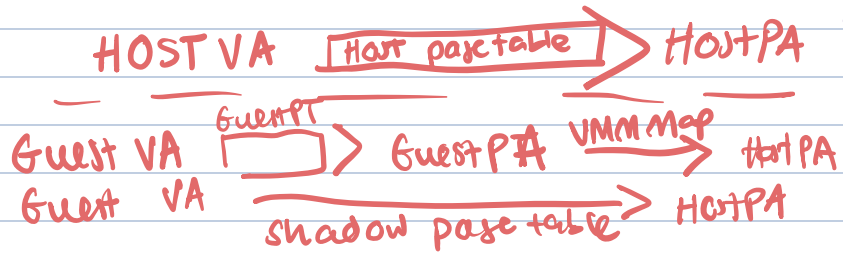
- General goals of virtualization:
  1) Fidelity = identical execution
  2) Performance
  3) Safety

simulation?
- lacks perf

execute guest instructions on real CPU?
perf

Trap-and-emulate: (run at CPL3 (protection level))
↳ only traps and emulates PRIVELEGED instructions
- but some instructions behave differently — no fidelity

- How does paging work?



Issue: VMM can't modify PT in place

↳ 1st time new process PT is accessed - fill in SPT

- Problem: x86 not virtualizable w/ trap-and-emulate
  - main problem: CPL3 vs. CPL0 → you can tell you're in CPL3
    - sometimes priveleged bits masked out

· Two solutions:

BT (Binary Translation) - rewrite offending instructions

HW (Hardware Virtualization) - CPU maintains priveleged state, executes privelged guest instructions

- CPU maintains guest copy of priveleged state in VMCS
- hardware saves and restores priveleged reg. state to/from VMCS as it switches from VMX root to VMX non-root

* MMU: EPT

Guest VA —GuestPT→ Guest PA —EPT→ Host PA

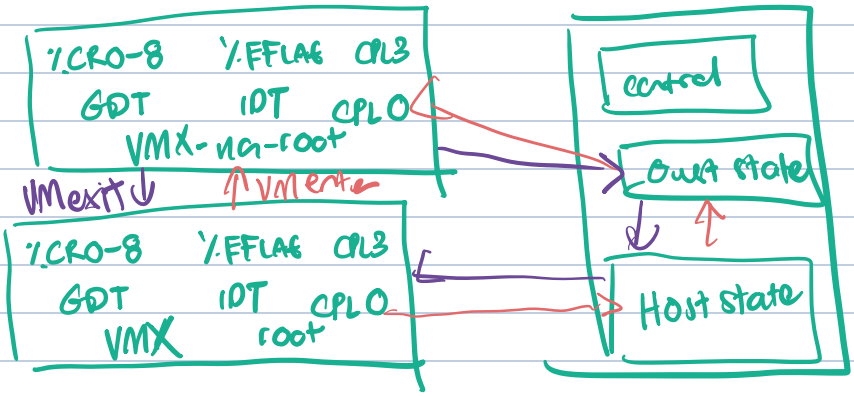
- EPT better when PT contents change frequently
- shadow PT better when PT rare stale

· now: what is VT-x?

↳ hardware extensions that make x86 virtualizable w/ fidelity, perf, and safety

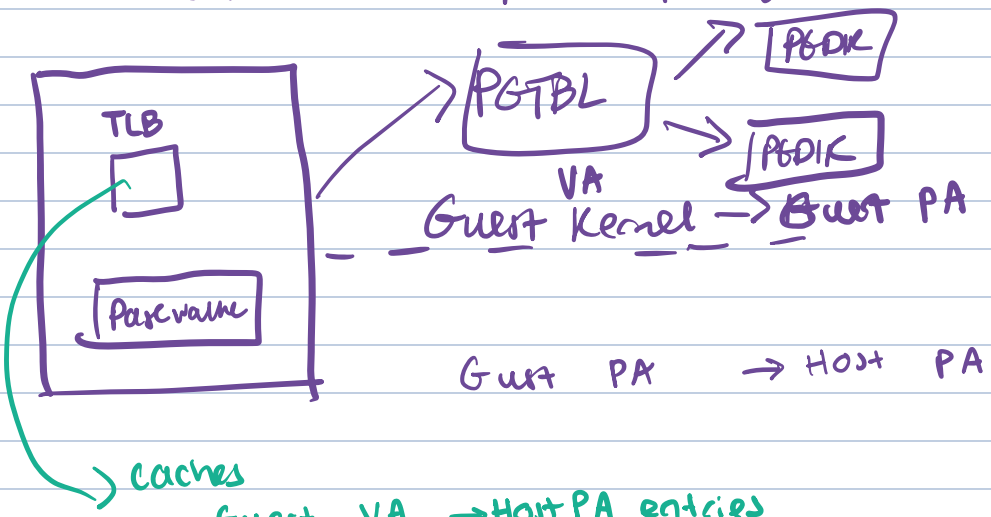
· goals: DIRECT execution of priveleged instruction

- two CPU modes:
  - VMX root mode
  - VMX non-root mode (guest)
  - each has its own protection rings



- enter is either LAUNCH or resume

EPT? enables DIRECT execution of guest PT interactions
- reads and writes to PT in memory
- maintain two layers of paging translation



Dune:
- use EPT, VT-x to support Linux Processes
  - dune mode = isolated w/ VT-x non-root mode
    ↳ not w/ cpl3+ PT protections (trap + emulate)
  - configure EPT so dune process can only access PAs it can normally have access to
    * process can manipulate its own PT! (using %CR3)
    - fast exceptions due to shadow IDT
    - can run sandboxed code (where most apps come)

* GC app that is sped up by Dune:
  - GC: find live data, reclaim non live data

- concurrent GC (Boehm):
  - mutator runs in // w/ trace
  - at some point, mutator might modify pointers in already traced objects
  - we need to figure out which ones were modified
- Dune:
  - clear dirty bits at beginning of GC
  - → scan for set PTE dirty bits to find written page

Sources: https://abelay.github.io/6828seminar/notes/65810_lec_dune.pdf