The Impact of Virtualization on Computer Architecture and Operating Systems

Mendel Rosenblum

LELAND STANFORD JUNIOR UNIVERSITY

ORGANIZED 1891

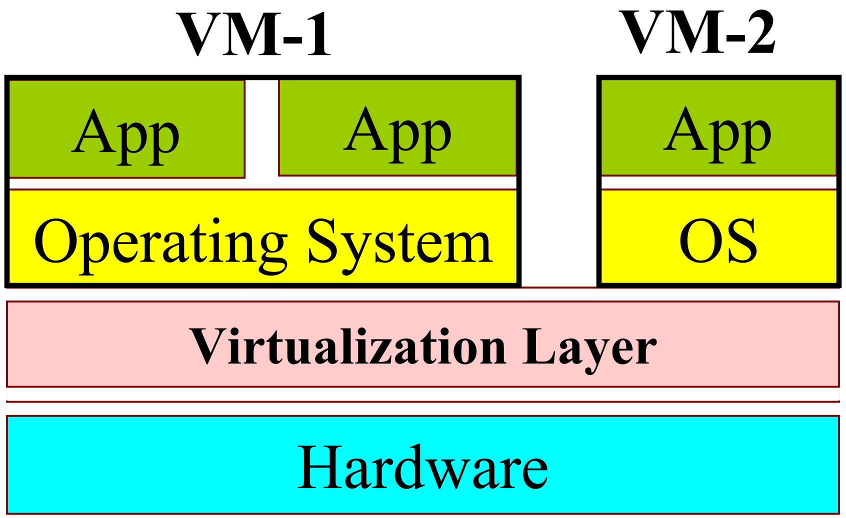
vmware

Talk Outline

- Virtualization
  - What is virtualization? Why is it so compelling?
- Implications for computer architecture
  - Known techniques and current challenges
- Implications for system software
  - Implications for operating systems & OS researchers
- Conclusions

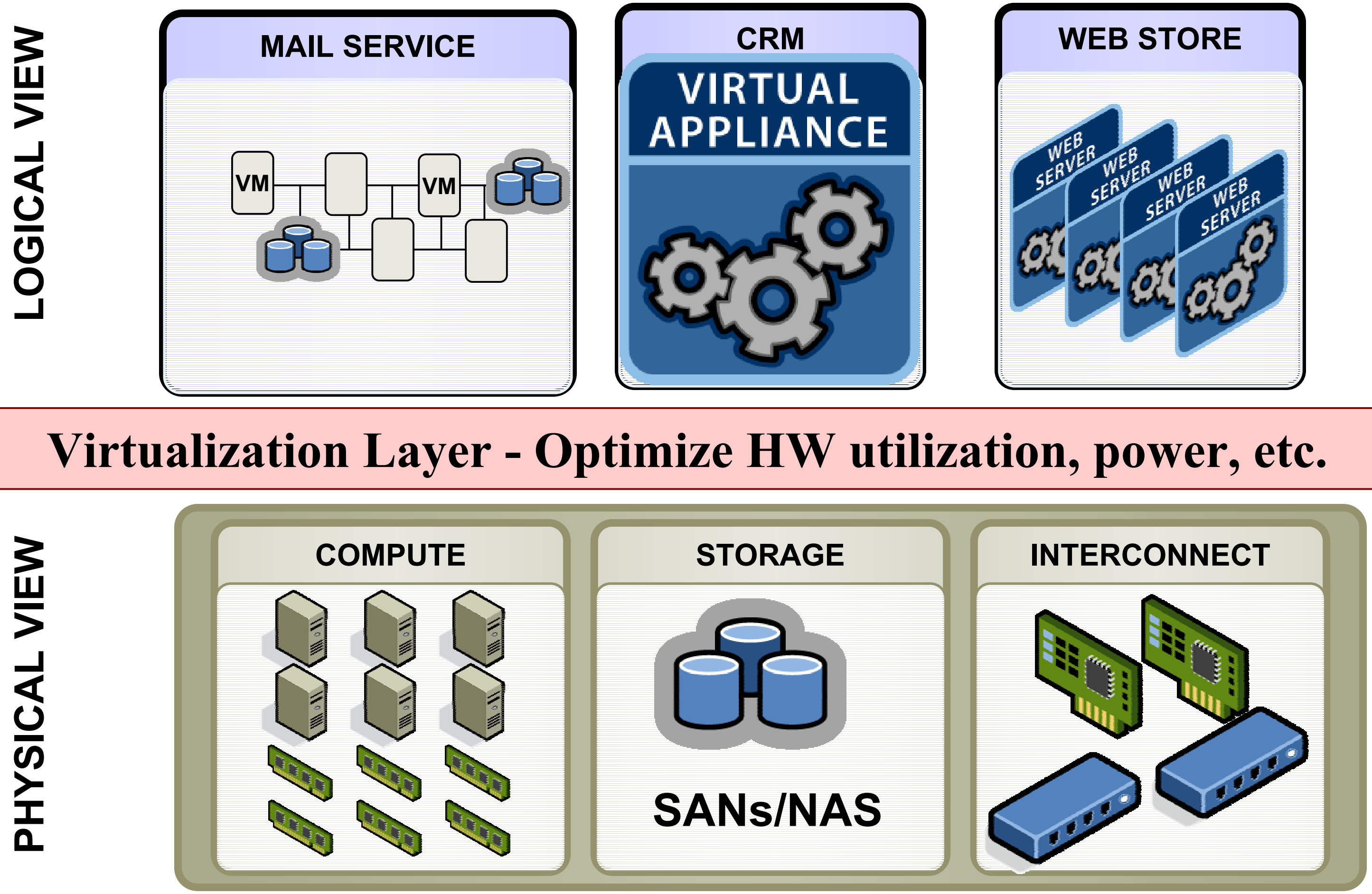
What is Virtualization?

