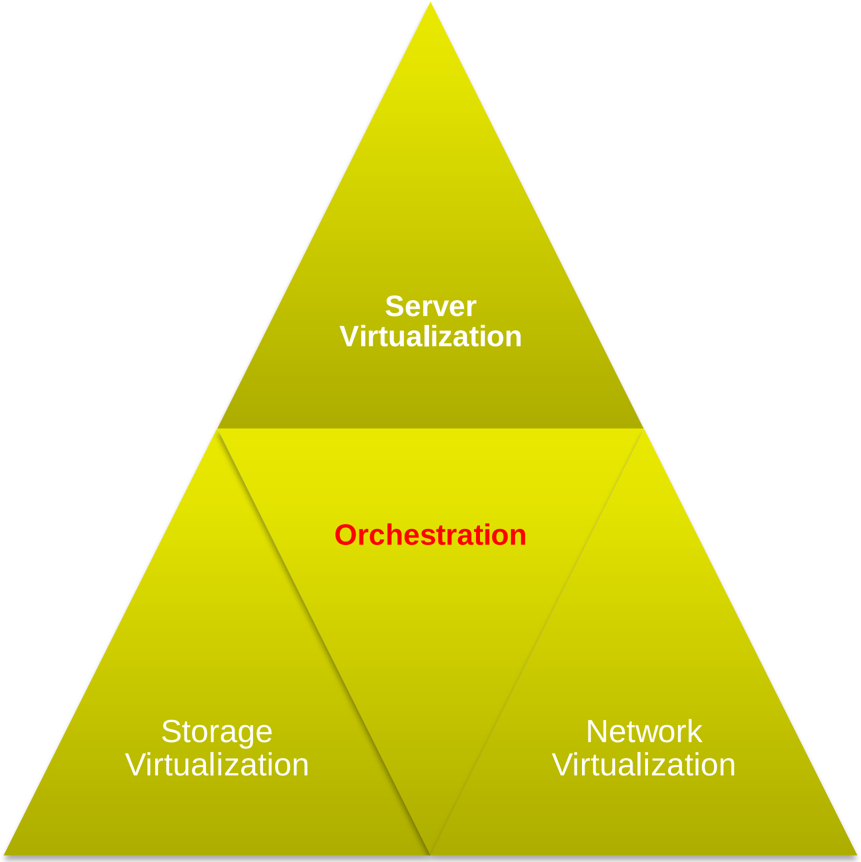
SIC

Serviços e Infraestruturas de Comunicação

Server Virtualization

SIC & the 3 vertices of datacenter virtualization…



today's menu special: Server Virtualization

Virtualization – key concepts

What is virtualization?

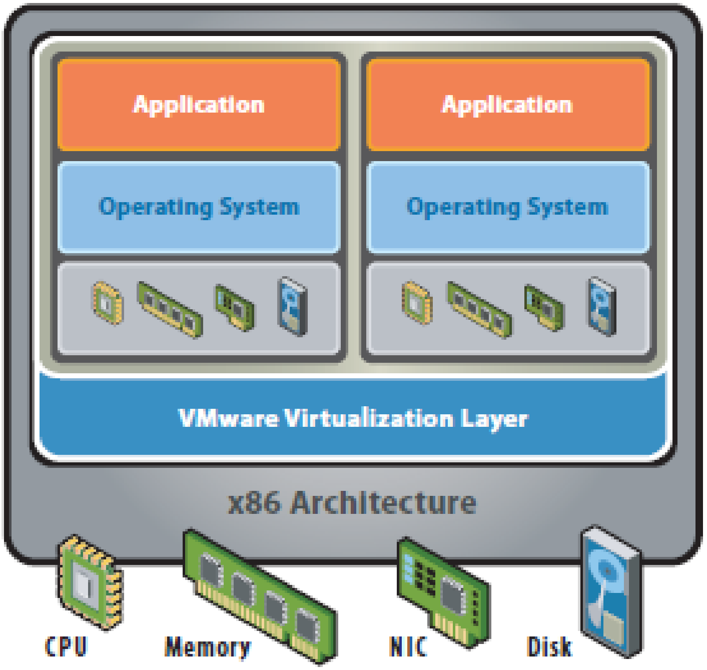
Frequently found definitions:

- *“Virtualization is a technique for* ***hiding the physical characteristics of computing resources from the way in which other systems, applications, or end users interact with those resources****. This includes making a single physical resource (such as a server, an operating system, an application, storage device, or network) appear to function as multiple logical resources; or it can include making multiple physical resources (such as storage devices or servers) appear as a single logical resource.”*
- *“A layer mapping its visible interface and resources onto the interface and resources of the underlying layer or system on which it is implemented.”*

Virtualization – key concepts

A few examples of virtualization:

- Virtual memory
- Virtual Local Area Networks (VLANs)
- Virtual Private Networks (VPNs)
- Virtual execution environments (e.g., JVM)
- Virtual **storage** volumes
- Virtual desktops
- Virtual **servers**



Virtualization – key concepts

What is a "*Virtual Machine* (VM)"?

- Execution environment that supports the execution of code (operating system, applications) as a physical machine.

Process Virtual Machines:

- VM is an application installed on top of an operating system, offering applications a uniform execution environment (hardware abstraction, sandboxing).
- Well-known example: *Java Virtual Machine.*

System Virtual Machines:

- Allows multiplexing hardware resources between multiple operating systems (VMs).

