Virtualization

Separating the abstract view of computing resources from the implementation of these resources

A layer of indirection between abstract view and implementation
- Hides implementation details
- Controls mapping from abstract view to implementation

"any problem in computer science can be solved with another layer of indirection"
- David Wheeler
Virtual Machines

Layer of Indirection (VMM)

Virtual Machine

Operating System

Physical Machine

CPU

Mem

Net

App 1

App 2

App 3

App 4

App 5

Virtual Machine Monitor (VMM)
Machine Virtualization

- A virtual machine abstracts the computing resources of a physical machine into virtual resources.
- End users only see the virtual resources:
  - Can install their operating systems and run their applications on the virtual machines.
- A Virtual Machine Monitor (or Hypervisor) is a software layer that implements the mapping from virtual resources to physical resources.

Virtual Machine Monitors

- Strong isolation between virtual machines.
- Flexible mapping between virtual resources and physical resources:
  - Can have more virtual resources than the corresponding physical resources.
  - Can reallocate physical resources among VMs.
- Pause, resume, checkpoint, and migrate virtual machines.
Virtual Storage

Why Use Virtual Machines?

- Server consolidation
  - Traditional IT setup: one machine per application (DBMS, web server, mail server, …)
  - Provisioned for peak load. Usually under-utilized
  - Instead, can run multiple applications on virtual machines that share the same physical machine
  - Save hardware costs and administration/operation costs
Server Consolidation

- Consolidate onto a single machine
  - Easier to manage
  - Less total capacity than the original two
  - Better utilization than the original two

\[ P_{12} < P_1 + P_2 \]

Consolidation

- **Economies of scale**
  - Cheaper provisioning, administration, power, networking, and cooling

- **Users benefit too**
  - Efficient access to a larger pool of resources with better manageability and fault tolerance

**Worldwide spending on servers in 2007: US$200 billion**
(30% new servers, 10% power and cooling, 60% administration)

Source: IDC, 2008
Cloud Computing

- Consolidation on massive, shared, hosted computer clusters

Why Use Virtual Machines?

- Improved manageability
  - Dynamic provisioning of resources to VMs
  - Migration of VMs for load balancing
  - Migration of VMs to avoid down time during upgrades

- Isolation between VMs
  - Security
  - Privacy
  - Fault tolerance
Why Use Virtual Machines?

- Application compatibility
  - Different environments for different applications

Why Use Virtual Machines?

- Software development and testing
  - Multiple environments for development and testing

- Software deployment
  - Preconfigured virtual appliances
  - Repositories of virtual appliances on the web
Virtual Appliances

http://www.vmware.com/vmtn/appliances

Virtual Appliances

http://virtualappliances.net/downloads/
Why Not Use Virtual Machines?

- **Performance penalty**
  - Indirection through VMM adds overhead

- **Hiding details of physical resources**
  - Some applications make decisions based on assumptions about the physical resources

Basic Approach to Virtualization

- User Process (user mode)
  - syscall/exception
  - non-privileged access

- Kernel (privileged mode)
  - privileged access

- Exception Handler

- Physical Machine
  - CPU
  - Mem
  - Net

- Trap
Trap-and-Emulate Virtualization

- Run VMM in privileged mode
- Run OS in user mode
- Privileged operations by the OS will trap
- Trap handler in VMM emulates these operations as if they were run on the virtual machine
- Non-privileged operations can proceed as before with no intervention from the VMM
Architectural Obstacles

- Some machine architectures are not easy to virtualize
  - Notable example: x86

- Not all privileged operations trap when run in user mode
  - Example: `popf` (pop stack into flags)
    - Privileged mode: change user and system flags
    - User mode: change user flags only, no trap

- Some privileged state is visible in user mode
  - Example: Machine status word

- For an architecture like x86, **trap-and-emulate alone will not work**

Virtualization Approaches

- **Binary rewriting**
  - Operating system running in VM is **unmodified**
  - VMM scans **Guest OS** memory for problematic instructions and rewrites them
  - Example: VMware Workstation

- **Paravirtualization**
  - Software interface to VMM is **not identical to hardware**
  - Operating systems need to be **ported** to run on VMM
  - Simpler VMM and **faster virtual machines** than with trap-and-emulate
  - Example: Xen
Hardware Virtualization for x86

- Intel and AMD have both introduced processor extensions to help virtualization (Intel VT, AMD-V)
- Processor is aware of multiple virtual machine contexts (like process control blocks, but for entire operating system)
- New instructions to start/resume a VM
- New privilege level for VMM
- VMM selects which events should trap (vmexit)
  - Manipulating interrupt state, interacting with TLB, accessing control registers, …