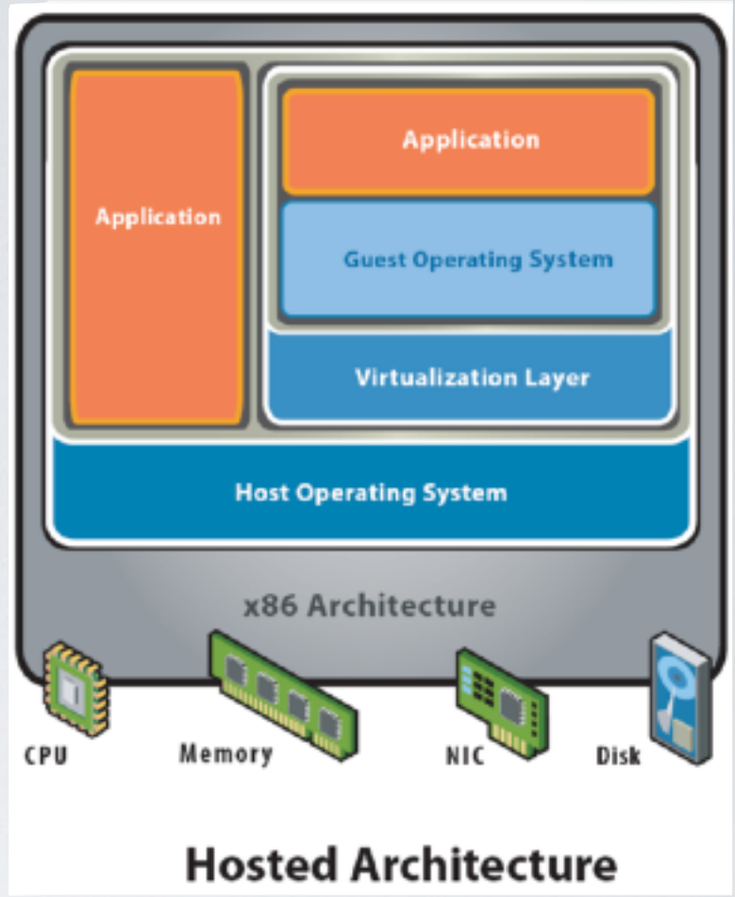
VMM case study 2 - *VMware*



VMware uses software virtualization

- Dynamic binary rewriting translates code executed in VM
  - Most instructions translated identically, e.g. `movl`
  - Rewrite privileged instructions with emulation code (may trap), e.g. `popf`
- Think JIT compilation for JVM, but full binary x86 to IR code to safe subset of x86
- Incurs overhead, but can be well-tuned (small % hit)

✓ VMware workstation uses hosted model

- VMM runs unprivileged, installed on base OS (+ driver)
- Relies upon base OS for device functionality

✓ VMware ESX server uses hypervisor model similar to Xen, but no guest domain/OS

Summary

VMMs multiplex virtual machines on hardware

- Export the hardware interface
- Run OSes in VMs, apps in OSes unmodified
- Run different versions, kinds of OSes simultaneously

Implementing VMMs means virtualizing CPU, Memory, I/O

➡ Lesson: never underestimate the power of indirection