Cloud Computing Applications and Services

(Aplicações e Serviços de Computação em Nuvem)

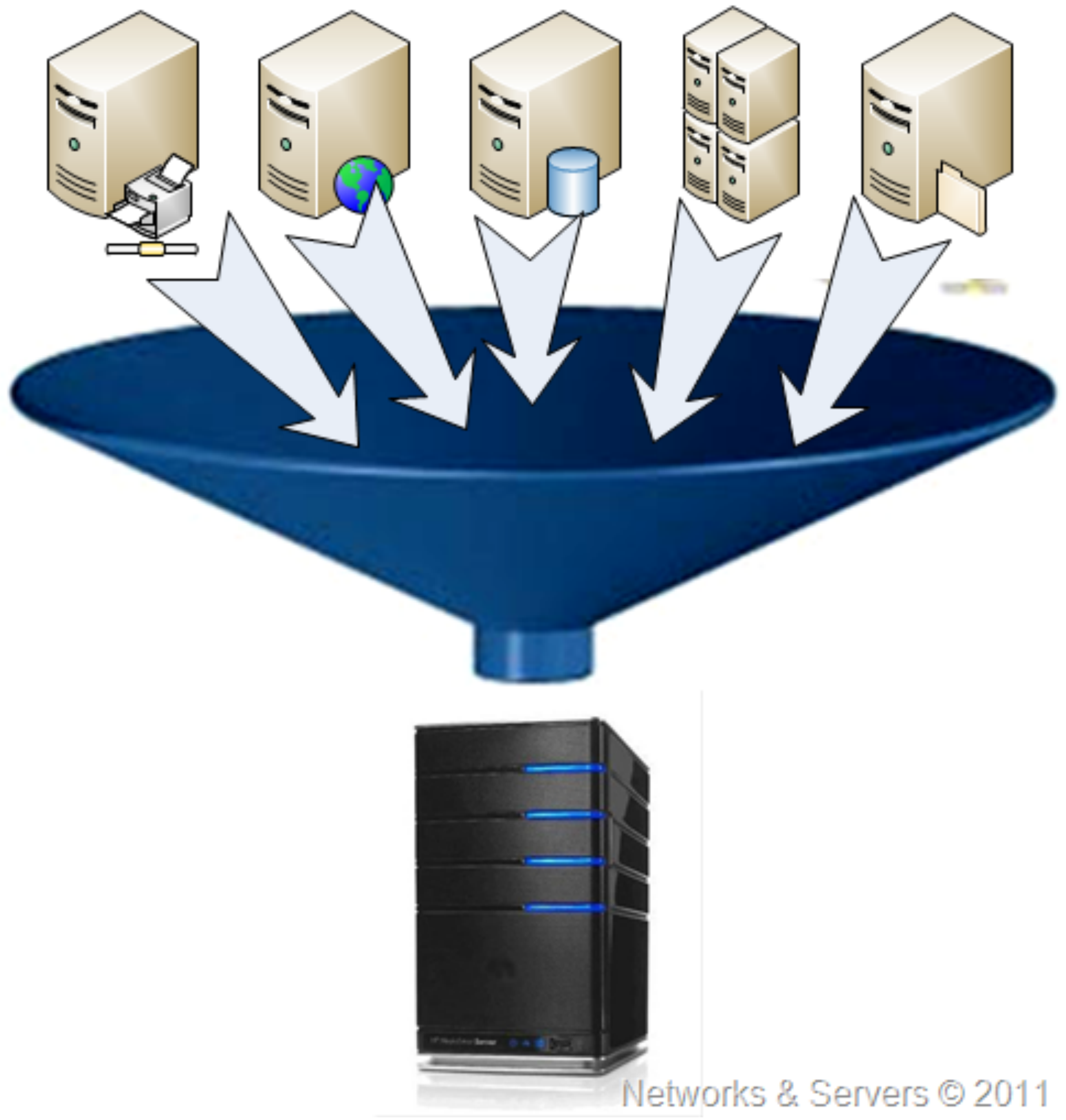
Virtualization (Part I)

University of Minho

2024-2025

Virtualization
Definition and examples

- Technique that allows creating a software-based virtual device or resource that, in practice, is an **abstraction** provided on top of existing hardware or software resources
- Examples:
  - Virtual Machines (VMs)
  - Virtual Networks
  - Virtual Memory
  - Logical Storage Volumes
  - …



