K22 – Operating Systems: Design Principles and Internals

Fall 2025 @dit

Vaggelis Atlidakis

Lecture 21

References: Similar OS courses @Columbia, @Stanford, @UC San Diego, @Brown, @di (previous years); and textbooks: Operating Systems: Three Easy Pieces, Operating Systems: Principles and Practice, Operating System Concepts, Linux Kernel Development, Understanding the Linux Kernel

What is Virtualization?

What is Virtualization?

The ultimate destination: Abstraction of physical host hardware into mu isolated guest machines, called Virtual Machines (VMs)

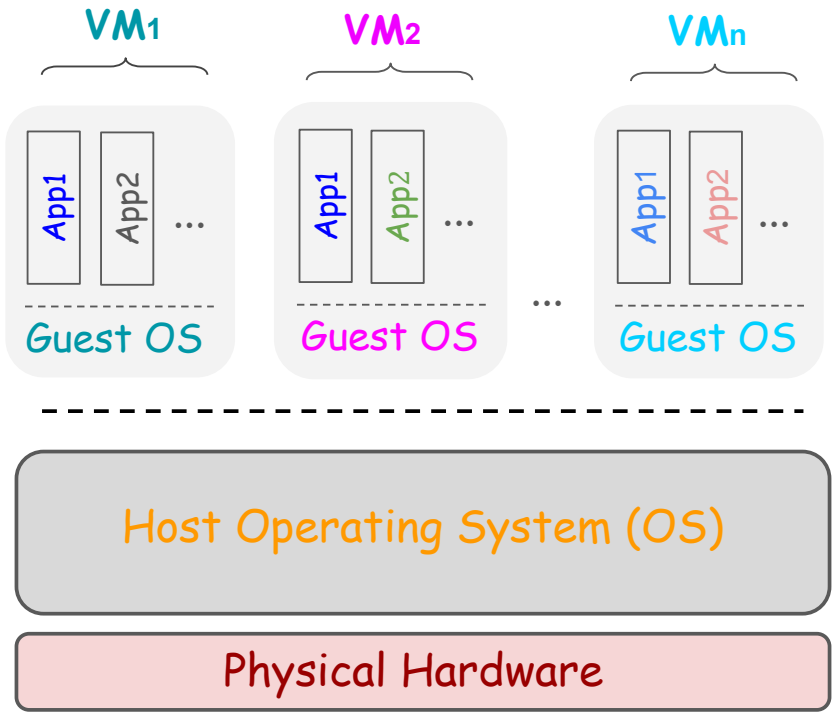
What is Virtualization?

The ultimate destination: Abstraction of physical host hardware into mu isolated guest machines, called Virtual Machines (VMs)

> Each VM can run **any** unmodified guest OS

> Guests OSes cannot distinguish whether they run on a VM, or on a physical machine

What is Virtualization?



Why virtualization?

Why virtualization?

Sustainable model for management of compute

- Want to maintain your servers?
  - Hardware, software, cooling, pay the bills?

Why virtualization?

Sustainable model for management of compute

- Want to maintain your servers?
  - Hardware, software, cooling, pay the bills?
- Electricity paradigm
  - Yeah...you can have your own generator, but would you?

Why virtualization?

Sustainable model for management of compute

> Want to maintain your servers?

- Hardware, software, cooling, pay the bills?

> Electricity paradigm

- Yeah...you can have your own generator, but would you?

Better hardware utilization of large host machines

> Guesses on avg. CPU utilization of major cloud providers?

Why virtualization?

Sustainable model for management of compute

> Want to maintain your servers?

- Hardware, software, cooling, pay the bills?

> Electricity paradigm

- Yeah...you can have your own generator, but would you?

Better hardware utilization of large host machines

> Guesses on avg. CPU utilization of major cloud providers?

- Around 30%

Why virtualization?

Sustainable model for management of compute

- Want to maintain your servers?
  - Hardware, software, cooling, pay the bills?
- Electricity paradigm
  - Yeah...you can have your own generator, but would you?

Better hardware utilization of large host machines

- Guesses on avg. CPU utilization of major cloud providers?
  - Around 30%
- Resource Overcommitment: More virtual than physical CPUs (~10:1)

Why virtualization?

Sustainable model for management of compute

- Want to maintain your servers?
  - Hardware, software, cooling, pay the bills?
- Electricity paradigm
  - Yeah...you can have your own generator, but would you?

Better hardware utilization of large host machines

- Guesses on avg. CPU utilization of major cloud providers?
  - Around 30%
- Resource Overcommitment: More virtual than physical CPUs (~10:1)
- Resource Elasticity
  - VMs on/off on demand (saves customer $$)

Why virtualization?

Sustainable model for management of compute

- Want to maintain your servers?
  - Hardware, software, cooling, pay the bills?
- Electricity paradigm
  - Yeah...you can have your own generator, but would you?

Better hardware utilization of large host machines

- Guesses on avg. CPU utilization of major cloud providers?
  - Around 30%
- Resource Overcommitment: More virtual than physical CPUs (~10:1)
- Resource Elasticity
  - VMs on/off on demand (saves customer $$)
  - VMs live-migrated across hosts (saves provider $$)

Why virtualization?

Fault isolation at machine level

- Bugs in one guest VM do not affect others

Why virtualization?

Fault isolation at machine level

- Bugs in one guest VM do not affect others
- Compromises in one guest VM do not affect others

Why virtualization?

