COMPUTING AS A SERVICE (IaaS)

PART I

COMPUTING AS A SERVICE (IAAS)

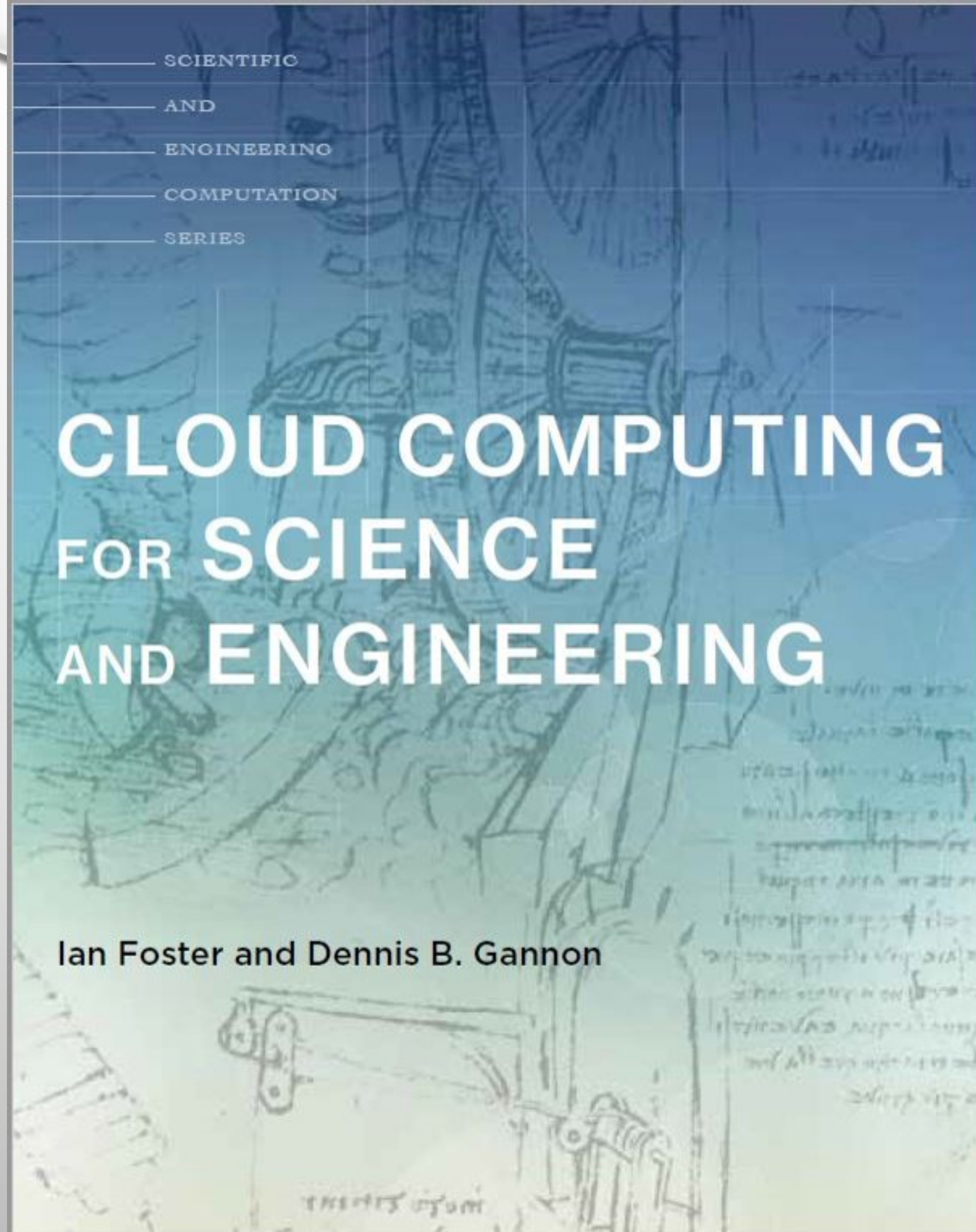
- THE TWO SIMPLEST FORMS OF CLOUD COMPUTING ARE PERHAPS THE BEST KNOWN TO SCIENTISTS AND ENGINEERS:

