Virtual Machines

Virtual Machine Architecture

Non-VM

User programs

OS

Hardware

Native VM

User programs

User programs

User programs

User programs on native OS

User programs

User programs

User programs

OS

OS

OS

VM monitor

Hardware

Hosted VM

User programs

User programs

User programs

User programs on native OS

User programs

User programs

User programs

OS

VMM

Native OS

Hardware

Why Virtual Machine?

Allow flexible management of "machines" at software level
- experimenting with new architecture
- debugging an OS
- checkpointing and migrating all state on a machine

Enhanced reliability and security
- VM monitor much smaller than OS, therefore:
  - the full privileged code base (VM monitor) is small
  - the trusted code base (VM monitor) is small

Strong isolation between VMs
- fault and resource isolation
- your Xen/Linux assignment

Virtual Machines

- Virtual machine architecture
  - Virtualization: A piece of software that provides an interface identical to the underlying bare hardware.
    - the upper-layer software has the illusion of running directly on hardware
    - the virtualization software is called virtual machine monitor
  - Multiplexing: It may provide several virtualized machines on top of a single piece of hardware.
    - resources of physical computer are shared among the virtual machines
    - each VM has the illusion of owning a complete machine

- Trust and privilege
  - the VM monitor does not trust VMs
  - only the VM monitor runs in full privilege

- Compared to an operating system
  - VM monitor is a resource manager, but not an extended machine

Virtual Machine Architecture

Non-VM

User programs

OS

Hardware

Native VM

User programs

User programs

User programs

User programs on native OS

User programs

User programs

User programs

OS

VMM

Native OS

Hardware

Hosted VM

User programs

User programs

User programs

User programs on native OS

User programs

User programs

User programs

OS

VMM

Native OS

Hardware
Virtualization Challenges

- CPU virtualization
  - how to switch out a VM?

- Memory virtualization
  - VM physical memory address may not be real machine address
  - a VM’s memory access must be restricted

- I/O virtualization
  - similar issues with memory virtualization

Virtualization Approach - Interpretation

- Do not directly run VM code ⇒ Interpretation
  - inspect each instruction in software and realize its intended effects using software
  - Nachos VM does this

- CPU virtualization
- Memory virtualization
- I/O virtualization

- Problem: too slow!

Virtualization Approach – Direct Execution

- Directly executing VM code to attain high speed

- CPU virtualization
  - VM monitor catches timer interrupts and switches VM if necessary

- I/O access virtualization
  - cause a trap to VM monitor, which processes appropriately
  - extra overhead is not too bad

- Memory virtualization
  - a trap at each memory access is not a very good idea
  - How?

Memory Virtualization Under Direct Execution (protected page table)

- From the VM OS’s view, the page table contains mapping from virtual to VM physical addresses
- For proper operation, the page table hooked up with MMU must map virtual to real machine addresses

- VM OS cannot directly access the page table
  - each page table read is trapped by VM monitor, the physical address field is translated (from real machine address to VM physical address)
  - each page table write is also trapped, for a reverse translation and for security checking
Memory Virtualization Under Direct Execution (shadow page table)

- VM OS maintains virtual to VM physical (V2P) page table
- VM monitor
  - maintains a VM physical to machine (P2M) mapping table
  - combines V2P and P2M table into a virtual to machine mapping table (V2M)
  - supplies the V2M table to the MMU hardware
- Page table updates
  - any VM change on its V2P page table must be trapped by VM monitor
  - VM monitor modifies V2M table appropriately

Virtual Machine Transparency

- Full transparency (perfect virtualization):
  - stock OS (without change) can run within VM
  - VMware

- Less-than-full transparency (para-virtualization):
  - modified OS runs within VM
  - Xen
  - for performance (memory virtualization)
    - batched page table accesses through explicit monitor calls
  - for simplicity (I/O virtualization)

VMware Memory Management [Waldspurger OSDI 2002]

- Transparent VM memory need estimation
  - Working set estimation through sampling
- Transparent VM memory size adjustment
  - Ballooning
- Discover and share pages of the same content over multiple VMs.
  - discover: compare hash coding of pages.
  - share: copy-on-write.
  - How often do pages have the same content?

Live Migration

- Migrating a VM from one physical machine to another
  - minimal freeze time

- Migration approaches
  - stop the VM; move the VM state to the new machine; start it
  - stop the VM on the old machine; set up the skeleton on the new machine (all, or most, page table entries invalid) and then start it
  - keep the VM running on the old machine; move state over on the background; then repeatedly move dirty state until it is small; stop the VM on the old machine; move the final dirty state; start it on the new machine [Clark et al. NSDI 2005]
VM-based Intrusion Diagnosis

- In a normal system, a superuser has full trust of the machine
  - when an intruder assumes the superuser identity, he/she can erase all traces of the intrusion
- In a VM platform, the superuser of a VM does not have full trust of the machine
  - even if an intruder assumes the superuser identity of a VM, he/she cannot erase information recorded by VM monitor

- VM-based backtracking intrusions [King and Chen, SOSP 2003]