Chapter 3
Virtualization

Mastering Cloud Computing
Paul Talaga

UNIVERSITY OF Cincinnati

Virtualization

Typically synonymous with *hardware virtualization* and *IaaS.*

Causes for current interest:

- Increased computing power
- Underutilized hardware
- Lack of space
- Greening initiatives
- Rise of administration costs

UNIVERSITY OF Cincinnati

Characteristics of Virtualization

To create a virtual version of something:

- software environment (hardware vm)
- storage
- network (software defined net or VPN)

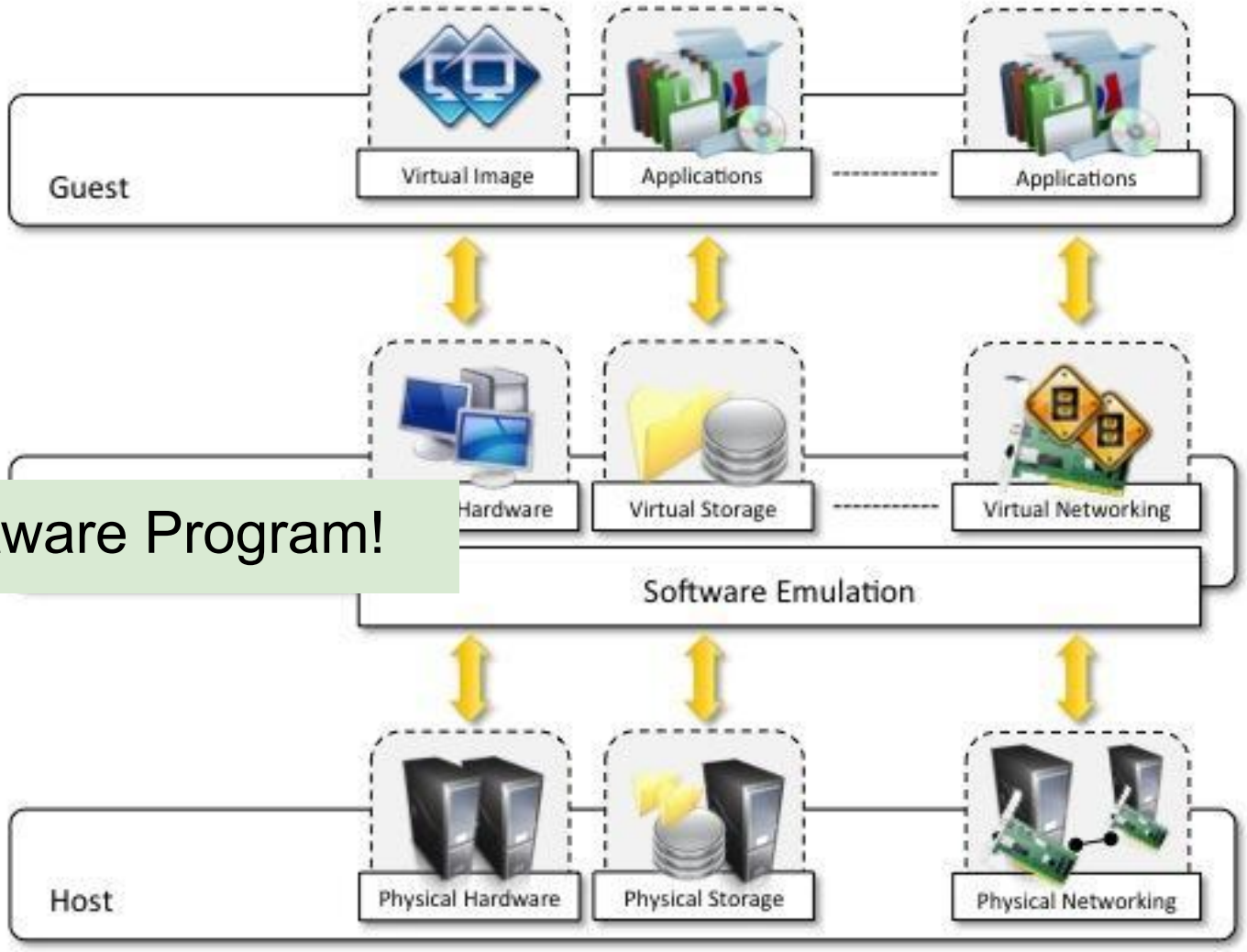
3 Components:

- Guest
- Host
- Virtualization layer

Started with IBM CP/CMS in early 70s

UNIVERSITY OF Cincinnati

A Software Program!



Increased Security

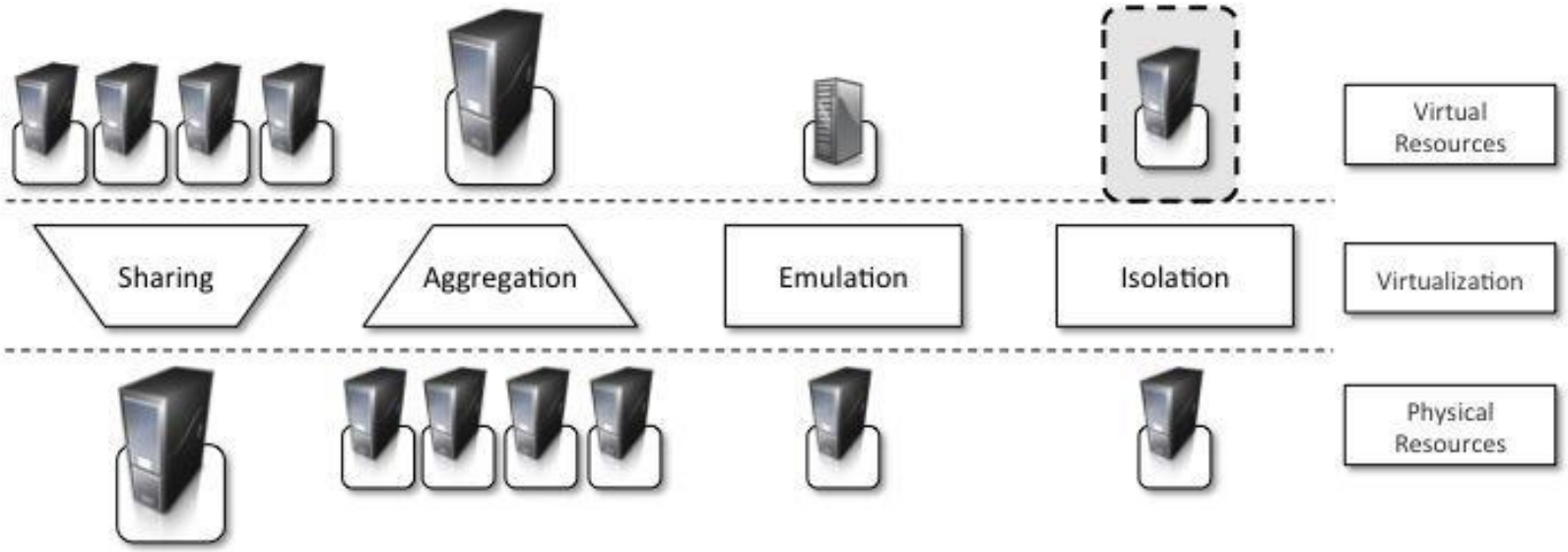
- VM manager can *control* and *filter* activity of the guest.
- Hide information on host
- Sandbox environment (JVM & .Net)
- Adjacent virtualized hosts can't spy on each other (ref, ref)