Virtualization in Practice

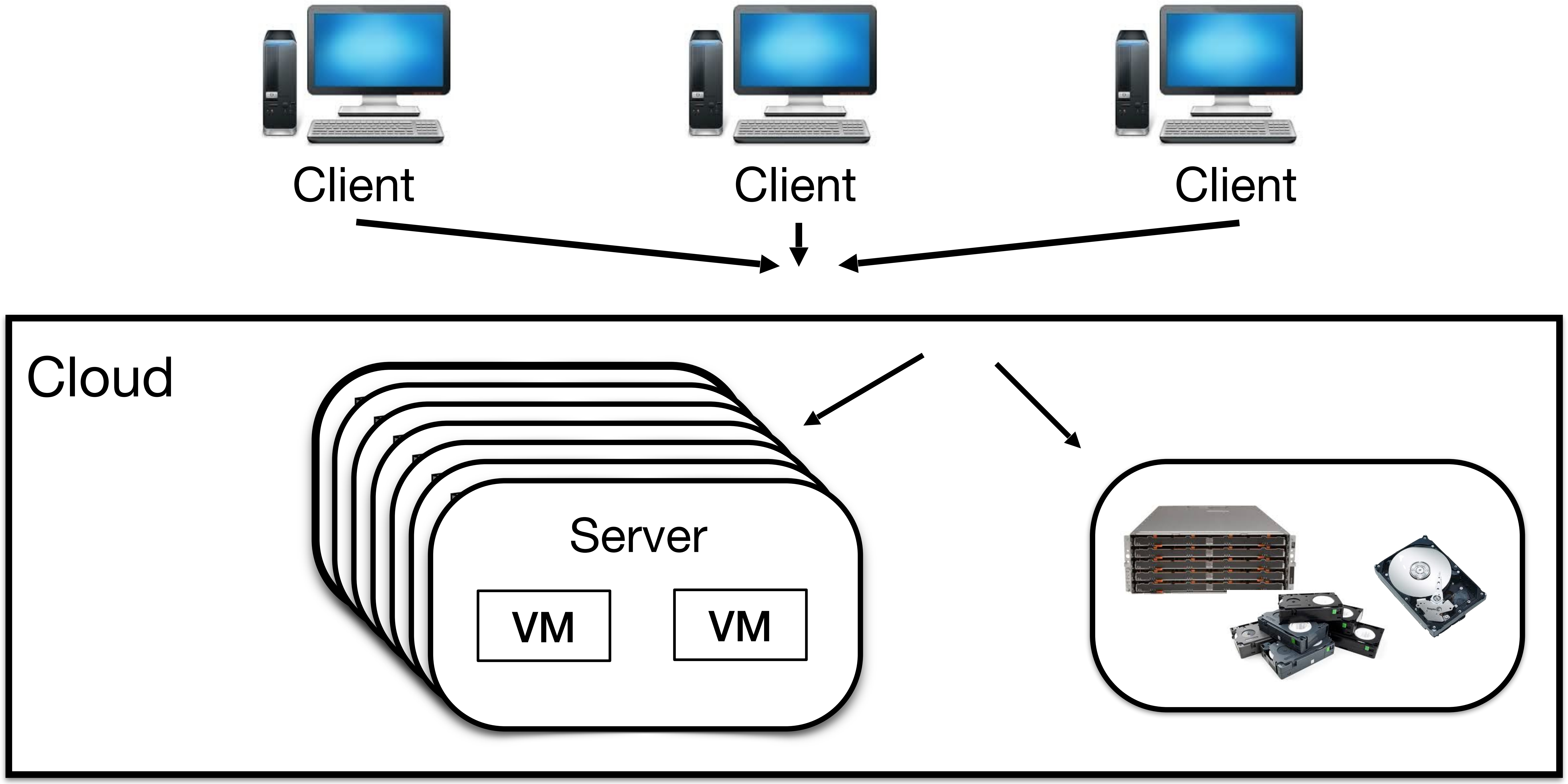
Virtual Desktop Infrastructures (VDIs)

- User desktop environments (terminals) are provided by a central service (e.g., VMs running on a cluster of servers)