- A level of indirection between hardware and software.



- Virtual Machine abstraction
  - Run all software written for physical machine.

User’s view of virtualization



LOGICAL VIEW

MAIL SERVICE

CRM

WEB STORE

Virtualization Layer - Optimize HW utilization, power, etc.

PHYSICAL VIEW

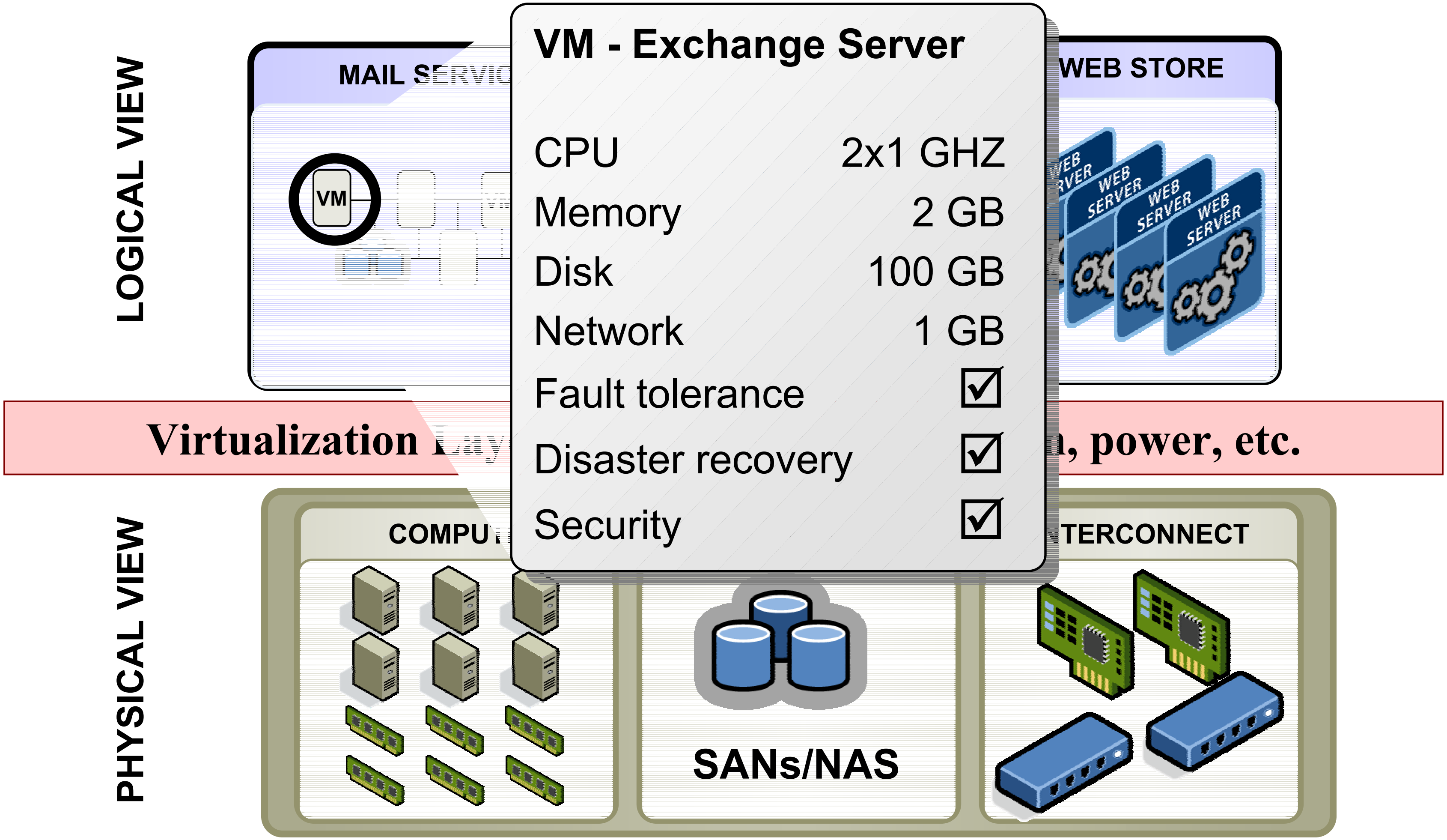
COMPUTE

STORAGE

INTERCONNECT

SANs/NAS

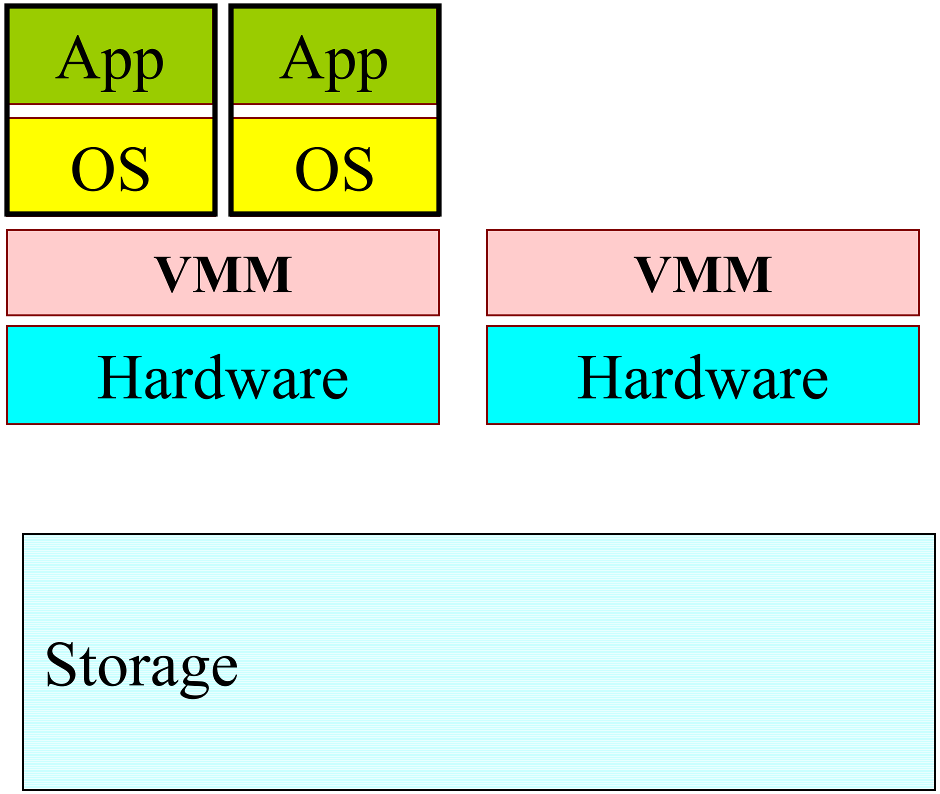
User’s view of virtualization



VM - Exchange Server

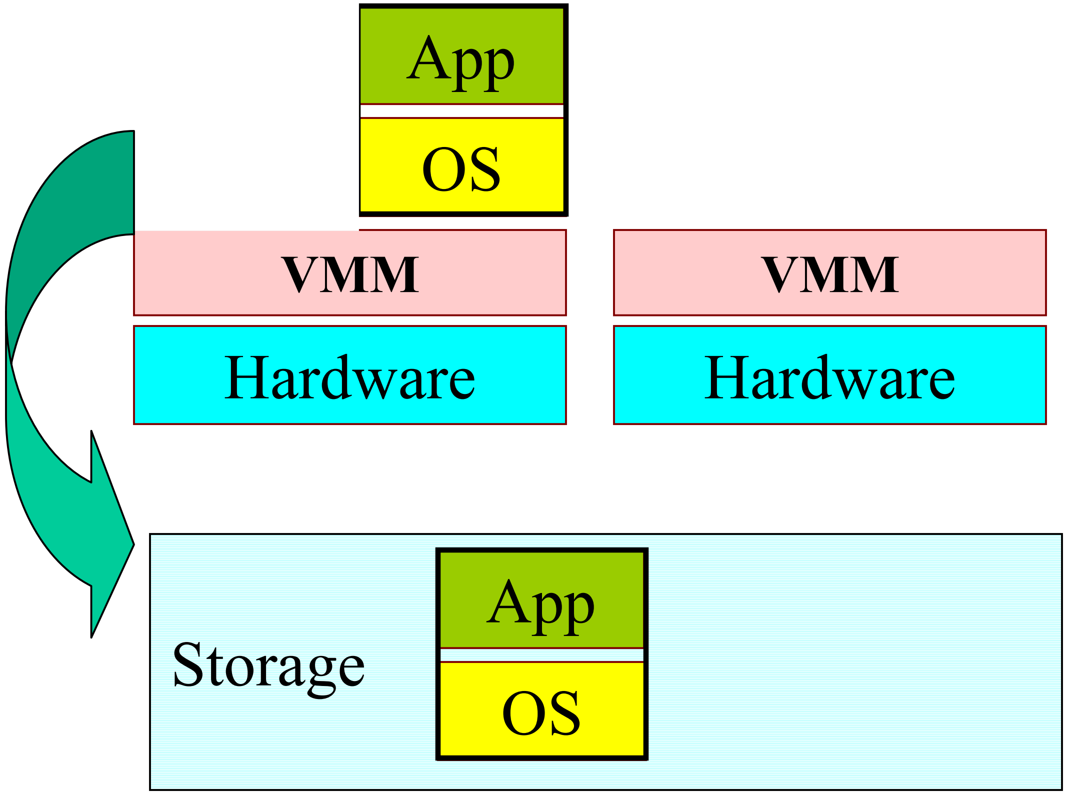
CPU	2x1 GHZ
Memory	2 GB
Disk	100 GB
Network	1 GB
Fault tolerance	☑
Disaster recovery	☑
Security	☑