Virtualization – key concepts
System Virtual Machines

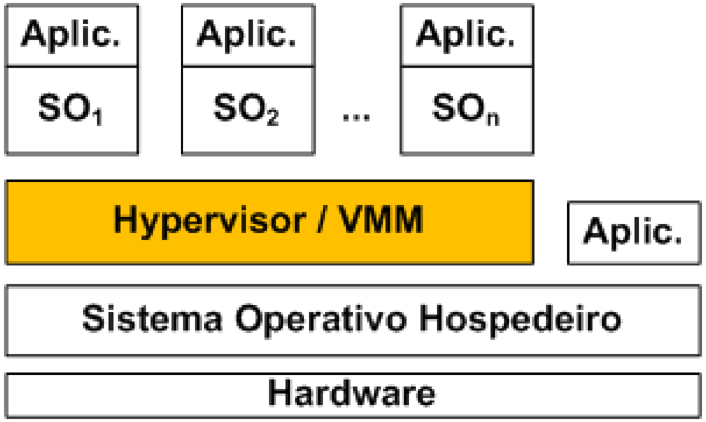
- The same hardware platform supports the simultaneous execution of multiple systems (VMs).
- The hardware is shared across multiple VMs.
- The various VMs are isolated from each other.
- Hardware itself can be virtualized: the hardware components that each VM “sees” (CPU, RAM, NICs, *storage*, etc.) may not correspond to the characteristics of existing physical hardware.
  [remember the CloudSigma VM creation menus discussed last week?]

- Software that controls the operation of VMs:
  - **Hypervisor, Virtual Machine Monitor (VMM)**

Virtualization – key concepts
System Virtual Machines

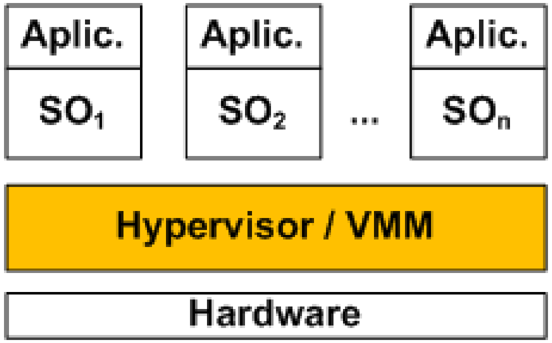
Hosted System Virtual Machines:

- The *hypervisor* works as a normal application on top of a “host” operating system. VMs execute on top of the hypervisor application.
- Also known as **Type 2 Hypervisor**.



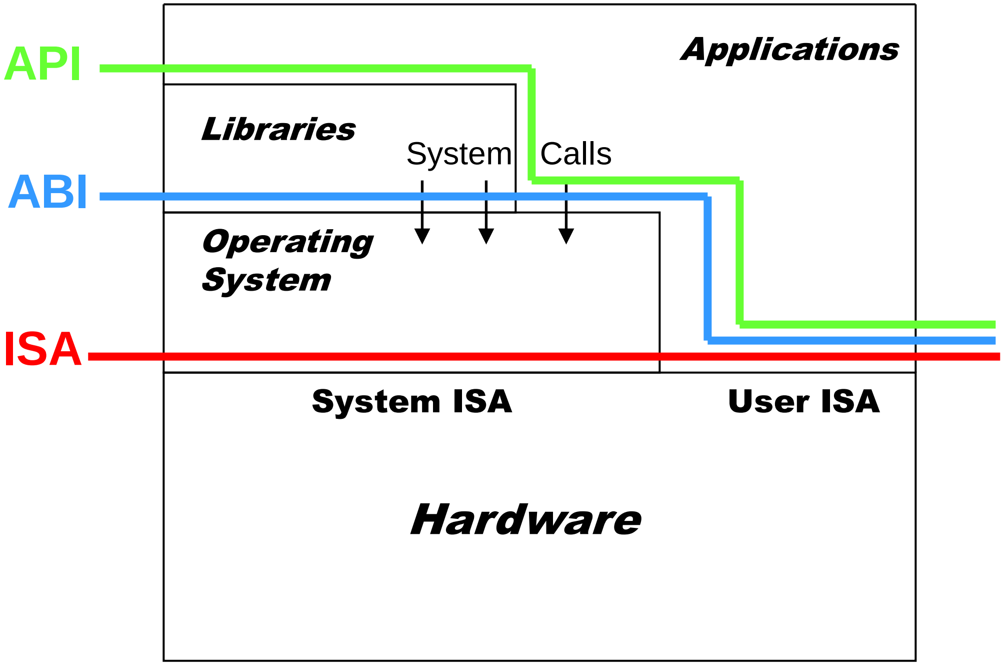
Native System Virtual Machines

- The *hypervisor* works directly on top of native hardware (bare metal installation).
- Also known as **Type 1 Hypervisor**.