STORAGE AND COMPUTING ON DEMAND

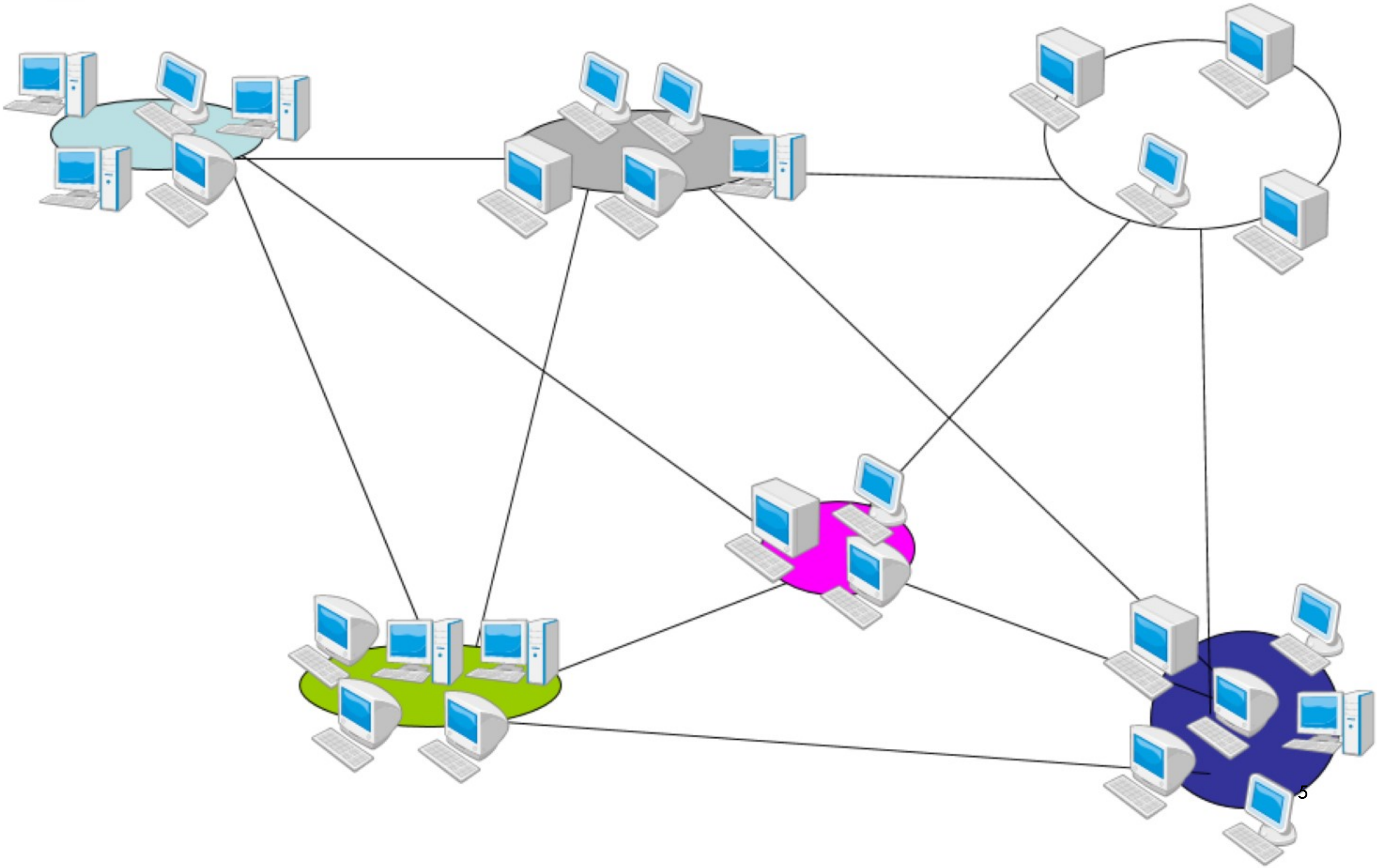
- STORAGE WE WILL DO NEXT
- LET'S START WITH COMPUTING AS A SERVICE

Virtualization Techniques for Cloud Computing

Prof. Chih-Hung Wu
Dept. of Electrical Engineering
National University of Kaohsiung
Email: johnw@nuk.edu.tw
URL: http://www.johnw.idv.tw