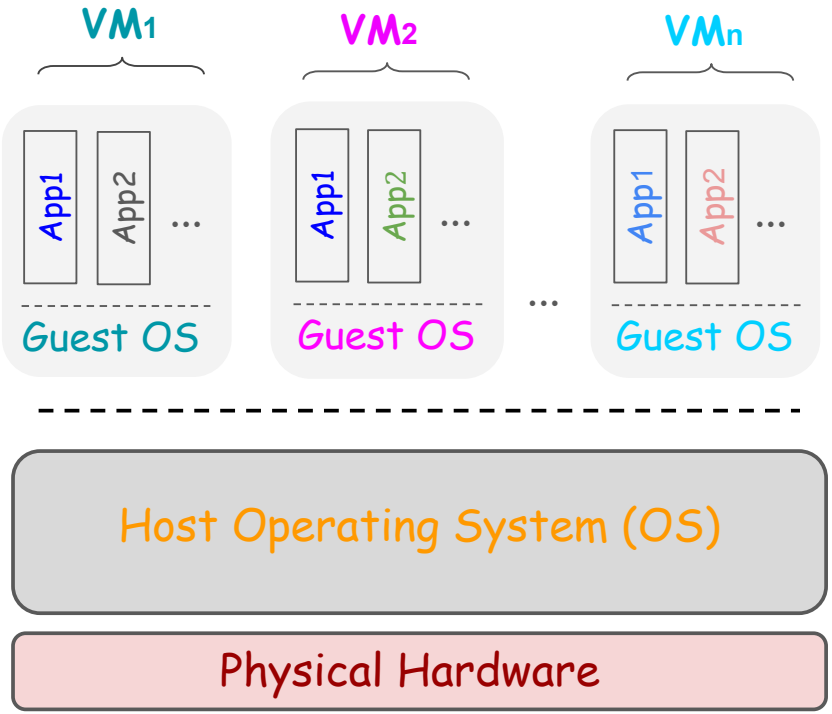
Fault isolation at machine level

- Bugs in one guest VM do not affect others
- Compromises in one guest VM do not affect others

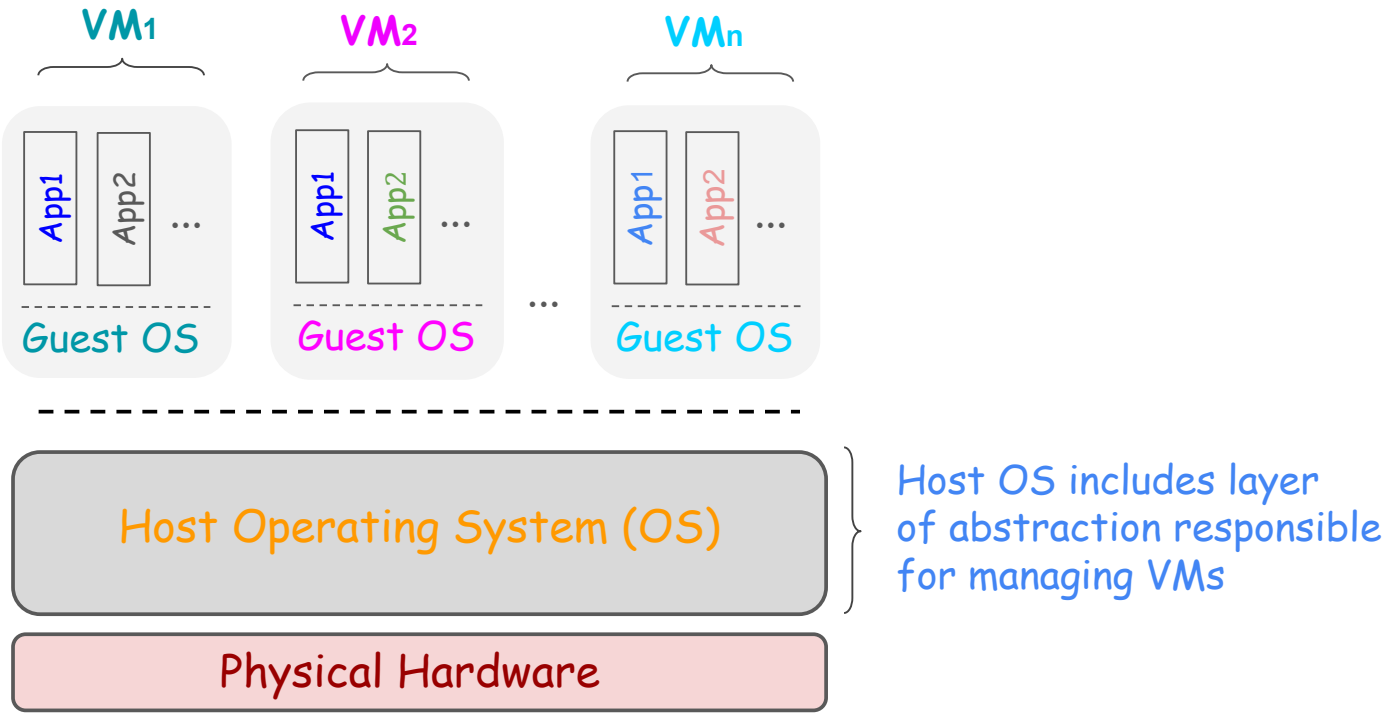
Speeds up development/testing of user-space code