Key low-level VMM operations



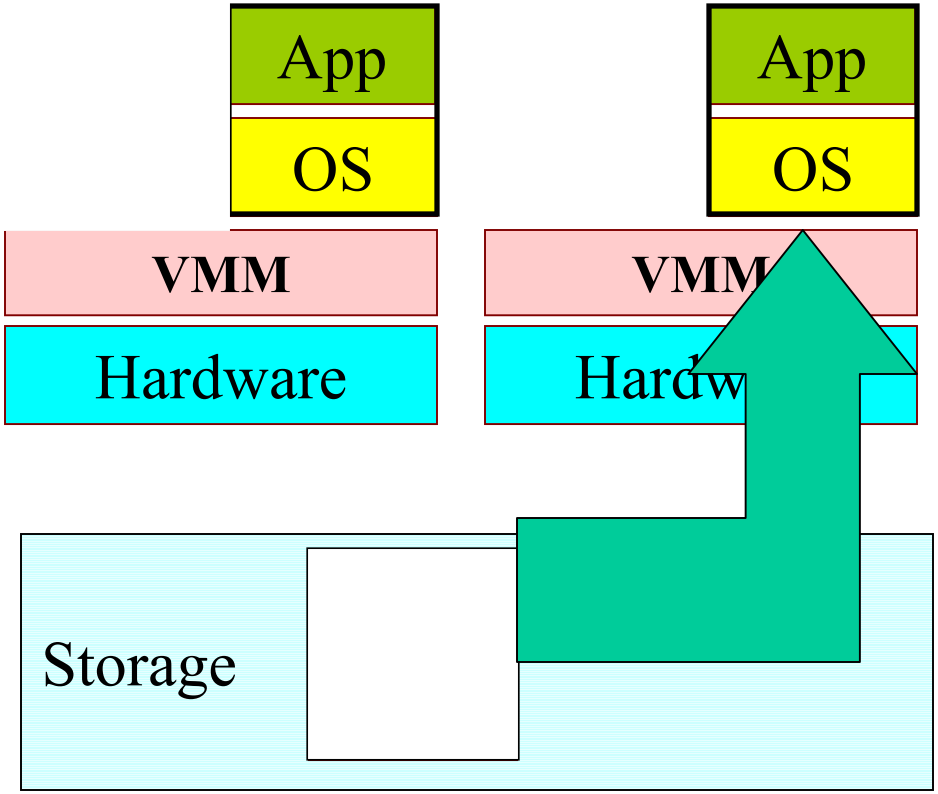
- Multiplex

Key low-level VMM operations



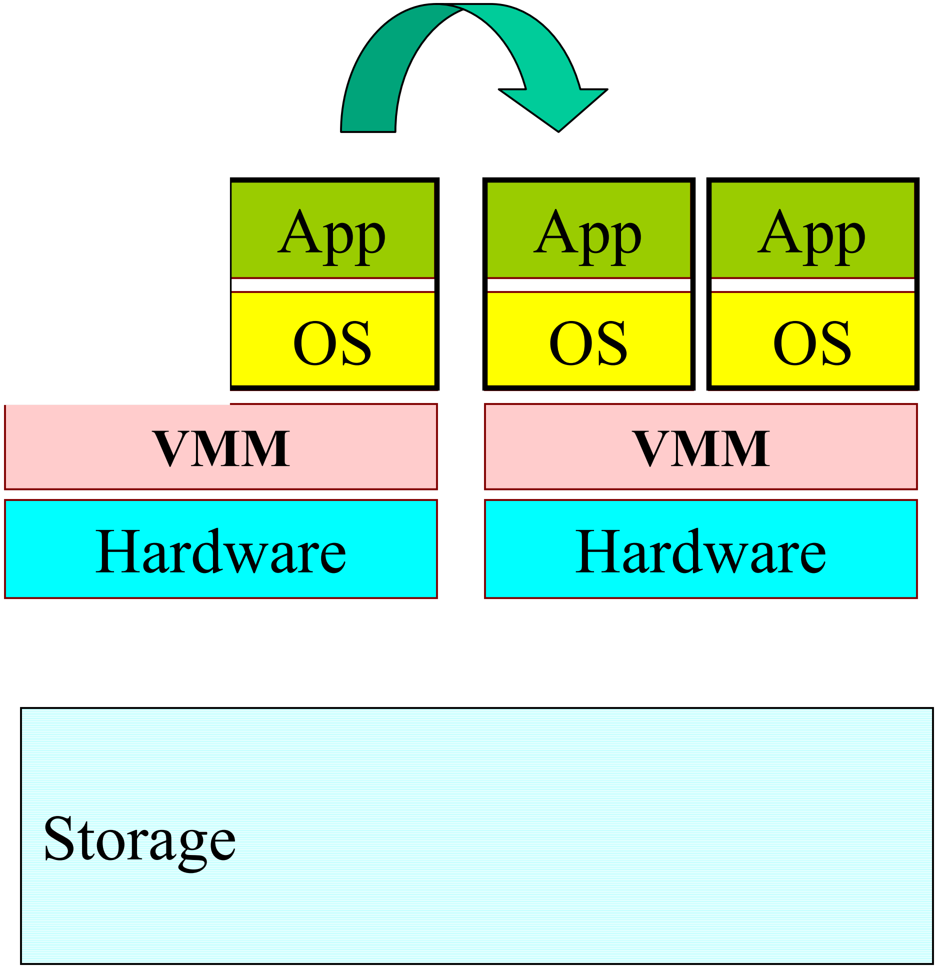
- Multiplex
- Suspend

Key low-level VMM operations



- Multiplex
- Suspend
- Resume (Provision)

Key low-level VMM operations



- Multiplex
- Suspend
- Resume (Provision)
- Migration

VMM Implementation

- **Safely** and **efficiently** multiplex the virtual hardware on the physical hardware
  - Virtual CPUs on Physical CPUs
  - VM’s Physical Memory on Machine’s Memory
  - VM’s I/O Devices on Real I/O Devices
- Norm is **time sharing** rather than **space sharing**.