SCIENTIFIC
AND
ENGINEERING
COMPUTATION
SERIES
CLOUD COMPUTING
FOR SCIENCE
AND ENGINEERING
Ian Foster and Dennis B. Gannon

In the computer-age…



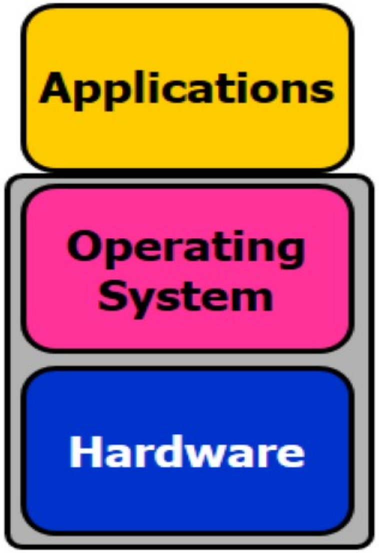
A Lot of Servers/Machines...

- Web server
- Mail server
- Database server
- File server
- Proxy server
- Application server
- …and many others

A Lot of Servers/Machines...

- The data-centre is FULL
  - Full of <u>under utilized</u> servers
  - Complicate in management
- Power consumption
  - Greater wattage per unit area than ever
  - Electricity overloaded
  - Cooling at capacity
- Environmental problem
  - Green IT

The Use of Computers



Traditional Server

The Traditional Server Concept

- Unless there are multiple servers, if a service experiences a hardware failure, then the service is down.
- System Admins can implement clusters of servers to make them more fault tolerant.
- However, even clusters have limits on their scalability, and not all applications work in a clustered environment.

The Traditional Server Concept

- Pros
  - Easy to conceptualize
  - Fairly easy to deploy
  - Easy to backup
  - Virtually any application/service can be run from this type of setup
- Cons
  - Expensive to acquire and maintain hardware
  - Not very scalable
  - Difficult to replicate
  - Redundancy is difficult to implement
  - Vulnerable to hardware outages
  - In many cases, processor is under-utilized

Support for Cloud…

- You need 100 CPUs now, rather than when your job reaches the head of the queue in your lab cluster.

- You need to access a GPU cluster or a computer with 1 TB memory, but such a resource does not exist in the lab.

- You need computers loaded with 10 different Linux variants to test the portability of your new software prior to distribution

- You want a machine outside your firewall that you can share with your collaborators for a project

Virtualization

- Public cloud data centres comprise many thousands of individual servers.
- Some servers are used exclusively for data services and supporting infrastructure and others for hosting your computations.
- When you compute in the cloud, you do not run directly on one of these servers in the way that you would in a conventional computational cluster.
- Instead, you are provided with a virtual machine running your favourite operating system.
- A virtual machine is just the software image of a complete machine that can be loaded onto the server and run like any other program.
- The server in the data centre runs a piece of software called a hypervisor that allocates and manages the server's resources that are granted to its "guest" virtual machines.
- When you run in a VM, it looks exactly like a server running whatever operating system the VM is configured to run

Virtualization

- In the late **1960s and early 1970s** , IBM and others created many variations on virtualization and eventually demonstrated that they could virtualize an entire computer.
- What resulted was the concept of a hypervisor : a program that manages the virtualization of the hardware on behalf of multiple distinct OSs.
- Each such OS instance runs on its own complete VM that the hypervisor ensures is completely isolated from all other instances running on the same computer.
- The OS allows multiple user processes to run simultaneously by sharing the resources among them; the hypervisor below the OS allows multiple OSs to share the real physical hardware and run concurrently.
- Many hypervisors are available today, such as Citrix Xen, Microsoft Hyper-V, and VMWare ESXi.
- We refer to the guest OSs running on the hypervisors as VMs. Some hypervisors run on top of the host machine OS as a process, such as VirtualBox an d KVM, but for our purposes the distinction is minor.