UNIVERSITY OF Cincinnati

Managed Execution

Additional features:

- Sharing - better utilization
- Aggregation - many looks like 1
- Emulation - provide different hardware to host
- Isolation - security



Other features

- Performance tuning - tune host for optimal performance - expose custom hardware to guest
- VM snapshots - pausing - saving - resuming
- VM migration - move a vm from one host to another, sometimes *while running*
- Portability - move VM from host to host

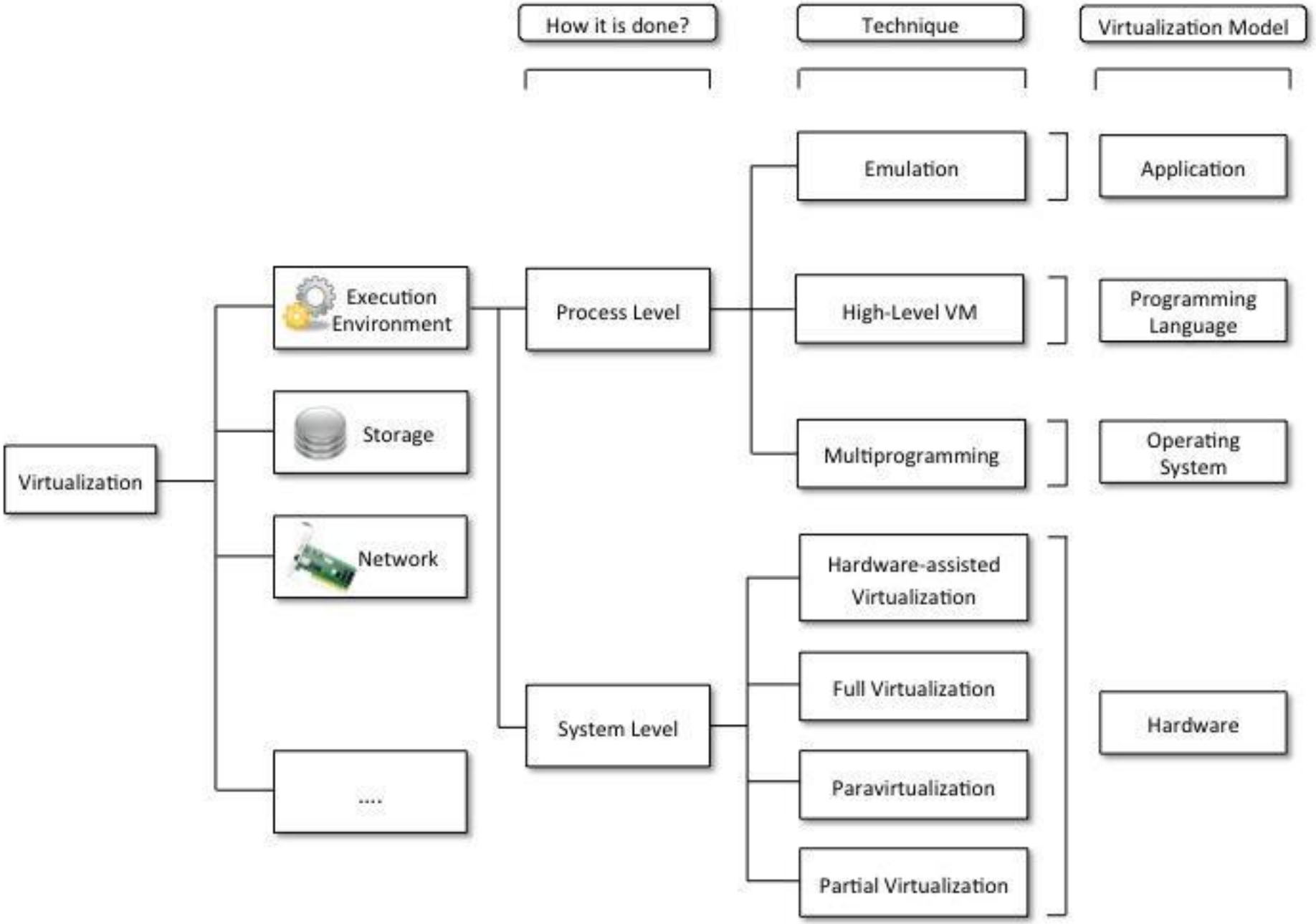
UNIVERSITY OF Cincinnati

Virtualization Taxonomy Execution Virtualization

2 Main categories:

- Process-level - on top of an existing OS
- System-level - Directly on hardware or minimal OS support