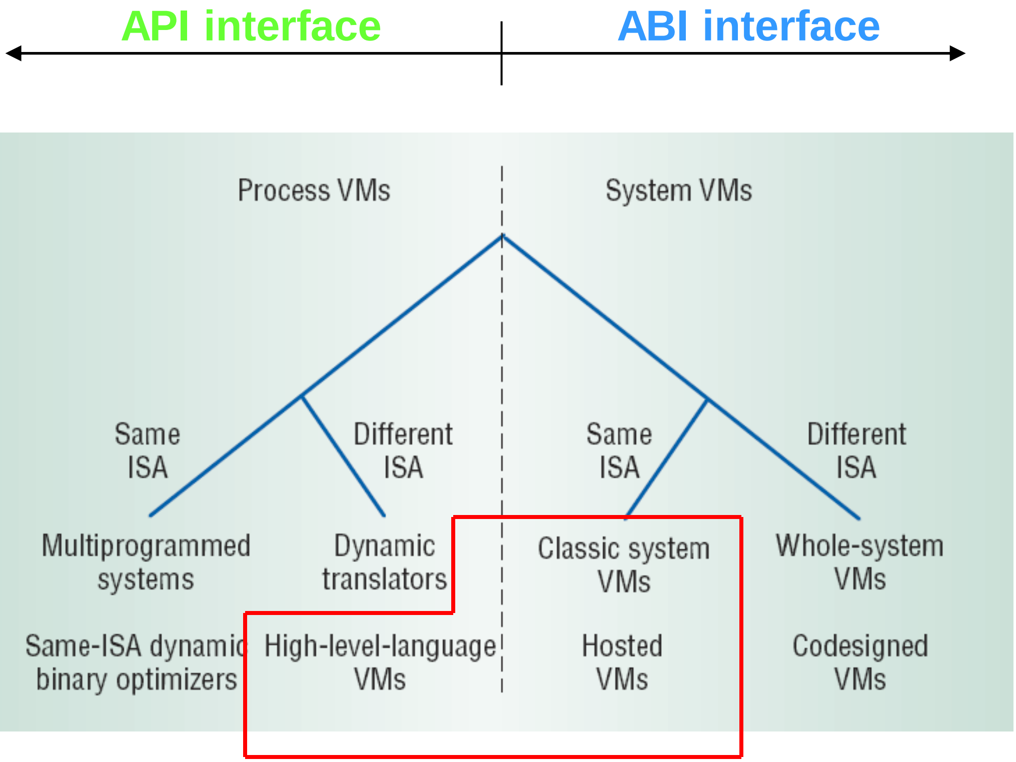
Virtualization – key concepts

Server Architecture & Interfaces



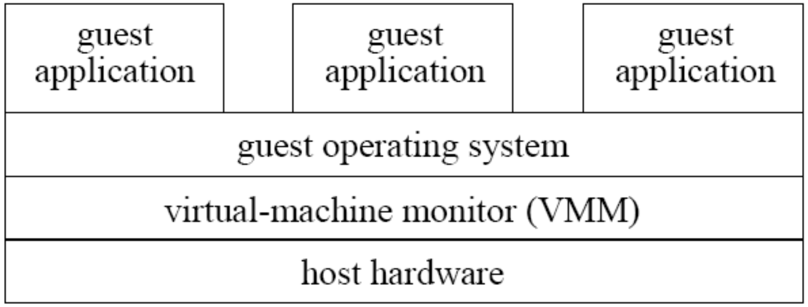
- **API** – Application Programming Interface
- **ABI** – Application Binary Interface
- **ISA** – Instruction Set Architecture

Virtualization – key concepts
Another way to look at system VMs



Source: Mariano Diaz, VirginiaTech, http://courses.cs.vt.edu/cs5204/fall09-kafura/Student-Presentations/Virtualization-Concepts-Diaz.ppt

Virtualization – key concepts
Another way to look at system VMs



guest application

guest application

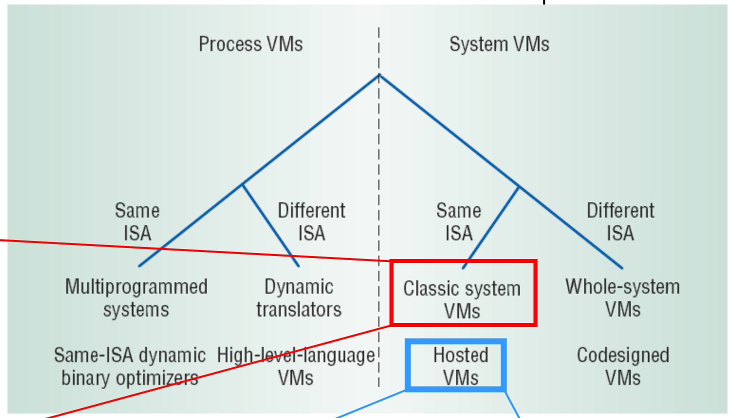
guest application

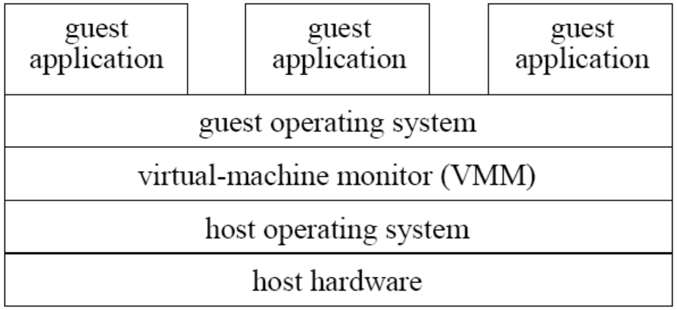
guest operating system

virtual-machine monitor (VMM)

host hardware

Type 1 – Native System VM





guest application

guest application

guest application

guest operating system

virtual-machine monitor (VMM)

host operating system

host hardware

Type 2 – Hosted System VM

Source: Mariano Diaz, VirginiaTech

Virtualization – key concepts
An old technology…

IBM VM/370

- Concurrent execution of multiple production operating systems
- Testing and development of experimental systems
- Adoption of new systems with continued use of legacy systems
- Ability to accommodate applications requiring special-purpose OS