- Develop and test apps for multiple OSes using a single machine

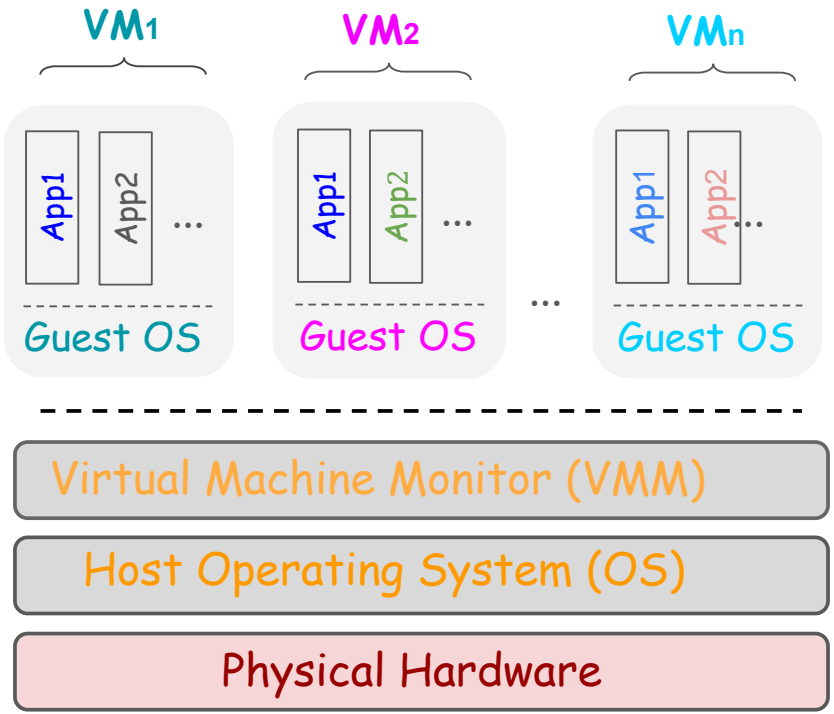
Virtual Machine Monitor (VMM) Approach



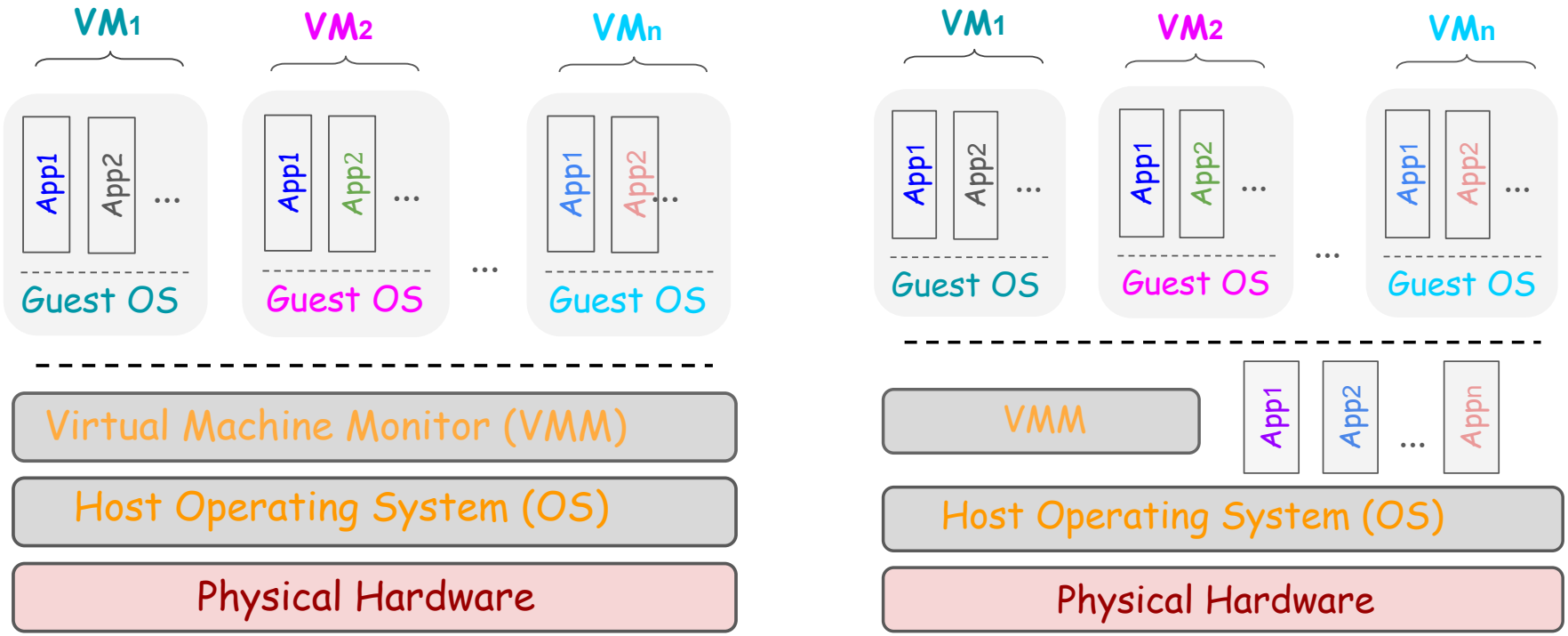
Virtual Machine Monitor (VMM) Approach



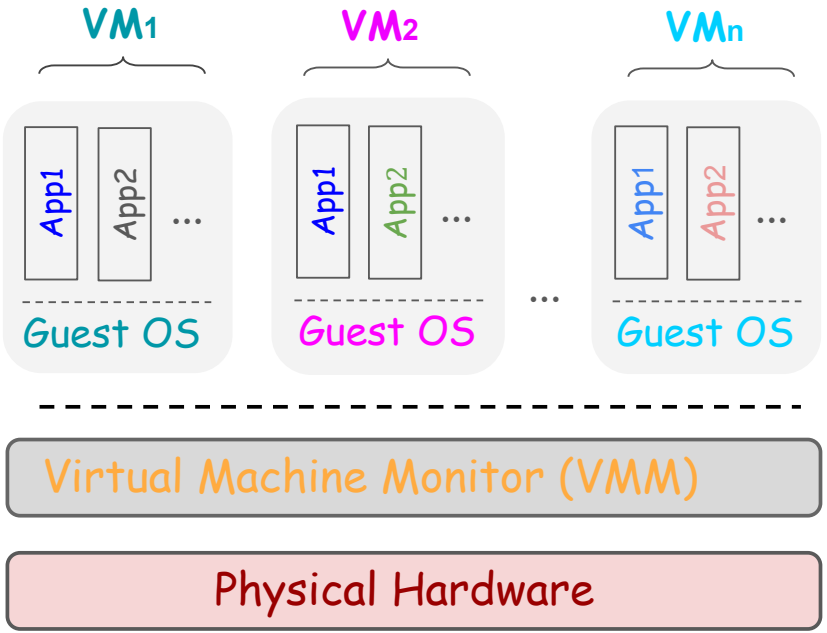
Virtual Machine Monitor (VMM) Approach



Virtual Machine Monitor (VMM) Approach