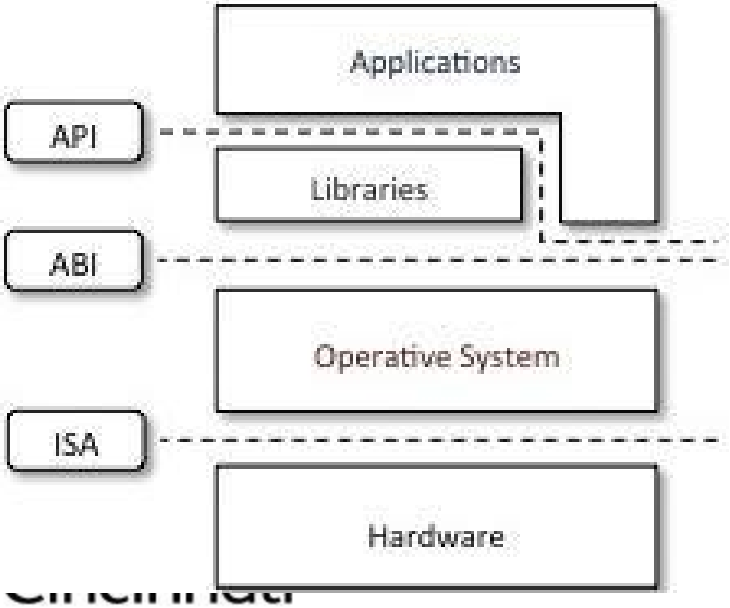
UNIVERSITY OF Cincinnati

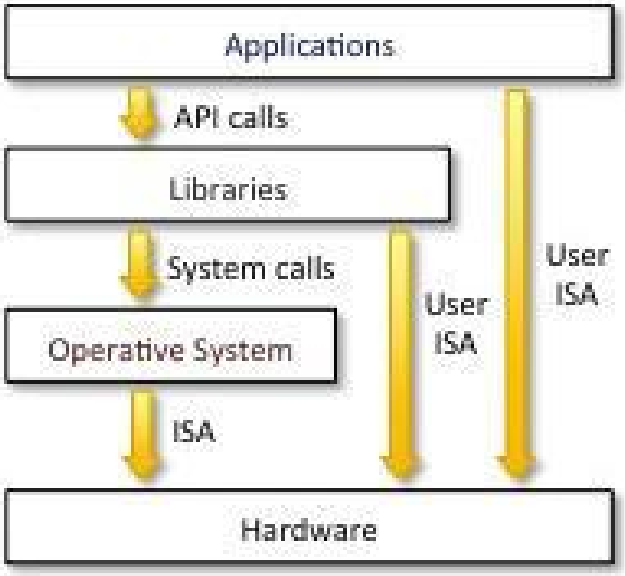

How it is done?
Technique
Virtualization Model
Virtualization
Execution Environment
Storage
Network
....
Process Level
System Level
Emulation
High-Level VM
Multiprogramming
Hardware-assisted Virtualization
Full Virtualization
Paravirtualization
Partial Virtualization
Application
Programming Language
Operating System
Hardware

Machine Reference Model

Defines layer of abstraction

- ISA - instruction set architecture, registers, memory, interrupts
- ABI - application binary interface, data-types, alignment, system-calls
- API - application programming interface, libraries and underlying OS





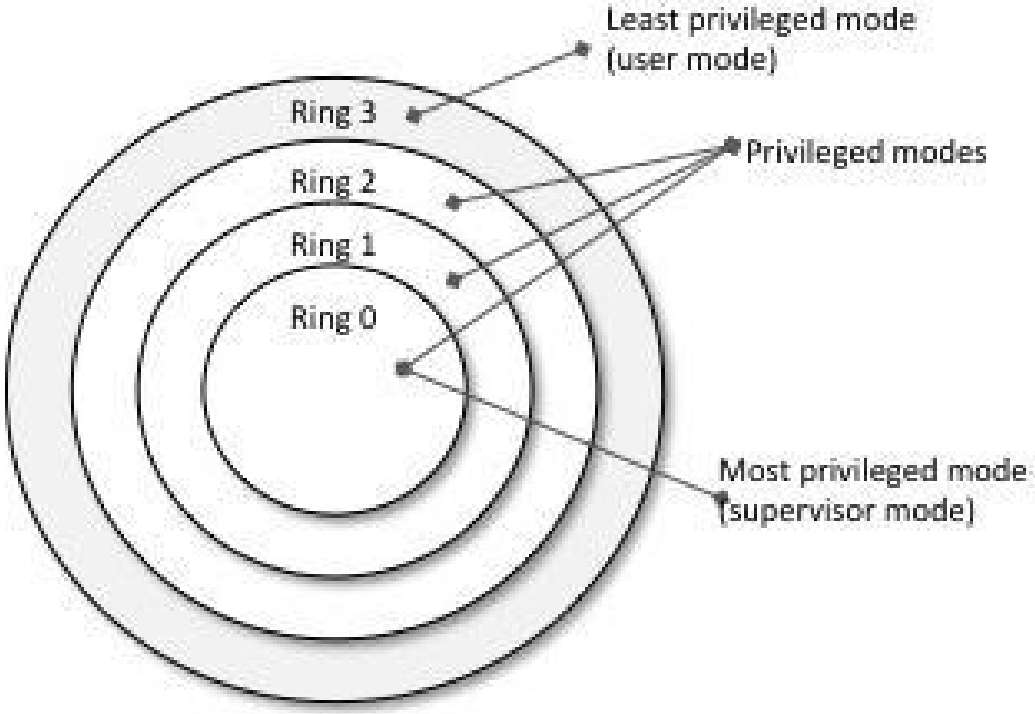
Layered Approach helps with Security

Hardware can allow different layers to execute different instructions.

- Privileged - change shared resources - IO and register changing instructions
- Nonprivileged - safe - don't access shared resources - math instructions

UNIVERSITY OF Cincinnati

Ring for Hierarchy of Privileges



Most recent systems only support 2 levels: Ring 0 for supervisor mode, and Ring 3 for user mode.

UNIVERSITY OF Cincinnati

Current Systems & Hypervisor

Use only 2 levels:

- Ring 0 - supervisor mode (kernel)
- Ring 3 - user mode
  - Sensitive instructions cause trap to kernel

Virtualization adds a *hyper*visor over Ring 0 - in reality they run at same level

BUT - how to isolate different OSs if they all need access to privileged instructions?

UNIVERSITY OF Cincinnati

Another VM Issue

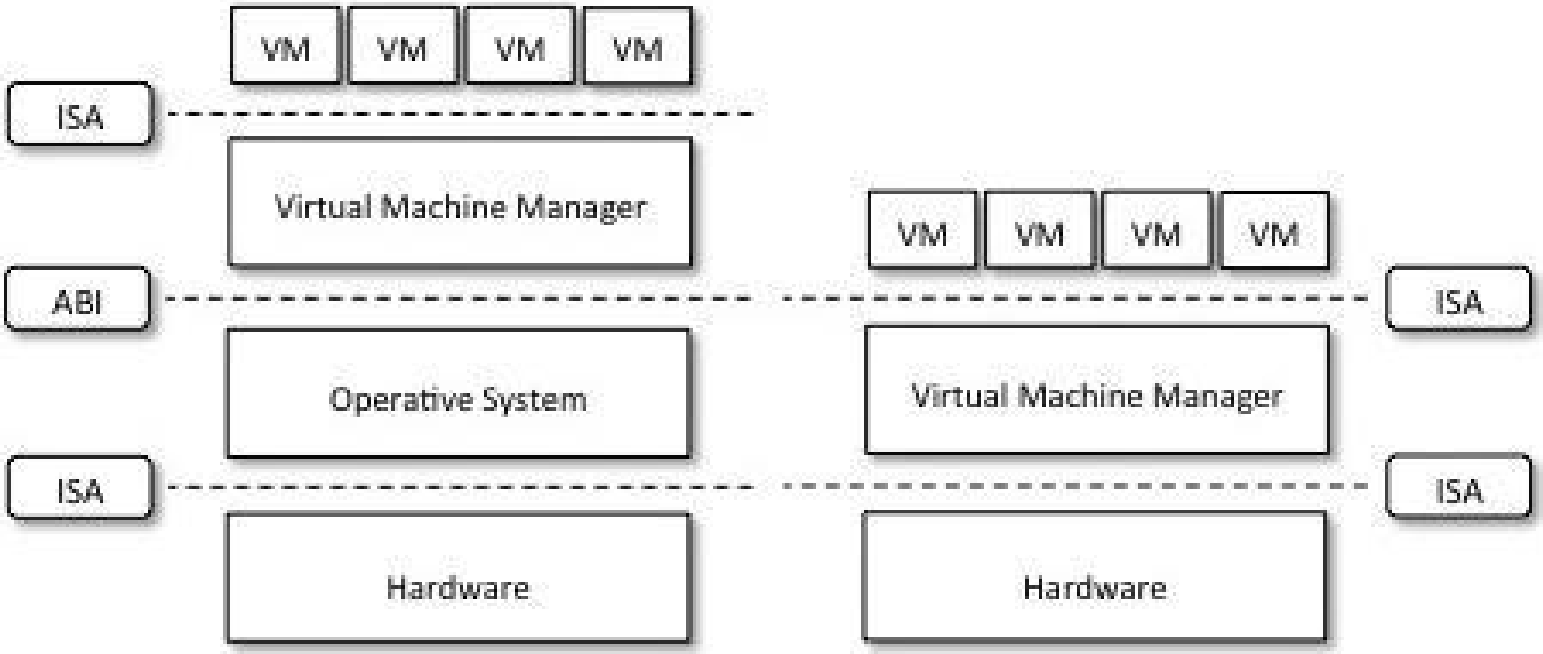
Original ISA had 17 sensitive instructions in user mode - can't cause a trap!