Virtualization

- Virtualization -- the abstraction of computer resources.
- Virtualization hides the physical characteristics of computing resources from their users, be they applications, or end users.
- This includes making a single physical resource (such as a server, an operating system, an application, or storage device) appear to function as multiple virtual resources; it can also include making multiple physical resources (such as storage devices or servers) appear as a single virtual resource.

Virtualization Basic Advantages

- For the cloud operator, virtualization has huge advantages.
- First, the cloud provider can provide dozens of different operating systems packaged as VMs for the user to choose from.
- To the hypervisor, all VMs look the same and can be managed in a uniform way.
- The cloud management system (sometimes called the fabric controller) can select which server to use to run the requested VM instances, and it can monitor the health of each VM.
- If needed, the cloud monitor can run many VMs simultaneously on a single server.
- If a VM instance crashes, it does not crash the server. The cloud monitor can record the event and restart the VM.
- User applications running in different VMs on the same server are largely unaware of each other.

Virtualization

- Virtualization is creation of an alternative to actual version of something:
  - virtual memory (more memory than physically available),
  - virtual time (buffering provides virtual/effective download time that less than the actual time),
  - Virtual hardware, desktop, disk, appliances, scenes,
  - Virtual worlds
- In our context it is realizing one or more complete computer systems as guests on the base machine/operating system.
- This offers an excellent conduit for delivering the vastly underutilized power of the multi-core and other resources such as storage and devices.

THEREFORE, WE SAY…

VIRTUALIZATION IS INDEED

A PART OF CLOUD

(AND US)

Virtualization

"We all live every day in virtual environments, defined by our ideas."

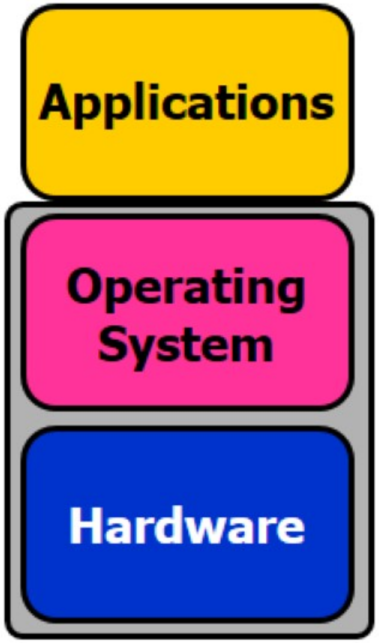
—Michael Crichton

Virtualization

"We all live every day in virtual environments, defined by our ideas."