Hardware support for virtualization

- Goals:
  - Reduce virtualization overheads
    - Goal: Run software same speed as without VMM.
  - Reduce the complexity of VMM software
    - Goal: Trusted code base small ~ 10K lines
- Old hat in mainframe world.
- Current status in the x86 world:
  - CPU -> First generation shipping now.
  - Mem -> First generation shipping soon.
  - I/O -> Still a work in progress. Big challenges.

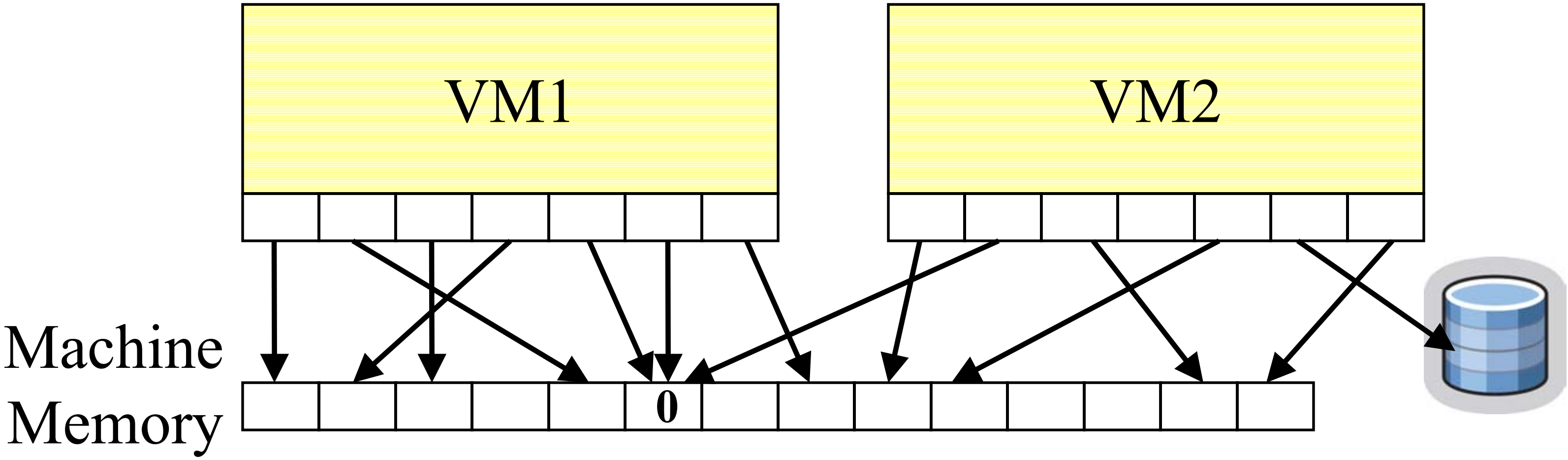
CPU Virtualization Today

- Classic VMM trick: Directly execute VM in less privileged mode on real CPU.
  - Trap and emulate privileged instructions.
- Popular x86 VMMs use binary translation to detect and emulate privileged inst.
  - Works well because of high trap overheads.

Virtual CPU architecture support

- From Mainframes: Microcode assist
  - Fewer traps
- x86 support: Intel’s VT, AMD-V
  - New mode for running VMs
    - Trap and emulate style.
  - Fewer and faster traps
- Right direction but challenges remain
  - See next talk for details.