Intel VT & AMD Pacifica moved these to privileged mode.

UNIVERSITY OF Cincinnati

Hardware-level Virtualization

- Virtualizes ISA - *system virtualization*
- Hypervisors manage system - virtual machine manager (VMM)
  - Type I - Runs directly on hardware - *native virtual machine*
  - Type II - Runs in an OS - *hosted virtual machine*



Type Details

- Type I - Runs directly on hardware - *native virtual machine*
  - More resource efficient (no OS in the way)
  - Must reinstall ‘OS’
  - ESX/ESXi just mini version of Linux
  - Ex: VMWare ESX/ESXi, MS HyperV
- Type II - Runs in an OS - *hosted virtual machine*
  - Easier to use, just install the program
  - Ex: VMWare Workstation/Fusion, VirtualBox, Kernel-based Virtual Machine (KVM)

Types are not definitive! KVM uses virtualization features in the kernel, but can do general purpose work. ESX(i?) is linux as well and you can run normal programs.

UNIVERSITY OF Cincinnati

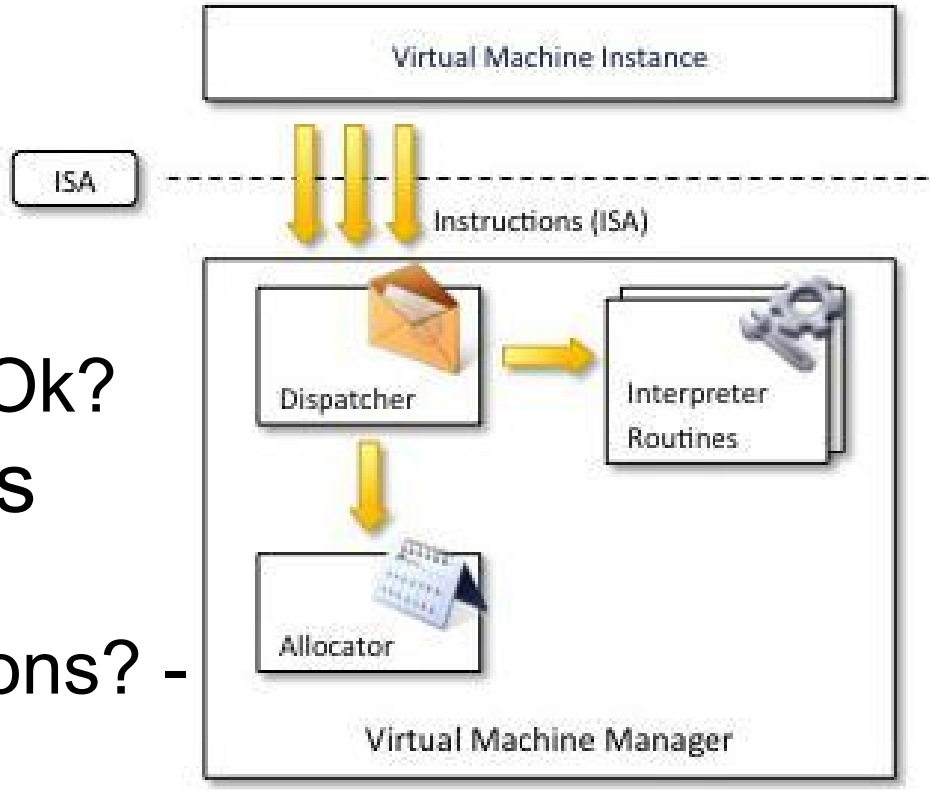
Type II

Type I

VMM in Detail

3 parts:

- Dispatcher (route instructions)
- Allocator
  - system resource Ok?
- Interpreter routines
  - What to do with sensitive instructions? - run special code!



UNIVERSITY OF Cincinnati

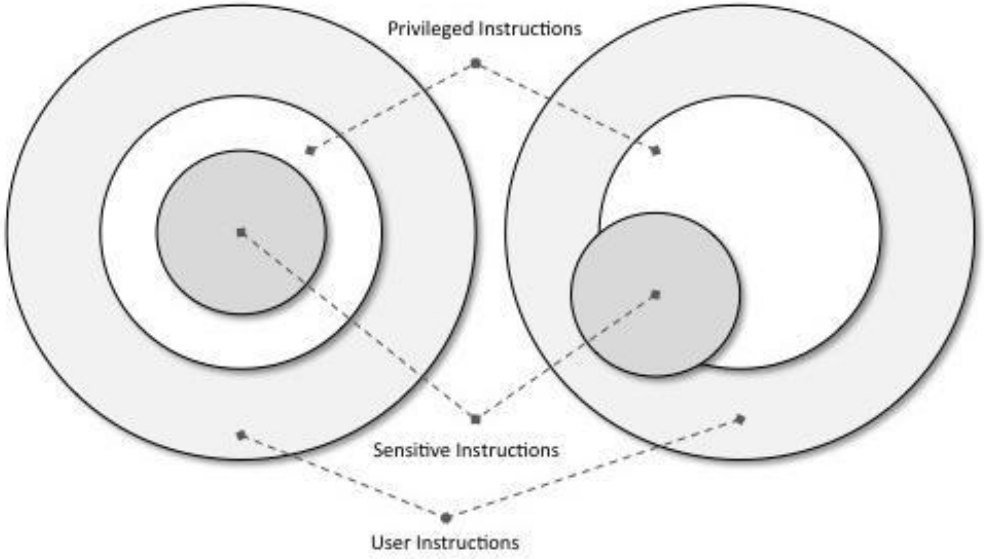
VMM Requirements - by Goldberg and Popek (1974)

- *Equivalence* - A guest should behave exactly as it did on physical hardware
- *Resource control* - VMM should be in complete control
- *Efficiency* - Significant amount of guest instructions untouched by VMM

UNIVERSITY OF Cincinnati

Goldberg and Popek's 3 Theorems for a VMM

1. For any conventional third-generation computer (IC), a VMM may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions.