- Introduced notions of *handshake* and *virtual-equals-real mode* to allow sharing of resource control information with the hypervisor
- Leveraged ability to co-design hardware, VMM, and guest OS

Want to know more about VM/370 and its predecessors?

https://share.confex.com/share/119/webprogram/Handout/Session11255/45%20Years%20of%20Mainframe%20Virtualization%202012-08-05.pdf

Virtualization – key concepts
An old technology… reloaded!

- Server/workload consolidation
- Compatible with evolving multi-core architectures
- Simplifies software distributions for complex environments
- Easy server (VM migration)
- Improved resource management and efficiency
- Workload (VM) isolation

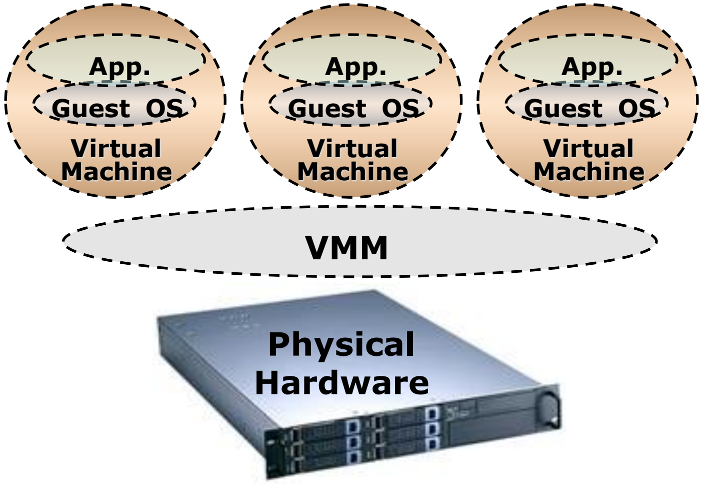
vmware®

KVM

Microsoft
Hyper-V

Xen™

ORACLE VirtualBox



Virtualization – key concepts
Why the revival of virtualization?

- Virtualization made sense in the 60’s and 70’s, when a single and insanely expensive mainframe was a scarce resource that needed to be shared across different applications and users.

- In the 80’s and the 90’s computing power went to the x86 PC. Hardware was apparently cheap and accessible to everyone.

- In the 2000’s the number of servers in the data center was increasing at unsustainable rates, resulting in the urging need for more efficient management of resources (energy, space, O&M, hardware costs…). At the same time: (i) the server hardware was more mature and virtualization-friendly, and (ii) faster networks and distributed computing technologies allowed higher centralization of resources.

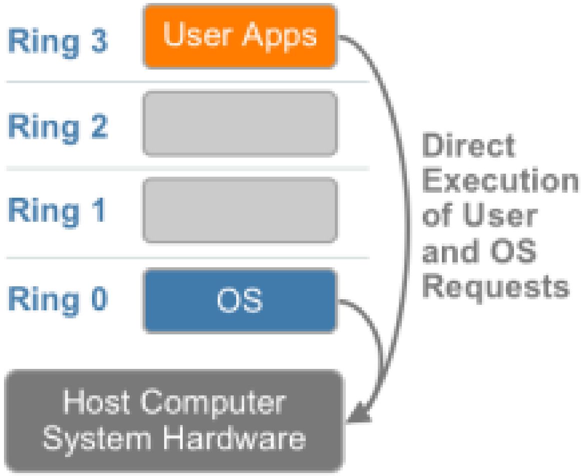
Virtualization – key concepts
How to virtualize the x86 platform?

Typical OS is designed to work at ring 0

Running the OS at ring 0 does not ensure proper isolation between multiple OS's (the VMM needs to "control" guest OS)

Simply moving the OS to ring 1 to create space for the VMM at ring 0 is not straightforward:

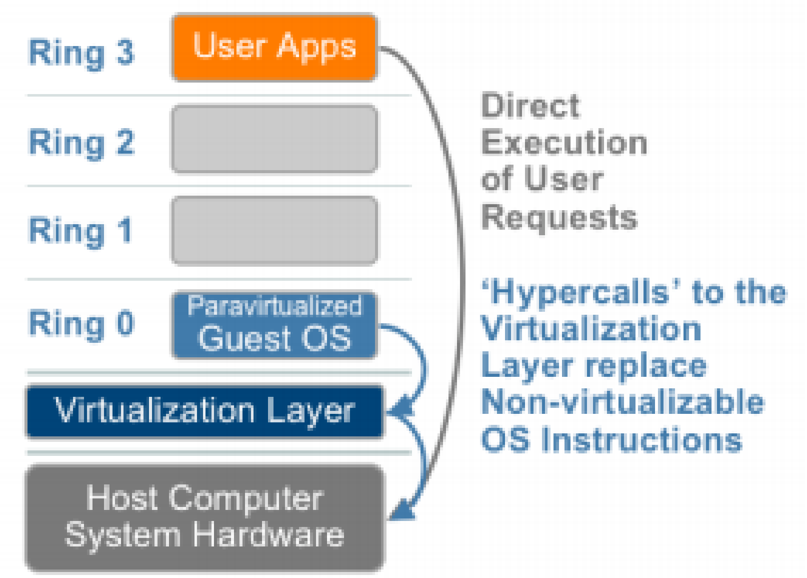
- Some instructions execute only in ring 0
- Instructions for low latency system calls (SYSENTER/SYSEXIT) always transition to ring 0 forcing the VMM into unwanted emulation or overhead
- Masking interrupts can only be done in ring 0
- Ring compression: paging does not distinguish privilege levels 0-2. Guest OS must run in ring 3, where it is not protected from its applications also running in ring 3
- Cannot be used for 64-bit guests on IA-32
- The OS can detect it is not running in ring 0


Ring 3
User Apps
Ring 2
Ring 1
Ring 0
OS
Direct Execution of User and OS Requests
Host Computer System Hardware

Virtualization – key concepts
Virtualization vs. Paravirtualization

Paravirtualization (OS-assisted virtualization)

- Modify the VM’s OS so it starts using the VMM’s services instead of direct hardware resources
- Demands explicit support from the hosted OS
- Not supported by all OS’s
- Isolation depends on “well mannered” hosted OS’s
- VMM and VM’s OS share Ring 0

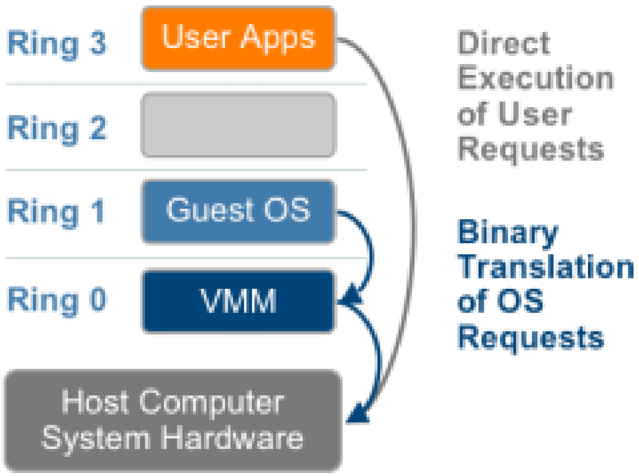


Virtualization – key concepts
Virtualization vs. Paravirtualization

Full virtualization (using binary translation, AKA “trap & emulate”**)**

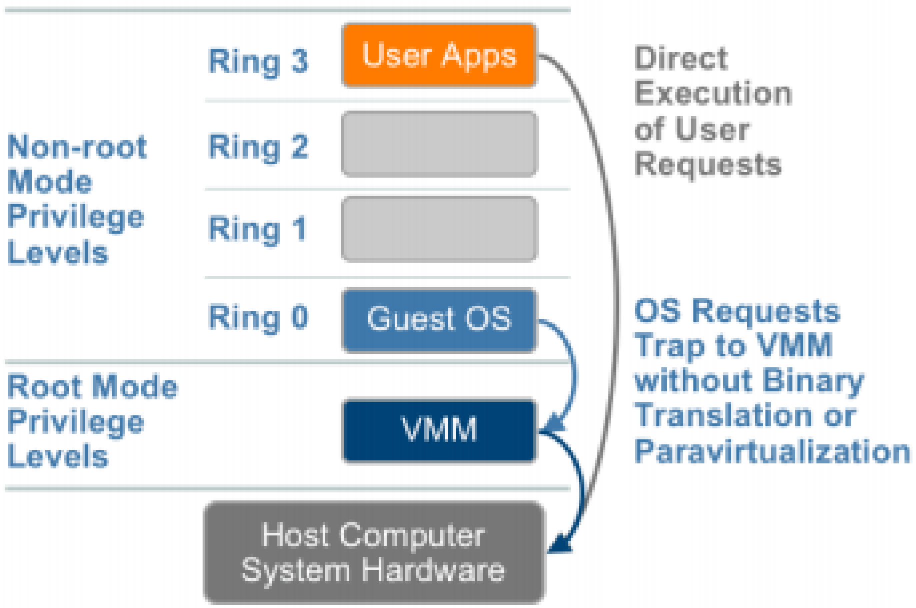
an OS intended for stand-alone use can successfully run inside a VM.