Virtual Physical Memory



- Virtual Memory like features:
  - Non-contiguous layout
  - COW sharing of identical pages
  - Demand paging allowing memory over-commit.
- Classic VMM: **Shadow Page Tables**
  - VMM uses page table with VA->MA

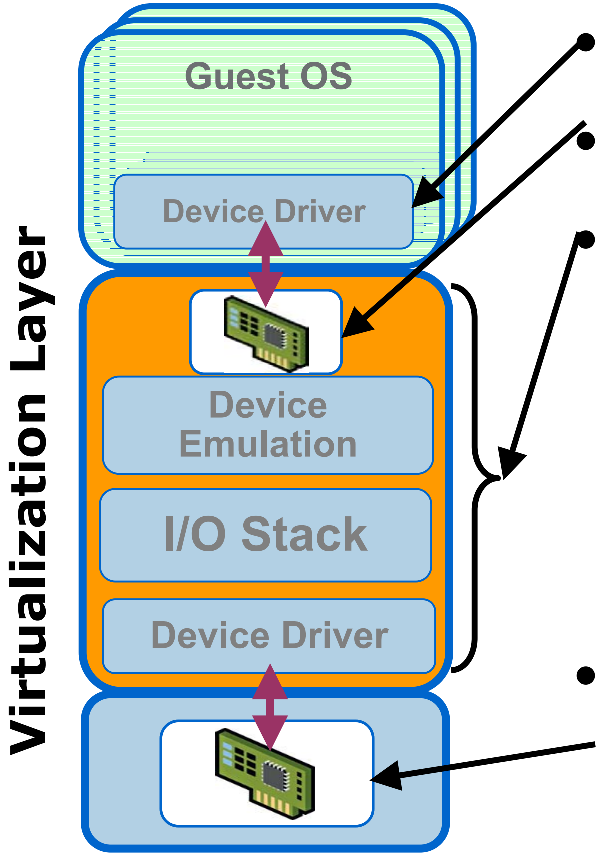
Memory Architecture Support

- Cost of shadow page tables can be high
  - Workloads with many start/exit process
  - OSes that “flip pages” to avoid copies & COW.
- Classic mainframe:
  - Hardware support for the PA->MA map
- x86 Support: AMD’s NPT, Intel’s EPT
  - Another “page table” for mapping

Modern I/O different from 1970’s I/O

- Can’t just read old papers to get solutions
- Large device diversity
  - Not everything is a channel architecture
- High performance I/O devices
  - 10Gig ethernet
  - 3D graphics

Current virtual I/O devices



- Guest device driver
- Virtual device
- Virtualization layer
  - emulates the virtual device
  - remaps guest and real I/O addresses
  - multiplexes and drives the physical device
  - I/O features, *e.g.*, COW disks,
- Real device
  - may be different from virtual device