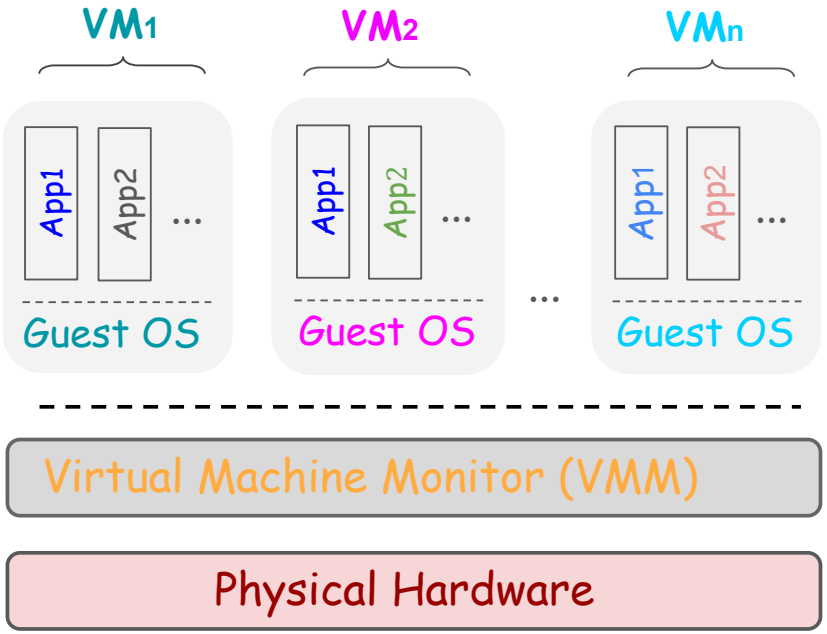


Type-1 Hypervisor (bare metal)



› Characteristics: Small codebase w/ scheduler, mem. management, and interrupt handling

Type-1 Hypervisor (bare metal)

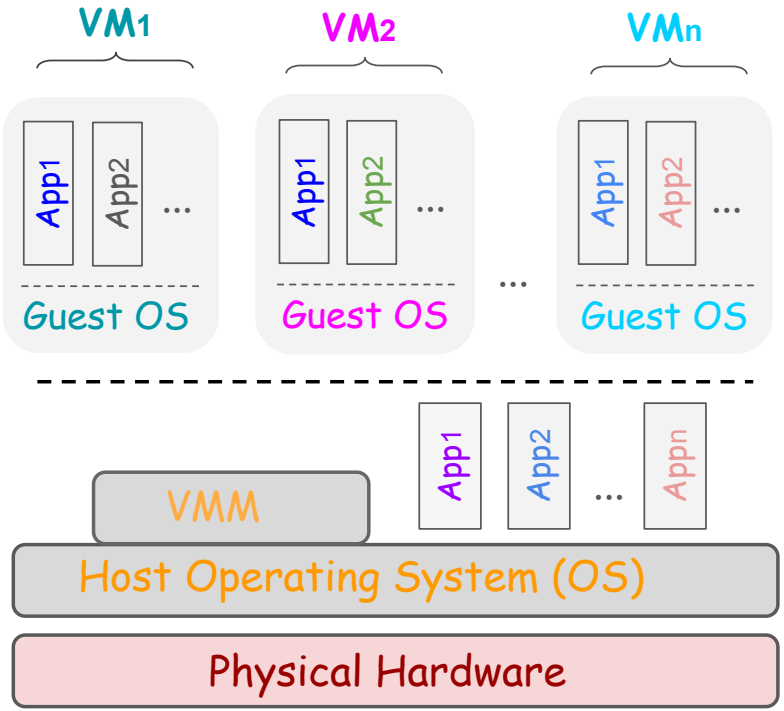


› Characteristics: Small codebase w/ scheduler, mem. management, and interrupt handling