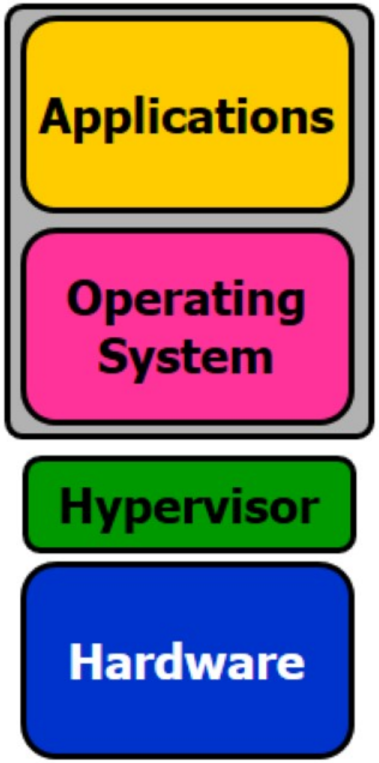
—Michael Crichton

The Use of Computers



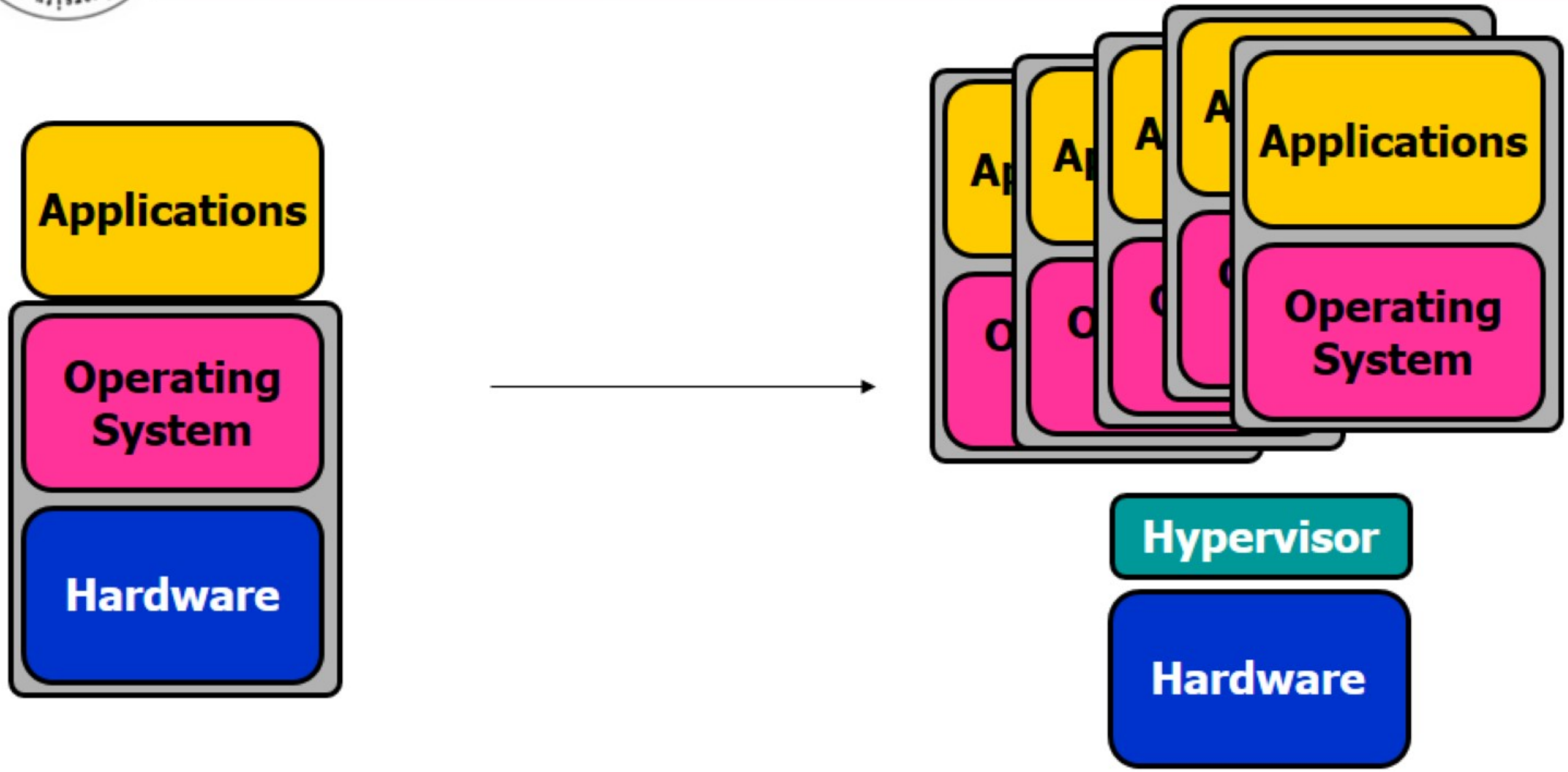
Traditional Server

Virtualization



With Virtualization

Virtualization -- a Server for Multiple Applications/OS



Hypervisors And Hosts

- A **hypervisor** is a piece of computer software, firmware or hardware that creates and runs virtual machines.
- **Hypervisor** is a software program that manages multiple operating systems (or multiple instances of the same operating system) on a single computer system.
- The hypervisor manages the system's processor, memory, and other resources to allocate what each operating system requires.
- Hypervisors are designed for a particular processor architecture and may also be called **virtualization managers** or Virtual machine Monitors (VMM)