<u>Examples</u>: VMWare Horizon 7, Amazon WorkSpaces

Virtualization in Practice

Simplified Cloud Deployment



<u>Examples</u>: Amazon EC2, Google Compute Engine, …

Advantages

Heterogeneity

- Hardware devices (physical resources) are **highly heterogeneous**
  - Different models of the same hardware type (e.g., CPU, GPU, RAM, disk, network models) may have different interfaces, drivers, …
- Virtualization can be used to **abstract** this heterogeneity and provide unified virtual resources
  - E.g., virtual CPUs, virtual disks, virtual networks, …
  - Virtual resources present a single (unified) interface, independently from the underlying hardware model being virtualized

Advantages

Transparency

- User interaction with virtual resources is similar to the interaction with a physical one
  - E.g., when you connect through ssh to a remote machine, the interaction is identical if you are using a VM or a bare-metal server
- **Transparency** in this context means that users do not need to change their approach (e.g., commands, programs, scripts, …) when using virtual resources

Advantages

Isolation

- Virtual resources sharing a physical resource must be **isolated**
- **Security**
  - We don't want the VM of one user accessing/modifying the memory of VMs from other users. Such could lead to nasty attacks and memory corruption
- **Performance**
  - The VM of one user should not compromise the performance of other VMs sharing the same physical resources (e.g., CPU, memory, disk, network)
- **Failures**
  - The failure of one VM should not lead to the failure of other VMs at the host

Advantages

Consolidation and management

- **Consolidation** of physical resources allows lowering costs and making better use of available hardware
  - A single server can be virtualized to, for example, run multiple operating systems (with VMs)
- **Managing** virtual resources is typically easier and more flexible than managing physical ones
  - E.g., VMs are easier/quicker to set up, destroy, migrate, …

Disadvantages

Performance and Over provisioning

- Virtualization often adds a **performance** penalty to applications/services, when compared to directly using the physical resources
  - The mechanisms used to abstract the physical hardware must perform additional tasks when mediating virtual requests into the actual hardware
- Increasing the number of virtual resources being served by the same hardware may lead to **over provisioning** and performance degradation (i.e., saturation of physical resources)
  - E.g., when multiple VMs, running in the same server, require more than the available amount of CPU cores, memory, disk/network bandwidth, …

Disadvantages

Security and Dependability

- If **isolation** is not properly addressed or, a malicious user has privileged access to the physical resources, the **security** of all virtualized resources may be compromised
  - E.g., a system administrator, with root access to a server, may compromise all VMs running there
- The failure of a single physical resource may compromise the **dependability** of several virtual resources using it
  - E.g., the failure of a server will lead to the failure of all VMs running there

Summary

- Note that the advantages and disadvantages discussed previously apply to different types of virtualization
  - Most of the previous examples use VMs but the same properties hold true for memory, network, and storage virtualization, for instance
  - **Homework:** Check that the previous properties also apply to other types of virtualization

- Now let's talk a bit more about Virtual Machines!