UNIVERSITY OF Cincinna

Goldberg and Popek's 3 Theorems for a VMM

2. A conventional third-generation computer is recursively virtualizable if:

1. It is virtualizable
2. A VMM without any timing dependencies can be constructed for it.

A VM in a VM (in a VM)? YES! But will be slow.

UNIVERSITY OF Cincinnati

Goldberg and Popek's 3 Theorems for a VMM

3. A hybrid VMM may be constructed for any conventional third-generation machine in which the set of user-sensitive instructions is a subset of the set of privileged instructions.

A hybrid virtual machine (HVM) traps user-level instructions as well.

UNIVERSITY OF Cincinnati

Hardware Virtualization Techniques

- Hardware Assisted
- Full virtualization
- Paravirtualization
- Partial virtualization

UNIVERSITY OF Cincinnati

Hardware Assisted

- Hardware provides features for virtualization - Intel VT & AMD V
- Adds extensions to x86-64
- Originally used in IBM System/370
- Examples: Kernel-based VM (KVM), VirtualBox, Xen, VMware, Hyper-V, Sun xVM, Parallels

UNIVERSITY OF Cincinnati

Full Virtualization

- Ability to run guest with no modifications.
- Guest has no idea it is virtualized.
- All privileged instructions need to be trapped
- Required for Windows OSs when HW not available.
- Works on all hardware, but SLOW

UNIVERSITY OF Cincinnati

Paravirtualization

- Non-transparent virtualization
- Guest must be changed - change sensitive instructions to VMM API calls
- Simplier VMM
- Much faster:
  - No traps! Code runs on bare hardware
  - OS/driver optimizations for greater speed
- Installing guest drivers makes full virtualization into para-ish
- Examples: Xen, IBM VM OS, VMware, Parallels

Partial Virtualization

- Not all features of hardware/OS are virtualized
- Examples:
  - Address space virtualization - virtual mem
  - RAID - disk
  - Network bonding
  - VPNs
    - Same hardware otherwise

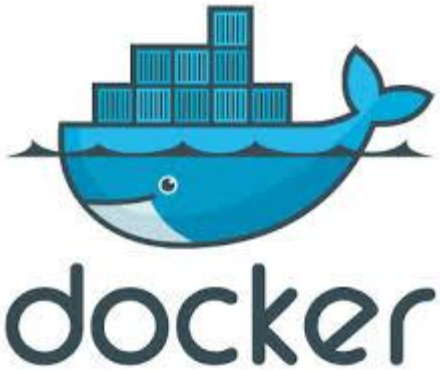
Practical Issues

- Hardware & Full Virtualization
  - Can overprovision CPU
  - Dedicated Ram
  - Dedicated disk (sometimes, delta drives possible)
- Paravirtualization
  - Can overprovision CPU **& RAM!**
  - Can set RAM ranges and let OSs self manage!

OS-level Virtualization

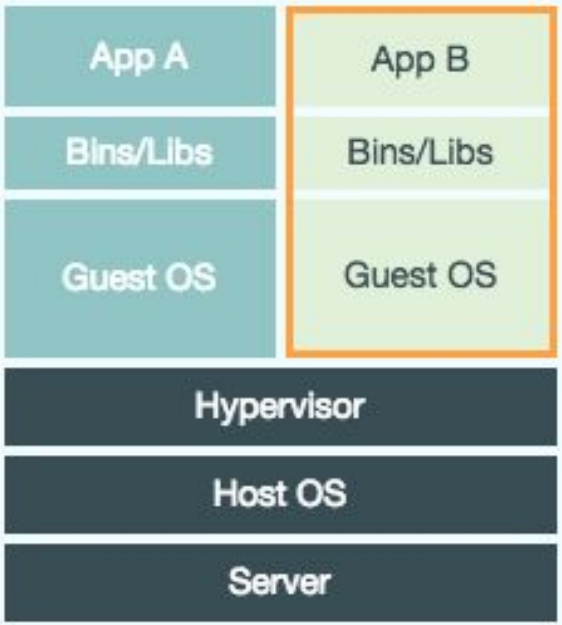
- Separate user-spaces
- No hypervisor needed
- Different filesystems, network configurations, software, access to devices
- Extension of the *chroot* unix concept
- **Little to no overhead**
- Can use any hardware, any software
- But, must all use same OS
- Ex: FreeBSD Jails, Parallels Virtuozzo Containers, OpenVZ, EC2? others..

UNIVERSITY OF Cincinnati



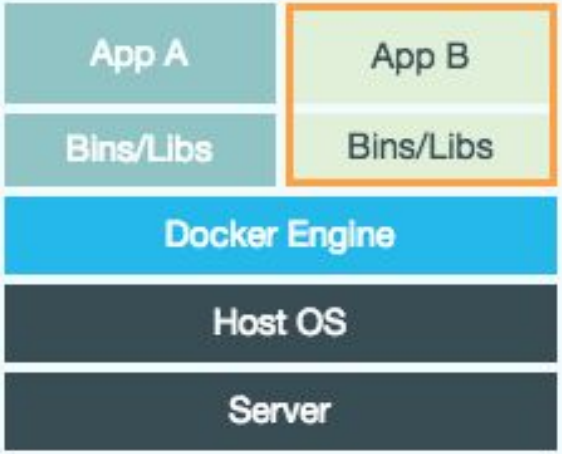
- Open Sourced Mar 2013 - Jan ‘14 capital
- Extension of LXC (Linux Containers)
- Uses Images to build other Images (Containers)
- Git feel
- Simple config file to generate images
- Lightweight & processes run on bare hardware (fast!)