Virtual Machine

- A virtual machine (VM) is a software program or operating system that not only exhibits the behavior of a separate computer, but is also capable of performing tasks such as running applications and programs like a separate computer.
- A virtual machine, usually known as a guest is created within another computing environment referred as a "host." Multiple virtual machines can exist within a single host at one time.
- A Virtual machine (VM) is an isolated runtime environment (guest OS and applications)

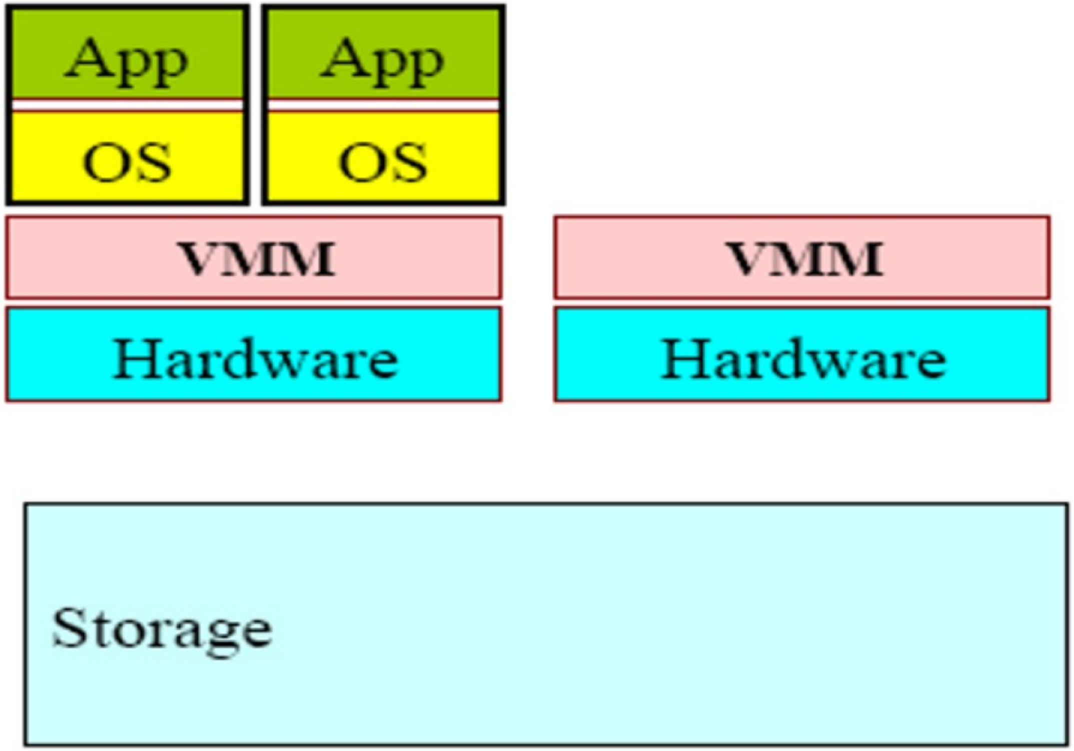
https://www.techopedia.com/definition/4805/virtual-machine-vm

Hypervisors And Hosts

- A **hypervisor** is a piece of computer software, firmware or hardware that creates and runs virtual machines.
- A computer on which a hypervisor is running one or more virtual machines is defined as a **host machine**.
- Each **virtual machine** has a **guest operating system**, which is managed by the hypervisor.
- Multiple instances of a variety of operating systems may share the virtualized hardware resources.