Virtual Machines

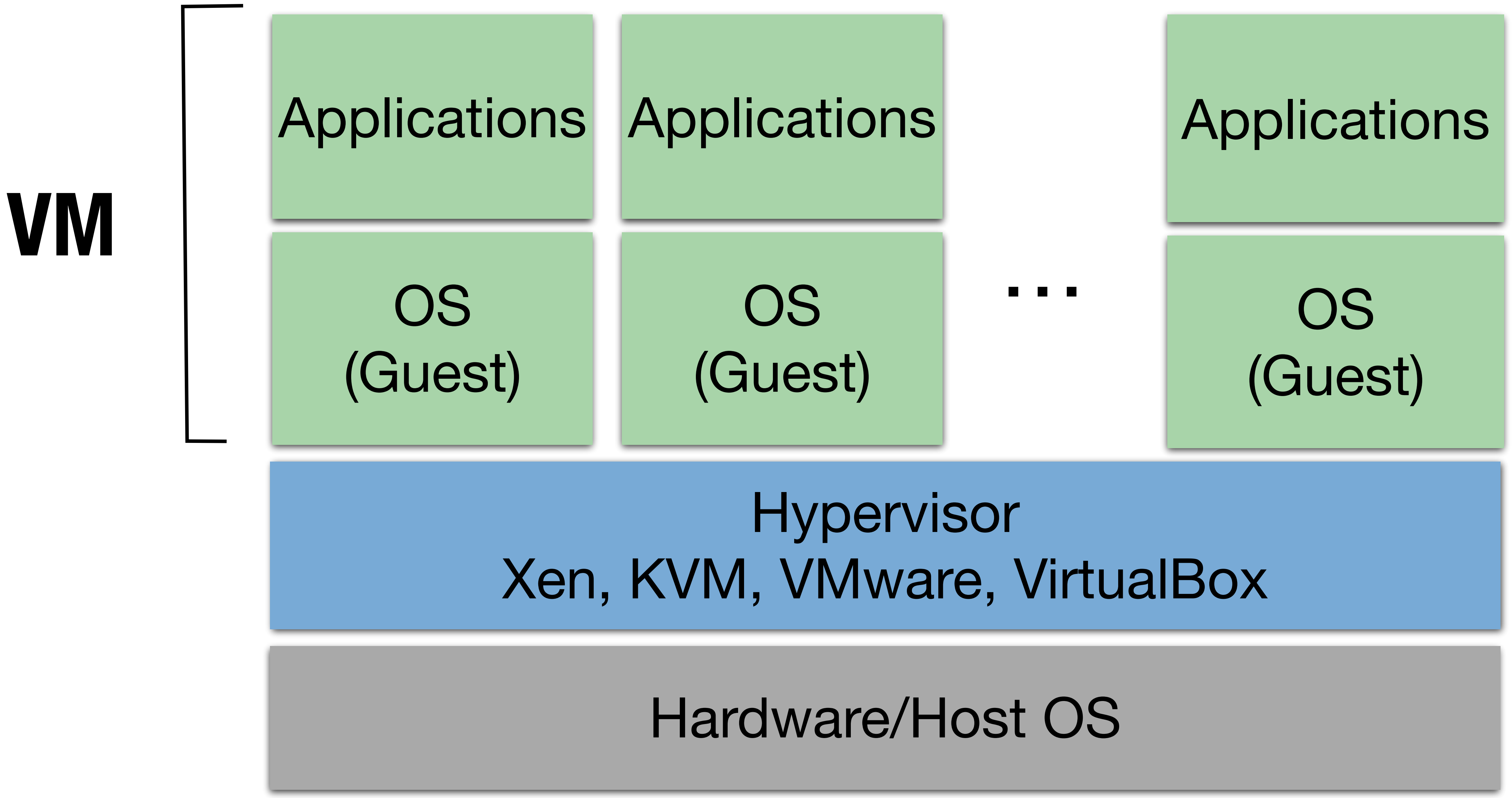
Context and motivation

- IBM mainframe systems (from about 45 years ago) allowed applications to use isolated portions of a given system's resources
- Virtualization became mainstream in the early 2000's with the X86 server architecture due to
  - Infrastructure costs (i.e., servers were expensive…)
  - Under-utilized resources (i.e., servers were not being used to their maximum capacity)
- Further, changing an application / service to run on different Operating Systems (OSs) is a costly and hard task

Virtual Machines

Definition and architecture

- Virtual Machines (VMs) allow running multiple OS flavors on top of the same server (simplistic definition)
- **Guest OS** (*i.e.*, VM) instructions are intercepted, translated, and executed on the **Host's OS and hardware**
  - CPU
  - RAM
  - Disk
  - Network
  - …



Virtual Machines

Hypervisor

Hypervisor
Xen, KVM, VMware, VirtualBox

- Also known as Virtual Machine Monitor (VMM)
  - Xen, KVM, VMware, and VirtualBox are examples of hypervisors
- The hypervisor controls the low-level interaction between VMs and the underlying host's OS and hardware
  - Provides access to the host's CPU, RAM, disk and network hardware
- But, how are physical resources **shared** and **accessed** by the VMs?

Virtual Machines

Host's CPU

- *Time slicing* - processing requests are sliced up and shared across VMs
- Similar to running multiple processes in the host OS
  - Remember the Operating Systems classes?
- **Caution!** Overcommitting vCPUs may lead to poor performance

Virtual Machines

Host’s RAM and Persistent Storage

- Each VM allocates a specific portion of the host’s RAM (memory) and persistent storage (*e.g.*, SSD, HDD) capacity
- Memory management mostly uses traditional OS mechanisms
  - Paging, Translation Lookaside Buffer (TLB), …
- Persistent storage is shared across VMs
  - must handle multiple writers/readers efficiently
  - can be allocated as required (*i.e.*, thin-provisioning)