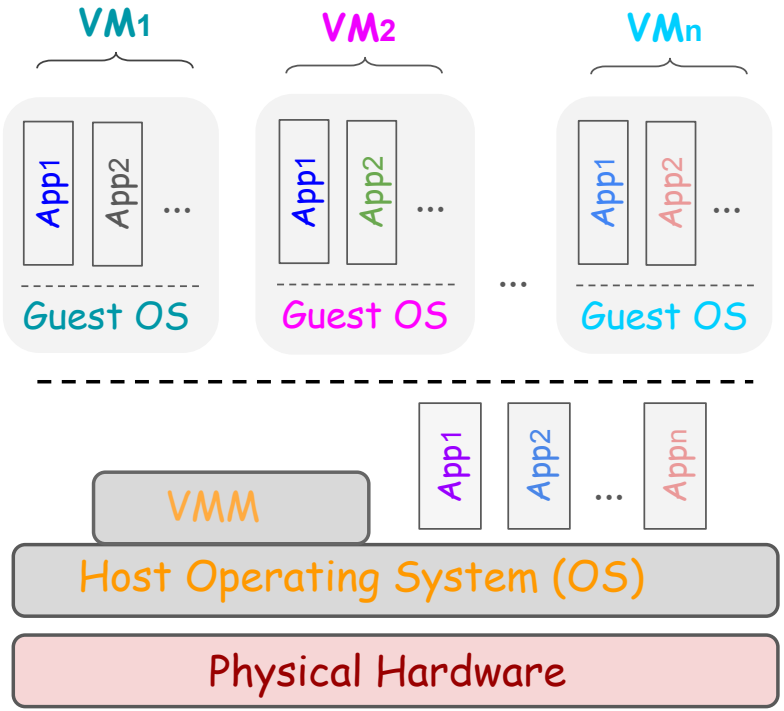
› Popular representatives

- VMWare ESXi (Broadcom)
- Microsoft Hyper-V (Microsoft)
- Xen (University of Cambridge)

Type-2 Hypervisor (hosted)



Type-2 Hypervisor (hosted)



> Popular representatives

- KVM (Linux)
- Parallels (Parallels)
- VirtualBox (Oracle)
- VMware Workstation (Broadcom)

Hypervisors used by Major Cloud Providers

Provider	Hypervisor	Hosts (est.)	VMs (est.)
AWS	KVM (nitro)		
Azure	Hyper-V		
GCP	KVM		
Alibaba	KVM		
Oracle Cloud	KVM		

Hypervisors used by Major Cloud Providers

Provider	Hypervisor	Hosts (est.)	VMs (est.)
AWS	KVM (nitro)	3 Millions	25 Millions
Azure	Hyper-V		
GCP	KVM		
Alibaba	KVM		
Oracle Cloud	KVM		

Hypervisors used by Major Cloud Providers

Provider	Hypervisor	Hosts (est.)	VMs (est.)
AWS	KVM (nitro)	3 Millions	25 Millions
Azure	Hyper-V	2 Millions	20 Millions
GCP	KVM		
Alibaba	KVM		
Oracle Cloud	KVM		

Hypervisors used by Major Cloud Providers

Provider	Hypervisor	Hosts (est.)	VMs (est.)
AWS	KVM (nitro)	3 Millions	25 Millions
Azure	Hyper-V	2 Millions	20 Millions
GCP	KVM	1 Million	10 Millions
Alibaba	KVM		
Oracle Cloud	KVM		

Hypervisors used by Major Cloud Providers

Provider	Hypervisor	Hosts (est.)	VMs (est.)
AWS	KVM (nitro)	3 Millions	25 Millions
Azure	Hyper-V	2 Millions	20 Millions
GCP	KVM	1 Million	10 Millions
Alibaba	KVM	0.8 Million	7 Millions
Oracle Cloud	KVM	0.3 Million	2 Millions

What needs to be virtualized?

› Processor
- Timeshare physical host processor cores to guest VMs
- Handle privileged instructions

› Memory
- Spaceshare host physical mem. to guest VMs physical mem.
- Provision guest page tables; hardware walks them w/o asking…

› Events (Exceptions and Interrupts)
- Vector hardware events to the correct guest VM
- Multiplex external hardware devices

Virtualizing the Processor: Time

VM's CPU cores must run on host's physical CPU cores

Virtualizing the Processor: Time

VM's CPU cores must run on host's physical CPU cores

- The hypervisor time-slices the host's physical cores
- Each VM cores runs for, at most, their dedicated timeslice

Virtualizing the Processor: Time

VM's CPU cores must run on host's physical CPU cores

- The hypervisor time-slices the host's physical cores
- Each VM cores runs for, at most, their dedicated timeslice
- Type-1 hypervisors may use a simple RR scheduler
  - E.g., Xen: sched_init_vcpu

Virtualizing the Processor: Time

VM's CPU cores must run on host's physical CPU cores

- > The hypervisor time-slices the host's physical cores
- > Each VM cores runs for, at most, their dedicated timeslice
- > Type-1 hypervisors may use a simple RR scheduler
  - E.g., Xen: sched_init_vcpu
- > Type-2 hypervisors may use host's kernel sched subsystem
  - E.g., KVM: vmx_vcpu_create
  - Schedulable ctxts (user-space threads) on host's scheduler