UNIVERSITY OF Cincinnati



Virtual Machines

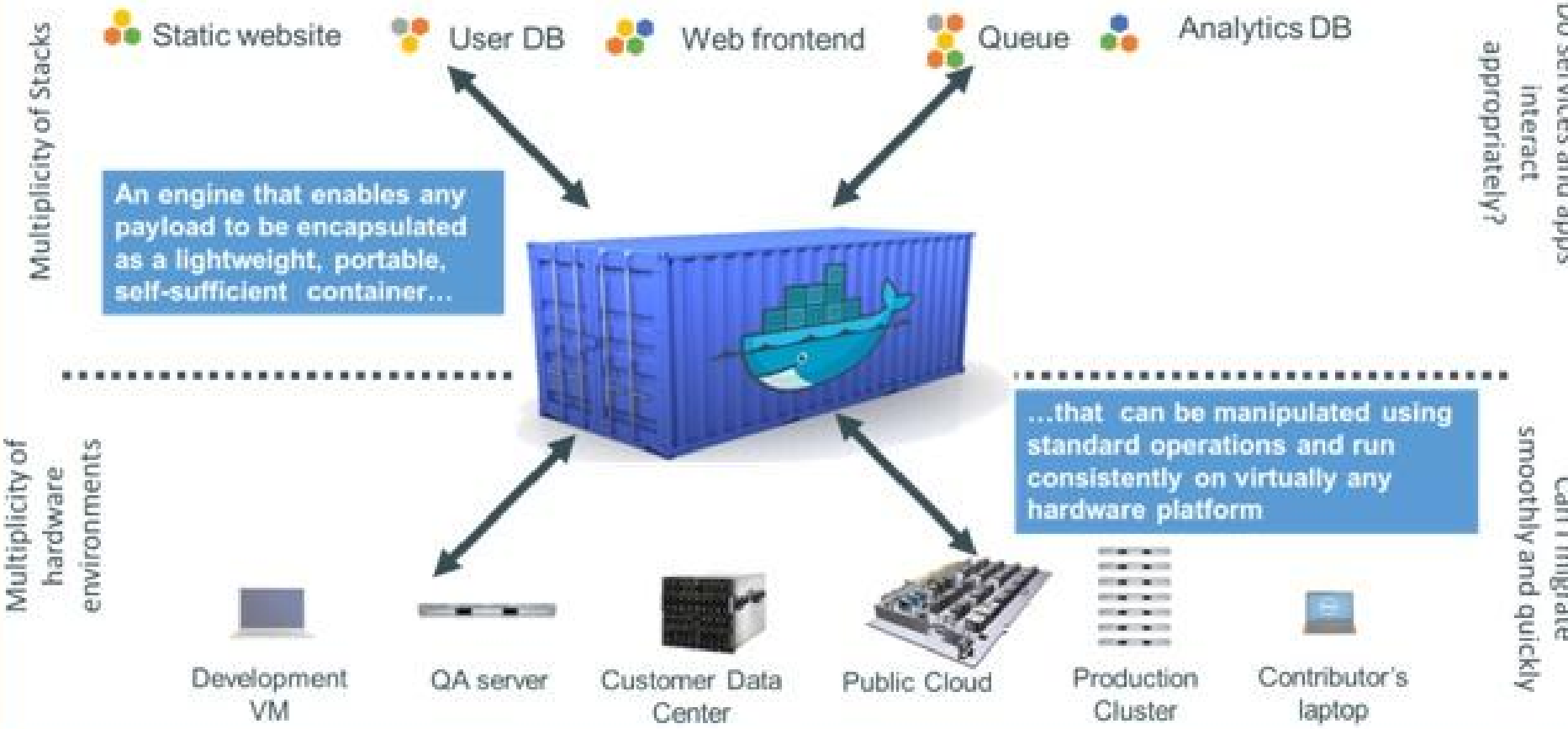
Each virtualized application includes not only the application - which may be only 10s of MB - and the necessary binaries and libraries, but also an entire guest operating system - which may weigh 10s of GB.

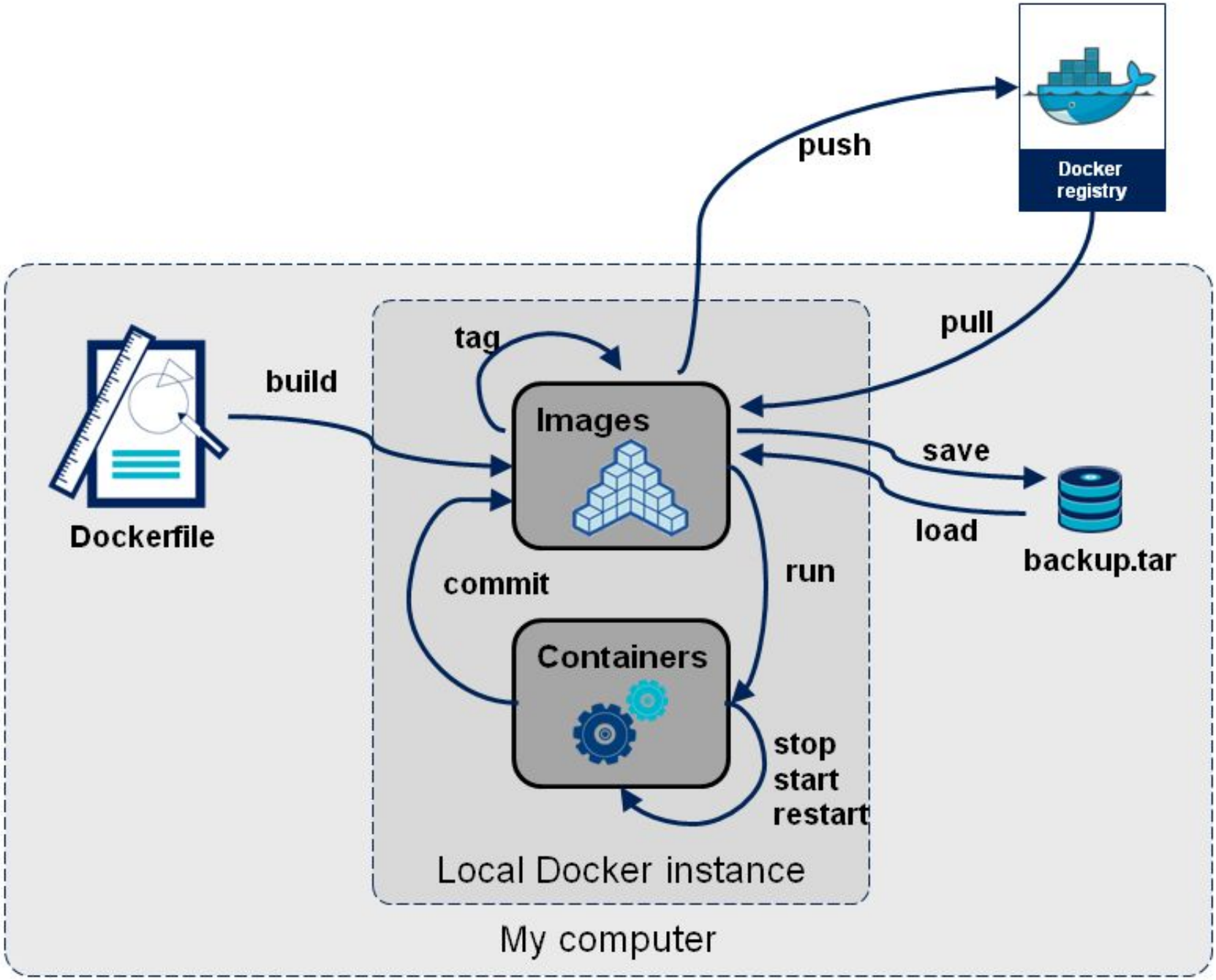


Docker

The Docker Engine container comprises just the application and its dependencies. It runs as an isolated process in userspace on the host operating system, sharing the kernel with other containers. Thus, it enjoys the resource isolation and allocation benefits of VMs but is much more portable and efficient.