- Move Guest OS to “Ring 1” without recompiling
  - OS is still able to manage apps
  - VMM becomes able to control the guest OS (isolation)
- Problem: the guest OS may have “ring 0” instructions or instructions whose semantics change from “ring 0” to “ring 1”.
- Instead of recompiling, use binary translation for those instructions
- “Common” instructions may still run directly at the hardware (faster)
- No need for support from the guest OS


Ring 3
User Apps
Ring 2
Ring 1
Guest OS
Ring 0
VMM
Host Computer System Hardware
Direct Execution of User Requests
Binary Translation of OS Requests

Virtualization – key concepts
Hardware-assisted virtualization

- CPU directly supports trapping of sensitive instructions
- VMM runs in a new mode, bellow “Ring 0”


Ring 3
User Apps
Direct Execution of User Requests
Non-root Mode Privilege Levels
Ring 2
Ring 1
Ring 0
Guest OS
OS Requests Trap to VMM without Binary Translation or Paravirtualization
Root Mode Privilege Levels
VMM
Host Computer System Hardware

- Examples of virtualization-assistance technologies:
  - Intel VT (AMD-V (*AMD Virtualization*) *Intel Virtualization Technology*)

Virtualization – key concepts
No longer a black-and-white world...

Today most server-level virtualization platforms use hardware-assisted virtualization and support, in a hybrid manner, both full virtualization (whenever it is important not to modify the hosted system) and paravirtualization (for performance optimization).

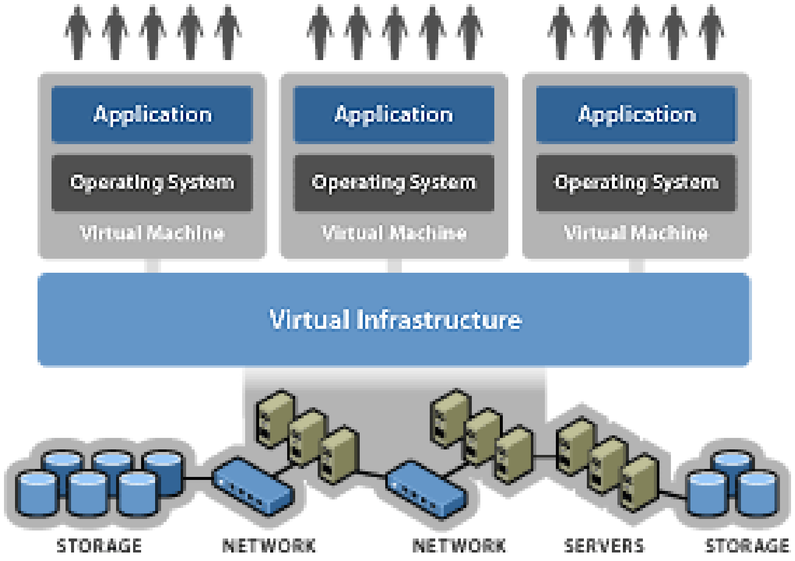
Interesting reading material for a deeper analysis:

https://www.vmware.com/techpapers/2007/understanding-full-virtualization-paravirtualizat-1008.html

(although this whitepaper is from 2007/2008, key concepts remain mostly unchanged)

Virtualization – key concepts
From Servers to Infrastructures

- The virtual infrastructure represents an additional virtualization step. It is composed of:
  - A set of physical servers, each with its own *hypervisor*.
  - Integrated management consoles for these *hypervisors*, including for instance monitoring (physical server, VMs), VM *snapshots*, VMs migration between servers, etc.
  - Automation tools for integrated O&M operations (e.g., disaster recovery).
  - Storage and network resources.



[Slightly off-topic:]
Virtual Appliances *server factory*

- VM + Operating System + software components installed for specific purposes (e.g., *web server*, *file server*).
  - e.g., *LAMP Appliances* (Linux + Apache + MySQL + PHP)
- *Virtual appliances* may be prebuilt by third parties. Examples:
  - https://marketplace.vmware.com/vsx
  - http://www.turnkeylinux.org
- *Appliances* may be used even by users without the technical skills required to correctly install and configure all the involved components – In alternative, a simplified (and often limited) configuration interface is provided.
  - e.g., no need to install the DBMS required to run that new app (just download and install a Virtual Appliance with everything preinstalled)

[Slightly off-topic:]
Virtual Appliances *server factory*

- The same concept may be used at corporate level: the *sysadmin* creates tailored appliances for later reuse:
  - A single installation can be performed with extra care, extensive optimizations and testing…
  - …for later usage multiple times…
  - …faster and with less effort…
  - …and with less risk of human errors.

- This type of appliances can be used at different levels:
  - OS
  - OS + Applications
  - SO + Applications + Configurations

Benefits from virtualization

Server Consolidation

- In the classic scenario each server supports a single application or service (or a relatively limited set of services). Hardware capacity is underused (typically around 5%-25%).

- Virtualization allows *appliance* consolidation *on physical servers*.

- Assuming a very conservative consolidation ratio of 1-to-10, for instance, a *datacenter* with 1000 servers uses 100 physical servers.
  - Less CAPEX costs
  - Less spending on the physical space (space rental)
  - Less spending on UPS and energy
  - Less spending on room cooling

Is 1-to-10 too optimistic? Think again

https://www.vmware.com/solutions/consolidation.html (apologies for commercial stuff)

[Slightly off-topic:] Virtual machine density (VM density)

- The number of VMs present in a single physical host that can be run normally without any of them being starved from any one resource
- Typical values greatly vary from scenario to scenario (depending on hosting hardware and VM workloads).
- Many authors point to 20-40 as common VM density values, but there are several cases where going over 100 is still OK:
  https://frankdenneman.nl/2016/02/15/insights-into-vm-density/
- Overprovisioning can further optimize these values:
  https://serverhobbyist.com/wp-content/uploads/2018/10/Dell-Best-Practices-for-Oversubscription-of-CPU-Memory-and-Storage-in-vSphere-Virtual-Environments_0.pdf

Benefits from virtualization

Enhanced availability:

- Each *appliance* is easily saved, including *snapshots* of its running state (including memory and *stack*, for instance), independently of the support for backups at the OS or application level.
- It becomes simpler to restore an *appliance* after hardware or software failures: just *rollback* to the latest image (“backup”) of that *appliance* (in the original server or in a new server, possible with different hardware features and different location).
  No need for *reboots* or software reinstalls.
- This process can be proactive, for instance for preventive hardware maintenance or planned upgrades.