Virtualizing the Processor: Privileged Instructions

Full instruction simulation

- Interpret each guest instruction
- Simulate its execution using guest's instructions
- Maintain each VM's state purely in software

Virtualizing the Processor: Privileged Instructions

Full instruction simulation

- Interpret each guest instruction
- Simulate its execution using guest's instructions
- Maintain each VM's state purely in software
- Example: Qemu

Virtualizing the Processor: Privileged Instructions

Full instruction simulation

- > Interpret each guest instruction
- > Simulate its execution using guest's instructions
- > Maintain each VM's state purely in software
- > Example: Qemu

Problem: Too slow

Virtualizing the Processor: Privileged Instructions

Trap-and-emulate

Virtualizing the Processor: Privileged Instructions

Trap-and-emulate

> Execute non-sensitive instruction directly on host

> Privileged instructions? Trap => Hypervisor emulates them

Virtualizing the Processor: Privileged Instructions

Trap-and-emulate

- Execute non-sensitive instruction directly on host
- Privileged instructions? Trap => Hypervisor emulates them
- Need to emulate only a subset of commands

Virtualizing the Processor: Privileged Instructions

Trap-and-emulate

> Execute non-sensitive instruction directly on host

> Privileged instructions? Trap => Hypervisor emulates them

> Need to emulate only a subset of commands

Problem: Not all sensitive instructions cause traps :-(

> E.g., ?

Virtualizing the Processor: Privileged Instructions

Trap-and-emulate

> Execute non-sensitive instruction directly on host

> Privileged instructions? Trap => Hypervisor emulates them

> Need to emulate only a subset of commands

Problem: Not all sensitive instructions cause traps :-(

> E.g., Disabling interrupts on x86

Virtualizing the Processor: Privileged Instructions

Paravirtualization

Virtualizing the Processor: Privileged Instructions

Paravirtualization

> Change the guest OS to cooperate with the hypervisor

Virtualizing the Processor: Privileged Instructions

Paravirtualization

> Change the guest OS to cooperate with the hypervisor

> Provide "hypervisor API" for guest sensitive ops

Virtualizing the Processor: Privileged Instructions

Paravirtualization

> Change the guest OS to cooperate with the hypervisor

> Provide "hypervisor API" for guest sensitive ops

> Modify guest OSes to use hypervisor API

Virtualizing the Processor: Privileged Instructions

Paravirtualization

> Change the guest OS to cooperate with the hypervisor

> Provide "hypervisor API" for guest sensitive ops

> Modify guest OSes to use hypervisor API

Tradeoff: Sacrifice transparency for better performance

Virtualizing the Processor: Privileged Instructions

Hardware-assisted virtualization

Virtualizing the Processor: Privileged Instructions

Hardware-assisted virtualization

- Adding a new privilege level

Virtualizing the Processor: Privileged Instructions

Hardware-assisted virtualization

- Adding a new privilege level
- Unmodified guest OSes run there

Virtualizing the Processor: Privileged Instructions

Hardware-assisted virtualization

- Adding a new privilege level
- Unmodified guest OSes run there
- Sensitive operations cause a VM exit (akin to mode switch)

Virtualizing the Processor: Privileged Instructions

Hardware-assisted virtualization

- Adding a new privilege level
- Unmodified guest OSes run there
- Sensitive operations cause a VM exit (akin to mode switch)
- E.g., Intel VT-x: VMX root vs. non-root mode
  - VMX non-root mode: Ring-0/3: Guest kernel-/user-space
  - VMX root mode: Hypervisor

Virtualizing Memory

Split host's physical mem. on guests' physical mem.

Virtualizing Memory

Split host's physical mem. on guests' physical mem.

> Cannot interpose on a hardware-managed TLB miss

Virtualizing Memory

Split host's physical mem. on guests' physical mem.

> Cannot interpose on a hardware-managed TLB miss

- The walker will transparently walk the page tables
- The hypervisor code has no chance of running

Virtualizing Memory

Split host's physical mem. on guests' physical mem.

> Cannot interpose on a hardware-managed TLB miss

- The walker will transparently walk the page tables
- The hypervisor code has no chance of running
- But…hypervisor must assign host pages to VMs

Virtualizing Memory

Split host's physical mem. on guests' physical mem.

› Cannot interpose on a hardware-managed TLB miss
  - The walker will transparently walk the page tables
  - The hypervisor code has no chance of running
  - But…hypervisor must assign host pages to VMs

› Solution: Shadow page tables

Virtualizing Memory: Shadow Page Tables

- Hypervisor-managed replica of guest page table

Virtualizing Memory: Shadow Page Tables

- Hypervisor-managed replica of guest page table
  - Guest writes x86 %cr3? Privilege operation => Trap

Virtualizing Memory: Shadow Page Tables

- Hypervisor-managed replica of guest page table
  - Guest writes x86 %cr3? Privilege operation => Trap
  - Hypervisor marks guest page table read-only
  - Copies guest page table to shadow page table

Virtualizing Memory: Shadow Page Tables

- Hypervisor-managed replica of guest page table
  - Guest writes x86 %cr3? Privilege operation => Trap
  - Hypervisor marks guest page table read-only
  - Copies guest page table to shadow page table
  - Sets %cr3 point to shadow page table