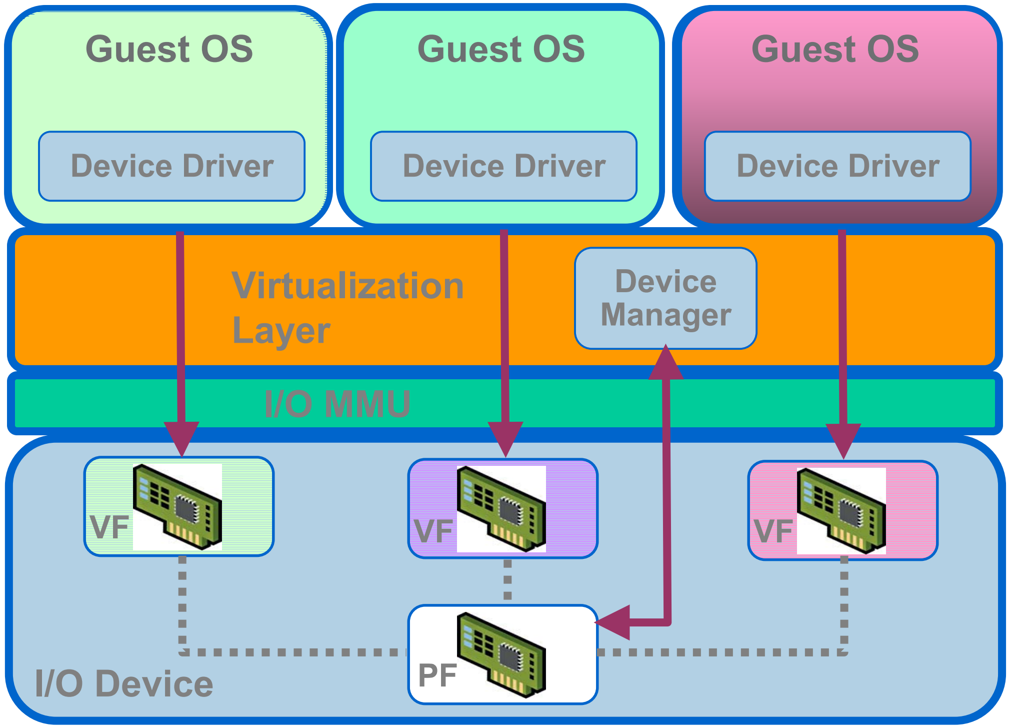
Much functionality in I/O stack

  - De-multiplexing I/Os
  - Converting formats (e.g. SCSI disk -> SAN)
  - Resource management (e.g. traffic shaping)
  - Fault tolerance
  - Enforce security policy

  ... and much more.

- Difficult for hardware to accelerate and maintain rich functionality

Passthrough I/O - Fast but inflexible



I/O devices with:

- **Multiple personalities**

  Interface per VM

- **I/O MMU for DMA**

  Remap PA to MA

  Validate addresses

- **Manageable by VMM**

Passthrough I/O Challenges

- DMA - Can the I/O MMU handle:
  - Discontinuous physical memory?
  - Read-only physical memory (COW)? (Disable?)
  - Paged out physical memory? (Disable?)
- VM Mobility
  - Can I migrate a VM?
  - Can I migrate to a system with a different I/O device?
- Interrupt routing
  - How does the interrupts get to the right VM?

Virtualization of 3D graphics

- UI devices such as GPUs are challenging.
  - Time sensitive and high performance
- How to multiplex screen?
  - Windows vs. full screen
- Virtualization leading to interest in remote display technology
  - Host many PCs on a server

Arch Support for Virtual I/O

- Challenge: Get **acceleration** and **flexibility**
  - Most hardware all or nothing.
  - Designers need to understand VMM functionality.
- Mobility support:
  - Standardized virtual interfaces for devices
  - Ability to load and store virtual device state

Summary of Hardware Support

- Current CPU trends are positive
  - Multicore, etc.
- Virtualization support should give:
  - Lower virtualization overheads
  - Simplify VMM implementation
  - Ubiquitous deployment
- Support should accelerate not replace
  - Give software the ability to use VMM layer.

Operating Systems & Virtualization

Traditional major roles for an OS:

1. Manage hardware resources of machine.
2. Export abstractions and functionality to support application programs.

Virtualization influences:

VM-optimized operating systems (today)

Operating systems for virtual appliances (future)

Paravirtualization

- Old idea, new term:
  - Modify OS to run better inside a virtual machine.
- Examples likely obsoleted by hardware support:
  - Remove trapping instructions from OS
  - Reduce shadow page table overheads
  - Reduce I/O device emulation overheads
- Resource management seems key:
  - GuestOS <-> VMM about resource management
  - CPU and memory resources
  - Get good inter-VM resource managment

More Interesting: Virtual Appliances

- Trend to use virtual machine as software distribution mechanism.
  - Applications and OS bundled together.
  - Like Appliance Computing without hardware
- Many benefits for software vendor and customer:
  - Choose OS based on application needs, not what customer has.
    - Functionality, performance, reliability, security, manageability.
  - Simplify testing and support.
  - Offload much from customer.
- Example: CRM system

Application-selected operating systems

For Virtual Appliances:

- Don’t need hardware management in OS.
  - Current OSes manage hardware.
- Only services for one application needed
  - Current OSes try to support broad range of applications.
- Look at hardware appliance operating systems for examples

Desirable properties for VA OSes

- Highly customizable
  - Include only what application needs.
- Supports common VA functionality
  - “Firmware” update
  - Browser-based interfaces
- Interfaces to VMM and IT infrastructure
  - Authentication, policies, etc.

Implications for operating systems

For Modern Operating Systems:

- Address the needs of applications or fade away

For Operating System Researchers:

- Now a much lower bar for OS adoption
  - In past need both drivers and application support.
- Opportunity for new OSes
  - Target the needs of particular application area
  - Be better in an important area:
    - Security, reliability, performance, manageability

Conclusions

- Virtualization is here and will be everywhere in near future.
  - Cannot handle future multicore without it.
- Large impact on how computing is now
  - Opportunities for architecture help.
  - Opportunities for new system software stacks.