(will future sysadmins still brag about long server uptimes?)

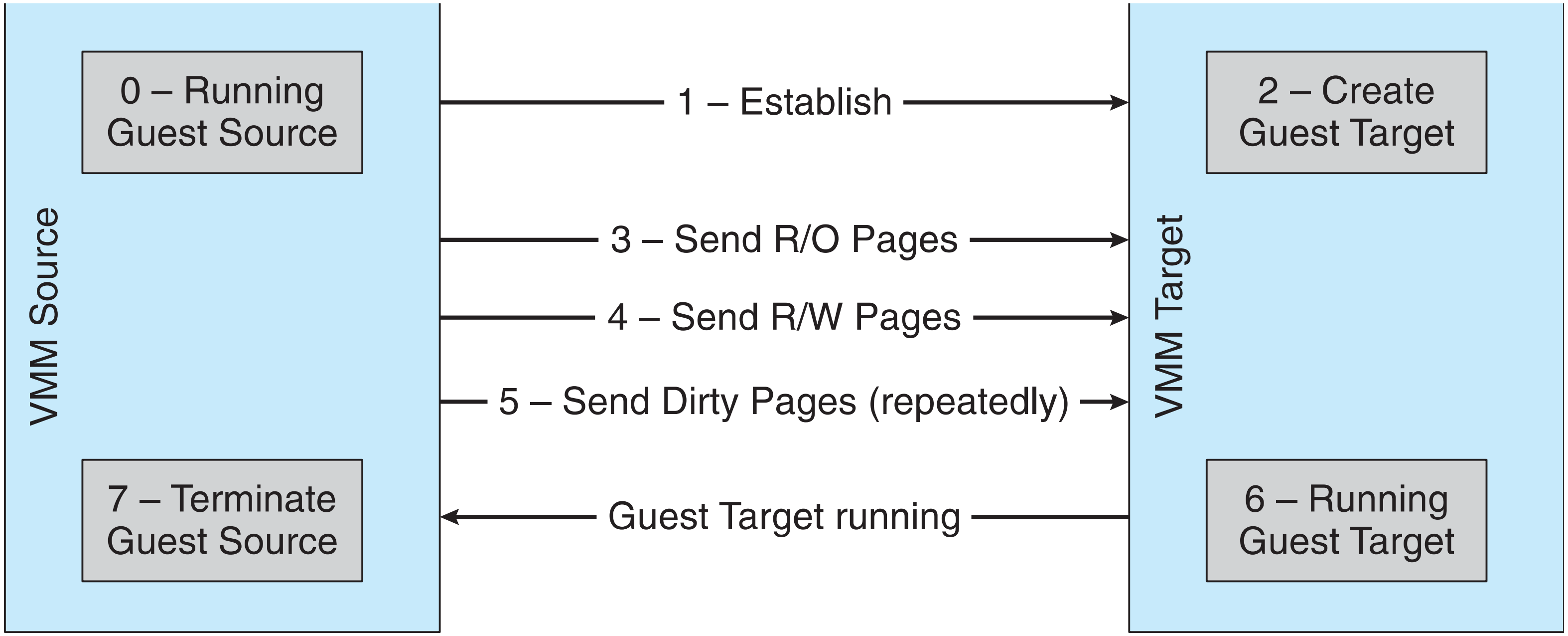
Benefits from virtualization

Live Migration:

- Running guest VM is moved between hosts without interrupting user access to the guest or its apps
- Source VMM establishes a connection with the target VMM
- Target VMM creates a new guest by creating a new vCPU, etc.
- Source VMM sends all read-only guest memory pages to the target
- Source sends all read-write pages to target, marked as clean
- Repeat previous steps (as during that step some pages were probably modified by the guest and are now dirty)
- When this cycle becomes very short, source VMM freezes the guest, sends VCPU's final state, sends other state details, sends final dirty pages, and tells the target to start running the guest
- Once the target acknowledges guest running, source terminates guest

How about Storage and Network?

Benefits from virtualization



Source: Operating System Concepts – 9th Edition, Silberschatz, Galvin and Gagne ©2013

Benefits from virtualization

Support for Legacy Systems:

- Several businesses still use legacy applications, conceived for obsolete hardware and/or operating systems which are no longer supported or commercialized.
- Virtualization allows porting such legacy applications to VM’s that can be installed on up-to-date *hardware*, thus decoupling the applications from the problems associated with the obsolete hardware (availability, performance, reliability, spares) they were built for.
- This allows extending the lifetime of legacy applications, with less maintenance costs, less risk of service failures and less performance restrictions.