Virtualizing Memory: Shadow Page Tables

- Hypervisor-managed replica of guest page table
  - Guest writes x86 %cr3? Privilege operation => Trap
  - Hypervisor marks guest page table read-only
  - Copies guest page table to shadow page table
  - Sets %cr3 point to shadow page table
  - Guest will use host's shadow page table w/o knowing

Virtualizing Memory: Shadow Page Tables

> Hypervisor-managed replica of guest page table
  - Guest writes x86 %cr3? Privilege operation => Trap
  - Hypervisor marks guest page table read-only
  - Copies guest page table to shadow page table
  - Sets %cr3 point to shadow page table
  - Guest will use host's shadow page table w/o knowing

> Shadow page tables can be built on demand

Virtualizing Memory: Shadow Page Tables

> Hypervisor-managed replica of guest page table
  - Guest writes x86 %cr3? Privilege operation => Trap
  - Hypervisor marks guest page table read-only
  - Copies guest page table to shadow page table
  - Sets %cr3 point to shadow page table
  - Guest will use host's shadow page table w/o knowing

> Shadow page tables can be built on demand
- Guest reads its page table? Shadow page tbl. used

Virtualizing Memory: Shadow Page Tables

> Hypervisor-managed replica of guest page table
  - Guest writes x86 %cr3? Privilege operation => Trap
  - Hypervisor marks guest page table read-only
  - Copies guest page table to shadow page table
  - Sets %cr3 point to shadow page table
  - Guest will use host's shadow page table w/o knowing

> Shadow page tables can be built on demand
- Guest reads its page table? Shadow page tbl. used
- Guest updates its page table? Shadow page tbl. updated

Virtualizing Events

Need to vector exceptions and interrupts to the correct VM

Virtualizing Events

Need to vector exceptions and interrupts to the correct VM

- Type-2 hypervisors (relatively easier to do)

Virtualizing Events

Need to vector exceptions and interrupts to the correct VM

- Type-2 hypervisors (relatively easier to do)
  - VM exit occurs on interrupts

Virtualizing Events

Need to vector exceptions and interrupts to the correct VM

- Type-2 hypervisors (relatively easier to do)
  - VM exit occurs on interrupts
  - Hypervisor inspects exit reason (see KVM exit reasons)
  - Hypervisor delivers interrupt to vCPU

Virtualizing Events

Need to vector exceptions and interrupts to the correct VM

- Type-2 hypervisors (relatively easier to do)
  - VM exit occurs on interrupts
  - Hypervisor inspects exit reason (see KVM exit reasons)
  - Hypervisor delivers interrupt to vCPU
- Type-1 hypervisors

Virtualizing Events

Need to vector exceptions and interrupts to the correct VM

- Type-2 hypervisors (relatively easier to do)
  - VM exit occurs on interrupts
  - Hypervisor inspects exit reason (see KVM exit reasons)
  - Hypervisor delivers interrupt to vCPU
- Type-1 hypervisors
  - Full virtualization: Hypervisor emulates interrupt handlers' logic
  - Paravirtualization: Modify guest OSes w/ virtual interrupt event queue

Virtualizing Events

Need to vector exceptions and interrupts to the correct VM

- Type-2 hypervisors (relatively easier to do)
  - VM exit occurs on interrupts
  - Hypervisor inspects exit reason (see KVM exit reasons)
  - Hypervisor delivers interrupt to vCPU
- Type-1 hypervisors
  - Full virtualization: Hypervisor emulates interrupt handlers' logic
  - Paravirtualization: Modify guest OSes w/ virtual interrupt event queue
  - Hardware support: Interrupt controller delivers events to guest OSes

AWS Case Study: Netflix

AWS Case Study: Netflix

Seriously large company: 0.36% of S&P500

AWS Case Study: Netflix

Seriously large company: 0.36% of S&P500

- Netflix market cap was $413B (as of yesterday)
- The GDP of Greece is ~$235B

AWS Case Study: Netflix

Seriously large company: 0.36% of S&P500

- Netflix market cap was $413B (as of yesterday)
- The GDP of Greece is ~$235B

Data is based on public resources [1,2,3,4]

- Phenomenal scale, even if data is a bit off

AWS Case Study: Netflix

Seriously large company: 0.36% of S&P500

- Netflix market cap was $413B (as of yesterday)
- The GDP of Greece is ~$235B

Data is based on public resources [1,2,3,4]

- Phenomenal scale, even if data is a bit off
- Runs almost entirely on AWS using >100,000 VMs

AWS Case Study: Netflix

Seriously large company: 0.36% of S&P500

- Netflix market cap was $413B (as of yesterday)
- The GDP of Greece is ~$235B

Data is based on public resources [1,2,3,4]

- Phenomenal scale, even if data is a bit off
- Runs almost entirely on AWS using >100,000 VMs
- ~15% of global downstream internet traffic