Virtual Machines

Host’s Network

- VMs share the host’s network bandwidth and can be configured with different network setups
  - **Host-only**: Shares the host’s networking namespace. The VM only has access to the host
  - **Nat**: Masks network activity as if it is done by the host (single network identity). The VM has access to external resources
  - **Bridge**: Uses the hypervisor to assign a specific IP to the VM. The VM is seen as another node in the physical network

Virtualization Modes

Full Virtualization

- Guest OS is fully abstracted from the underlying host's hardware (*e.g.*, VirtualBox)
- **Advantage:** No modifications to the guest OS means higher range of supported OS flavors, and easier migration/portability of VMs
- **Disadvantage:** all guest OS instructions must be translated by the hypervisor leading to potentially lower I/O and CPU performance
  - Hardware-assisted virtualization leverages **specific hardware** to reduce the performance penalty of instruction translation (*e.g.*, Intel VT-x, AMD-V)

Virtualization Modes

Paravirtualization

- Requires hooks/modifications at the guest OS to bypass the translation of costly OS instructions (*e.g.*, Xen)
- **Advantage:** Better CPU and I/O performance as the guest OS (i.e., cost of request translation is reduced)
- **Disadvantage:** Guest OS must be modified, which is worst for maintainability and portability
(i.e., one must use modified OS images to run VMs)

Virtualization Types

Type 1 - Bare Metal Hypervisor

- The hypervisor does not require a general-purpose OS at the host server (*e.g.*, VMware ESX)
  - The hypervisor is deployed directly on hardware as a “small operating system”
- Good performance (the small OS is optimized for virtualization purposes!) but it usually requires specific virtualization support at the hardware level

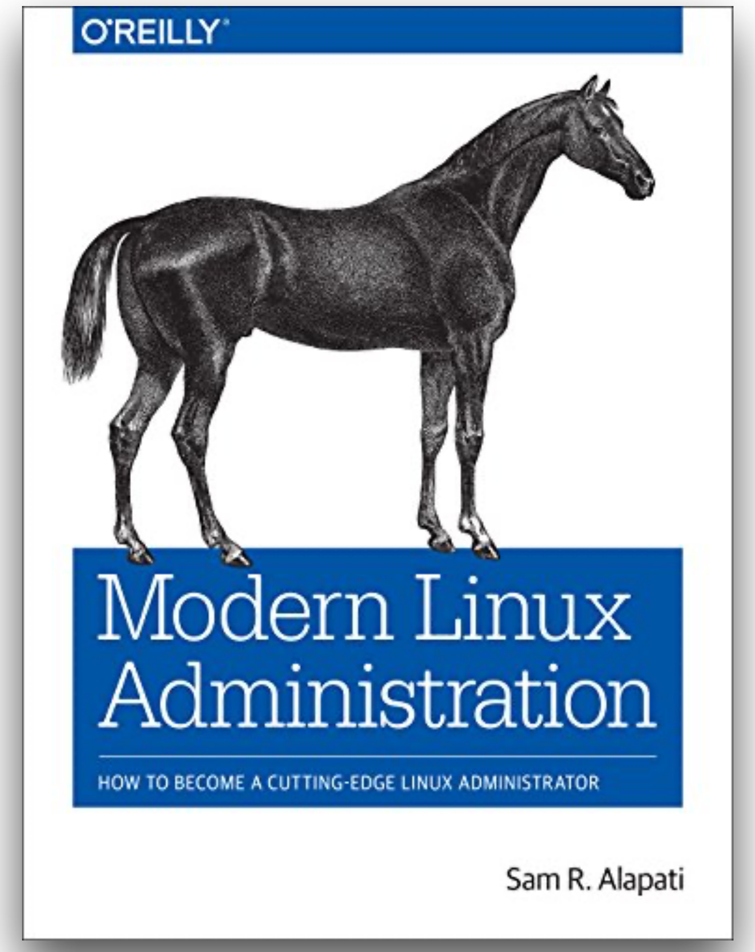
Virtualization Types

Type 2 - Hosted Hypervisor

- The hypervisor is deployed on top of a general-purpose OS (*e.g.*, VirtualBox, KVM, XEN)
  - **Note:** many of these hypervisors still require installing specific kernel modules on top of general-purpose OSs
- Worst performance… The OS is not optimized for virtualization purposes

Further Reading

- S. Alapati. ***Modern Linux Administration: How to Become a Cutting-edge Linux Administrator***. O'Reilly, 2016
- Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. ***Xen and the art of virtualization***. SIGOPS Operating Systems, 2003.



Questions?