Benefits from virtualization

Isolation of Services/Applications:

- It is no longer necessary to concentrate distinct services on the same server. Each appliance becomes specialized in a single service. This reduces the security and failure risks resulting from unforeseen interactions between distinct services.

- Testing new software becomes simple: just place it in isolated VMs and *rollback* to previous versions in case of problems.

Benefits from virtualization

Standardization:

- It becomes possible to tune applications/service deployments for reference platforms, independently of available hardware. This reduces costs and risks associated with the configuration process.

- Standardization may work at the company level (*sysadmin* defines a reference platform for all services) or by the provider of the application (which provides drivers and optimized configurations for a specific platform, instead of dispersing its effort across tenths of platforms.

How about portability and interoperability at VMM level?

Virtualization Tools

Main commercial platforms for server virtualization:

- VMware product suite
- Azure Virtual Machines
- Microsoft Hyper-V suite
- Citrix XenServer
- Oracle VM
- …

Main advantages of these platforms:

- O&M features (*virtual infrastructures*, image management, *disaster recovery*, dynamic resource management, etc.).
- Basic server virtualization functionalities already become a *commodity*, and even commercial products freely offer those functionalities.

[break-time - how about containers?]

After consolidating the physical server box...

Is the classic Virtual Machine looking too fat?

Do we really still need a complete OS for each VM?

Processes: *isolated address space, no isolation for files and networking, lightweight*

VM: *isolated address space, isolated files, and networking, heavyweight*

Containers: *isolated address space, isolated files and networking, lightweight*

[break-time - how about containers?]

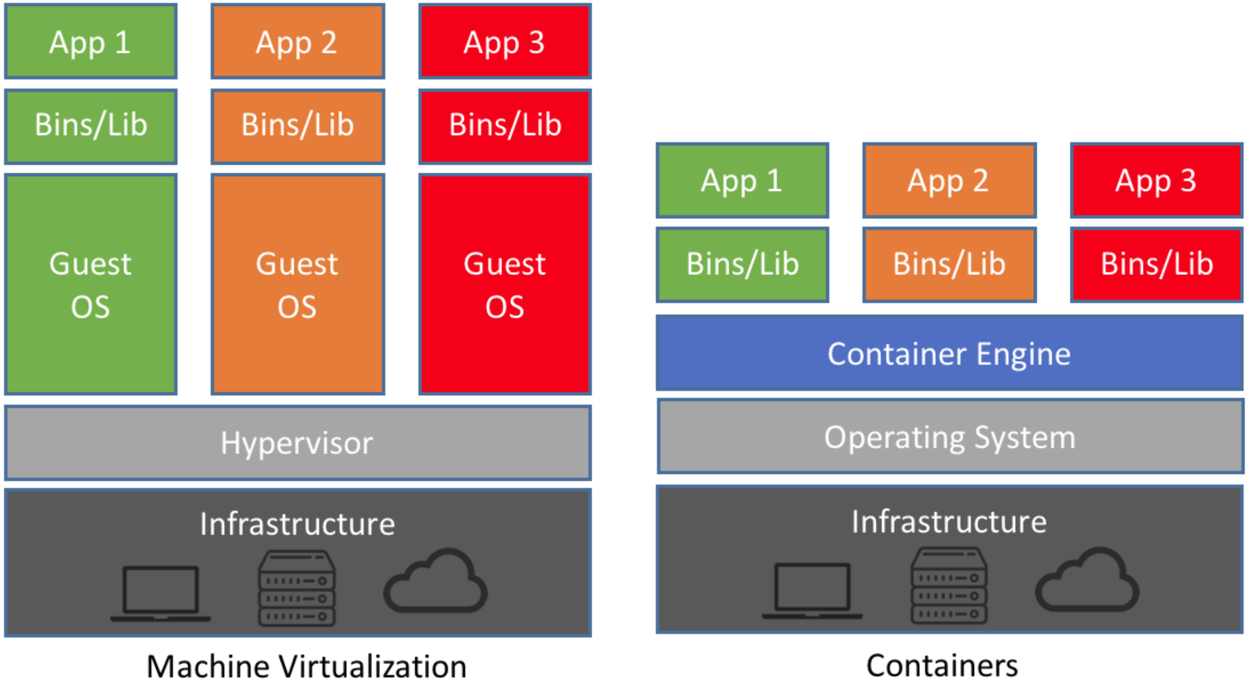


Image source: https://blog.netapp.com/blogs/containers-vs-vms/

Are we going from “fat servers” to “thin servers” to “thin services”?

https://www.slideshare.net/BodenRussell/realizing-linux-containerslxc

https://www.docker.com/

https://kubernetes.io/

Credits & Further Reading

Several slides were inspired by/include content from:

- Mariano Diaz, “Virtualization Concepts”, Virginia Tech CS5204
- The various links provided in the slides, especially:
  - https://www.vmware.com/techpapers/2007/understanding-full-virtualization-paravirtualizat-1008.html

Additional Reading

- Virtualization 2.0 for Dummies
  (freely available at VMware, registration required, also available at other links)
  (https://www.slideshare.net/emcacademics/virtualization-20-for-dummies)
- Commercial documentation (VMware, Microsoft, RedHat)
- On-line video sources:
  (beware: terminology is not always aligned with these slides!)
  - https://www.youtube.com/watch?v=NmrJlWDG0yk