AWS Case Study: Netflix

Seriously large company: 0.36% of S&P500

- Netflix market cap was $413B (as of yesterday)
- The GDP of Greece is ~$235B

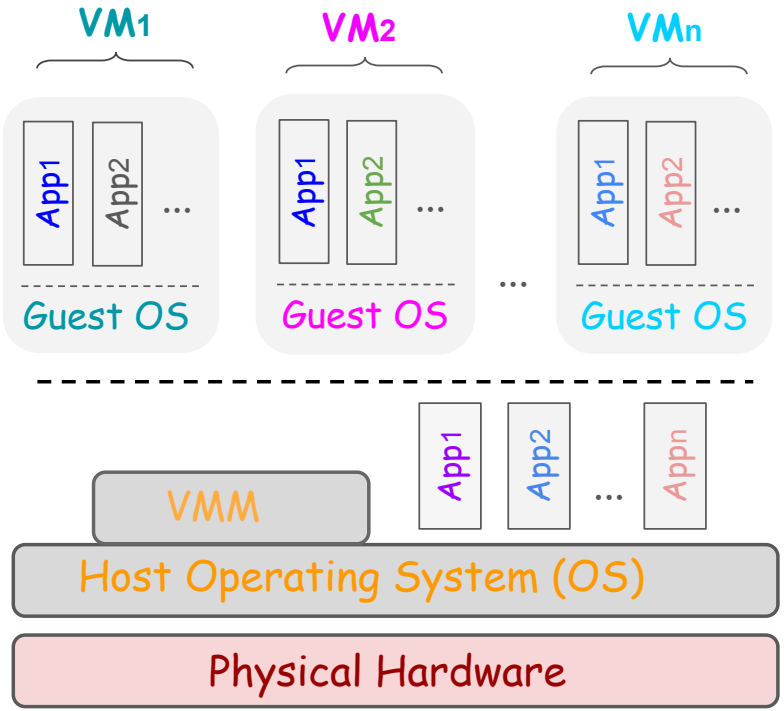
Data is based on public resources [1,2,3,4]

- Phenomenal scale, even if data is a bit off
- Runs almost entirely on AWS using >100,000 VMs
- ~15% of global downstream internet traffic
- Serves >1,000 PB / day of video streaming

Hypervisors used by Major Cloud Providers

Provider	Hypervisor	Hosts (est.)	VMs (est.)
AWS	KVM (nitro)	3 Millions	25 Millions
Azure	Hyper-V	2 Millions	20 Millions
GCP	KVM	1 Million	10 Millions
Alibaba	KVM	0.8 Million	7 Millions
Oracle Cloud	KVM	0.3 Million	2 Millions

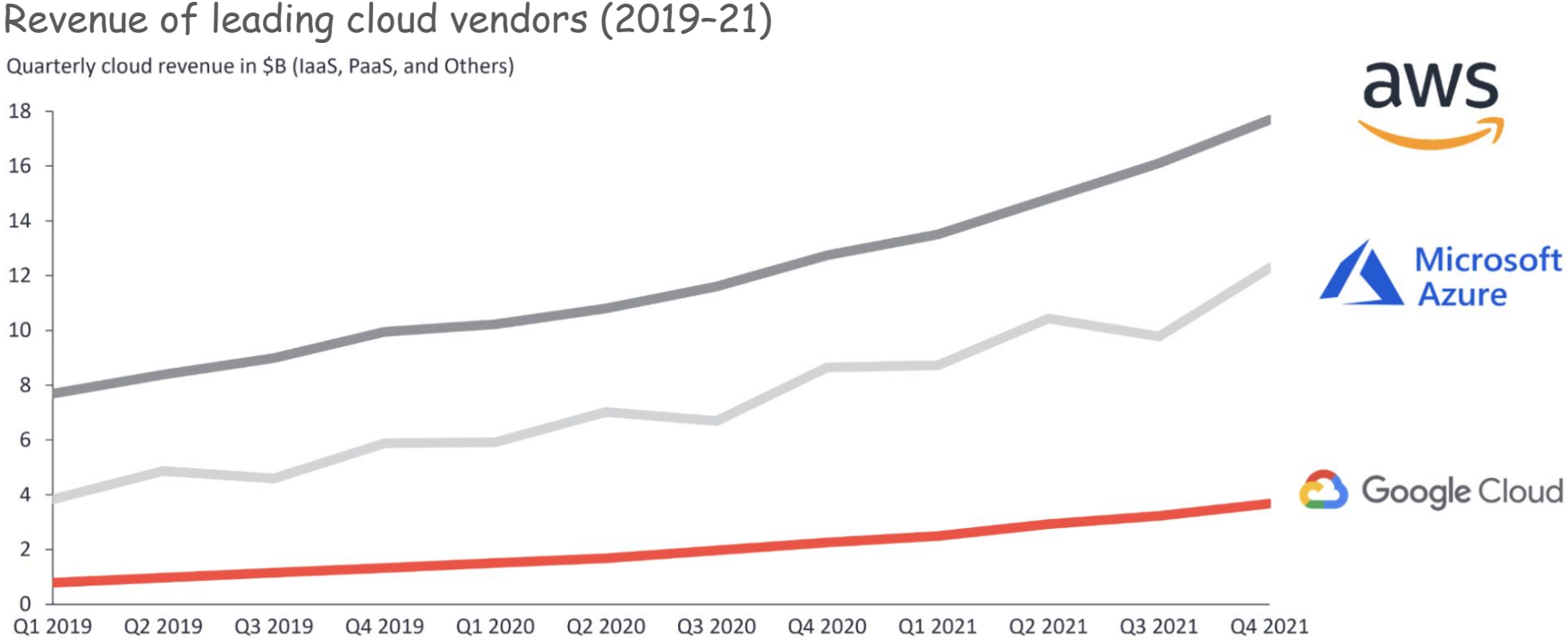
Type-2 Hypervisor



> Popular representatives

- KVM (Linux)
- Parallels (Parallels)
- VirtualBox (Oracle)
- VMware Workstation (Broadcom)

Why study OSes?



[1] https://iot-analytics.com/cloud-market/