Docker is a shipping container system for code




Docker registry
push
pull
tag
build
Images
save
load
backup.tar
Dockerfile
commit
run
Containers
stop
start
restart
Local Docker instance
My computer

Docker Commands

https://www.docker.com/tryit/

https://docs.docker.com/reference/commandline/cli/

a. `docker search centos`
b. `docker pull centos  (or any other image)`
c. `docker images`
d. `docker run -it <image name>`
e. `docker run -v </mymachine>:</vm> <image name> <cmd>`
f. `docker run -d -p 8080:80 <image name> <cmd>`
g. `docker ps (-a)`
h. `docker commit <image id> <repo name>`
i. `docker build <folder with Docker file>` (link)
j. `docker save <container id> > out.tar`
k. `docker import out.tar`
l. `docker save <container id> | gzip > out.tar.gz`
m. `docker rm <container id>`

UNIVERSITY OF Cincinnati

Allow regular users to use Docker on a system:

```
usermod -a -G docker <username>
```

Programming Language-Level Virtualization

- Allows programs to be run on different machines with different OS/architectures.
- Uses virtual machine to ‘run’ code: slow.
- Java (JVM), .Net (CLI), Python, Pascal, Groovy, Ruby
- Popular with enterprise apps
- Can be interpreted or *jitted*
- Stack-based (Java, CLI) or register-based
- Security a big win: easy to sandbox

Application-level Virtualization

- Allows applications to run in an environment it shouldn't
- Extra layer emulates missing pieces: filesystem, libraries, OS
- *2 methods:*
  - *Interpretation:* every ISA instruction is handled - minimal startup, huge overhead
  - *Binary translation:* Instructions translated to native, then cached and reused.
- Ex: WINE, CrossOver (Mac), VMware ThinApp packages an exe

UNIVERSITY OF Cincinnati

Storage Virtualization

- Physical location of data not important
- Accessed through logical path or similar
- Management of write access important
- Ex:
  - SAN - Virtualize SATA/SCSI cable - single computer at a time connected
  - NAS - Expose virtual filesystem over a network (NFS, AFP, SMB)
  - Dropbox.com, Box.com, SkyDrive

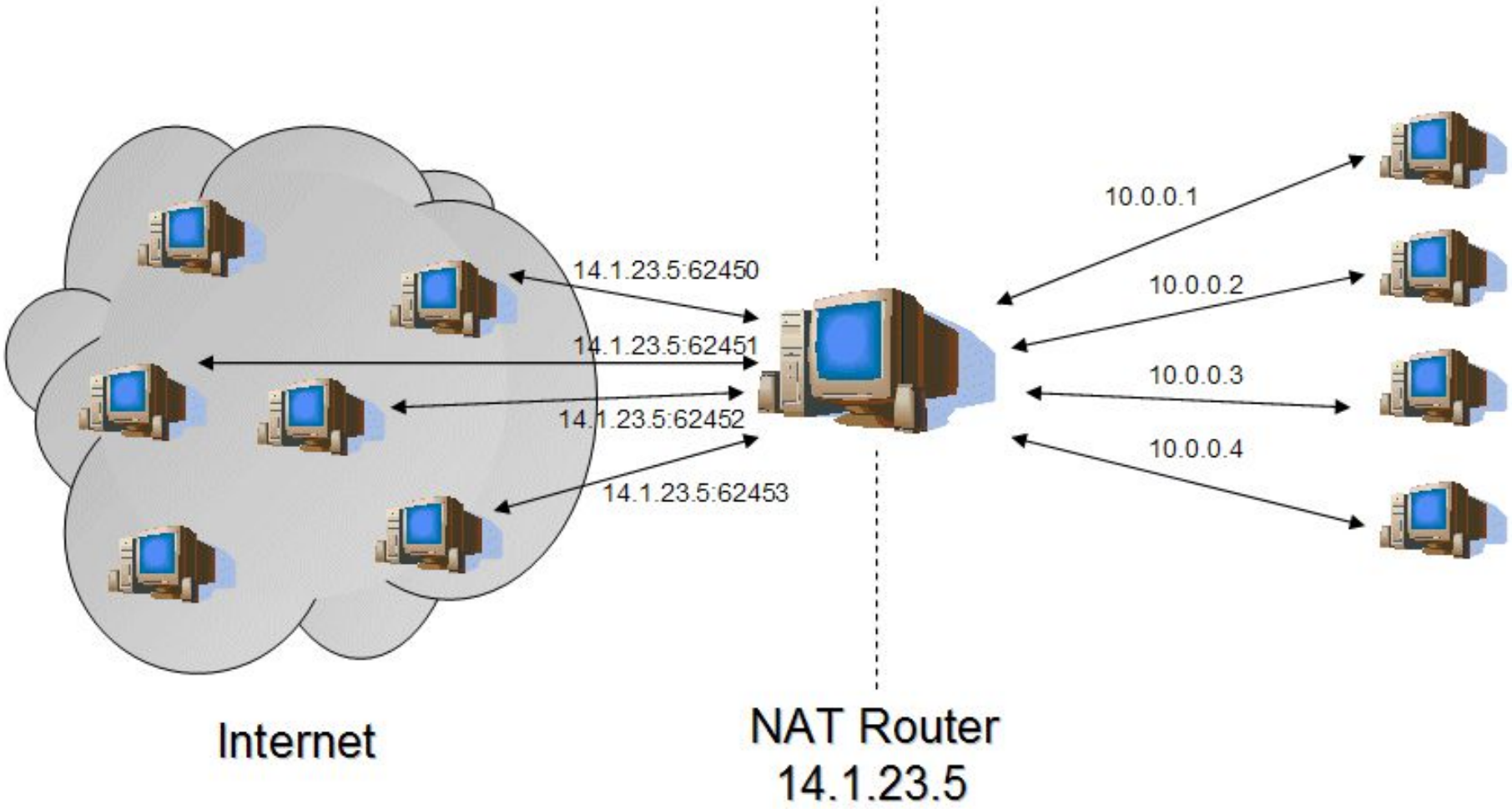
UNIVERSITY OF Cincinnati

Network Virtualization

- *External* net virtualization - VLAN - Separate layer 2 networks (broadcast domain) on same physical wire
- *Internal* net virtualization - simulated network in computer, as in to connect multiple VMs on in one host.
  - NAT (Network Address Translation)

UNIVERSITY OF Cincinnati

NAT Diagram



UNIVERSITY OF Cincinnati

Desktop Virtualization

- Allows a desktop environment remotely
- Pros: high availability, persistence, accessibility, ease of management
- Neg: lower refresh, lag
- Old idea… from mainfraime era
- Implemented with IaaS or in single OS
- Ex: RDP (Windows Remote Services), VNC, X Server.

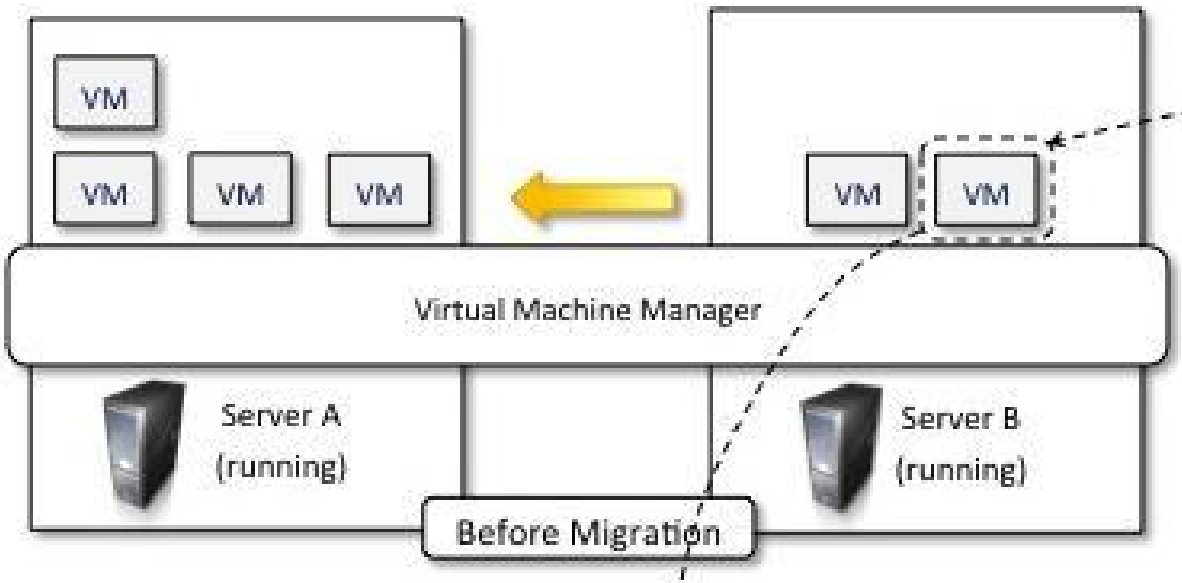
UNIVERSITY OF Cincinnati

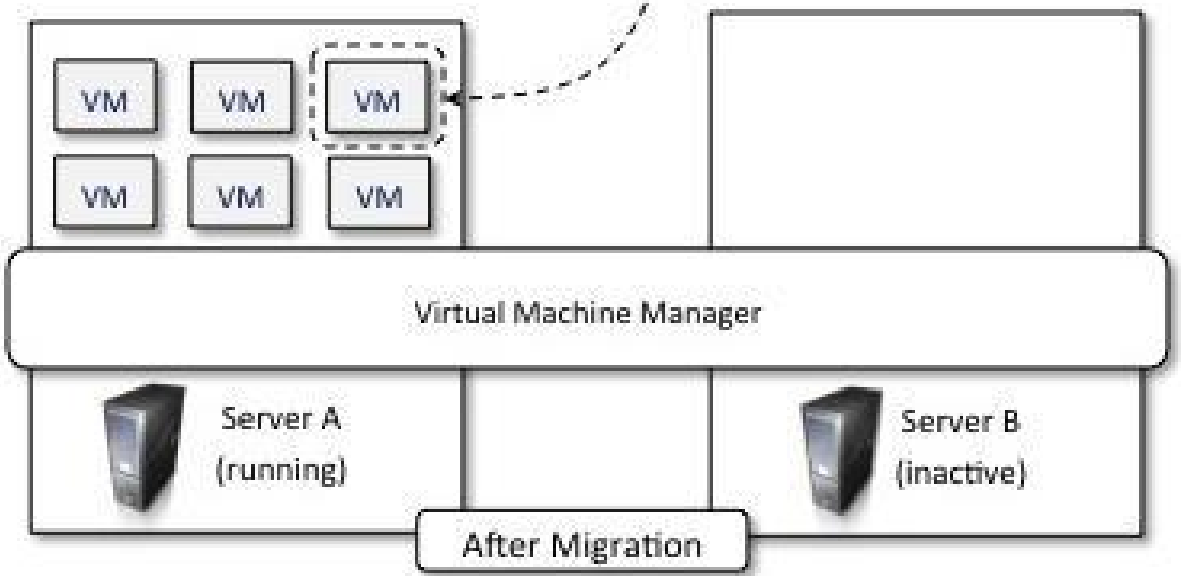
Virtualization and Cloud Computing

- Facilitates IaaS
- Facilitates PaaS
- Customization, security, isolation, manageability
- Allows better use of abstraction
- Opens new market-based computing opportunities

UNIVERSITY OF Cincinnati

VM Migration





Virtual Machine migration?

Virtual Network migration?

Virtualization Pros/Cons

- Pros:
  - Managed execution
  - Isolation
  - Portability / Migration
  - Efficiency
- Cons:
  - Efficiency - hypervisor or degraded environment
  - Security
    - Phishing
    - Malicious hypervisor: BluPill, SubVirt (rootkits)
    - Replace VM with compromised one (JVM)

UNIVERSITY OF Cincinnati

Technology Examples - FYI Not Tested - Details in Text

- Xen: paravirtualization
  - XenSource - Citrix (commercial)
  - Xen Cloud Platform - XCP - open source
- VMware: Full virtualization
  - Desktop
    - VMware Workstation
    - VMware Fusion (Mac)
  - Server
    - VMware GSX
    - VMware ESXi
    - VMware vSphere - cloud management solution

UNIVERSITY OF Cincinnati

Technology Examples - FYI Not Tested - Details in Text

- Microsoft Hyper-V
  - Part of Windows Server 2008 R2 onward
  - Windows Server Core (2008) Optimized for Hyper-V (removed extra MS components)
  - System Center Virtual Machine Manager (SCVMM) - manage VM's on multiple servers - can interoperate with VMware

UNIVERSITY OF Cincinnati

Virtualization Summary

- Ability to provide illusion of an environment
  - Runtime:
    - Hardware: IaaS - VMware, VirtualBox
    - OS Level: CHROOT, Docker
    - Programing language: JVM, Python
    - Application: WINE
  - Storage: SAN, NAS, Dropbox.com
  - Network: VLAN, Virtual Networking, NAT, VPN
  - Remote Desktop: RDP, X Server
  - Pros/cons

UNIVERSITY OF Cincinnati