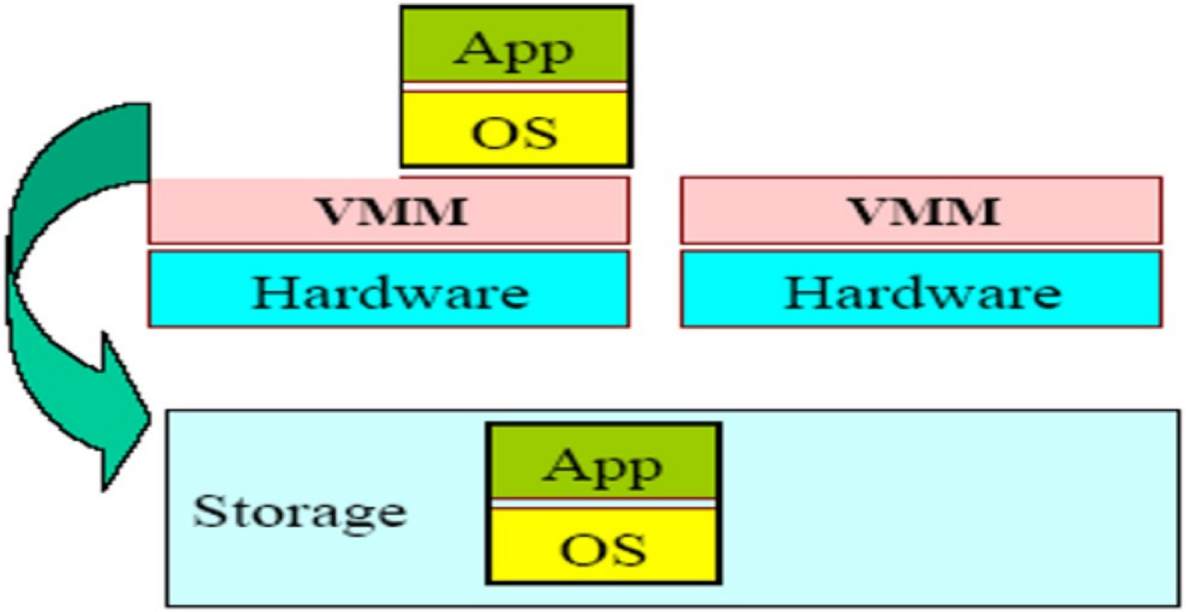
Low-Level VMM Operations (1)

•Multiplex



(Courtesy of Mendel Rosenblum, 2006)

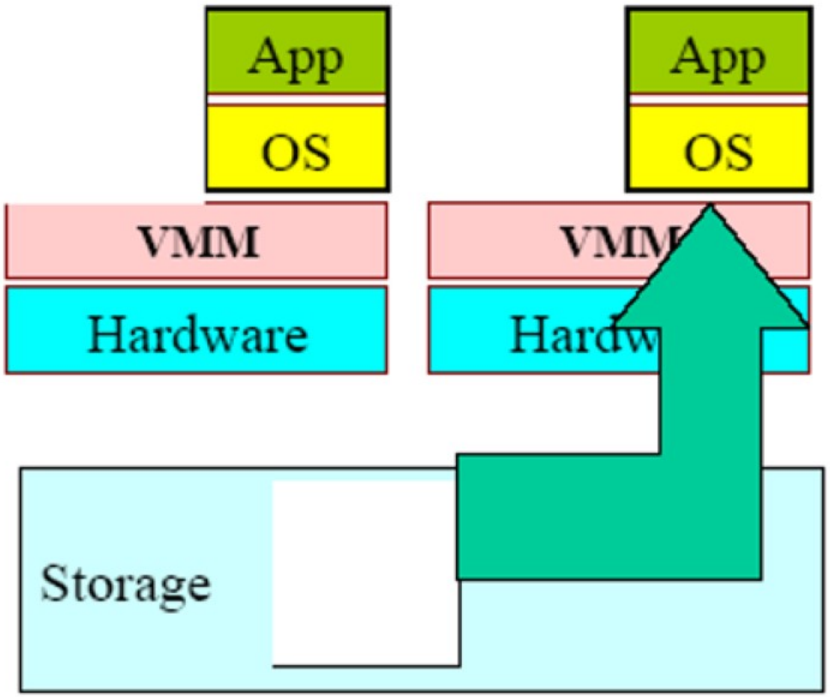
Low-Level VMM Operations (2)



- Multiplex
- Suspend

(Courtesy of Mendel Rosenblum, 2006)

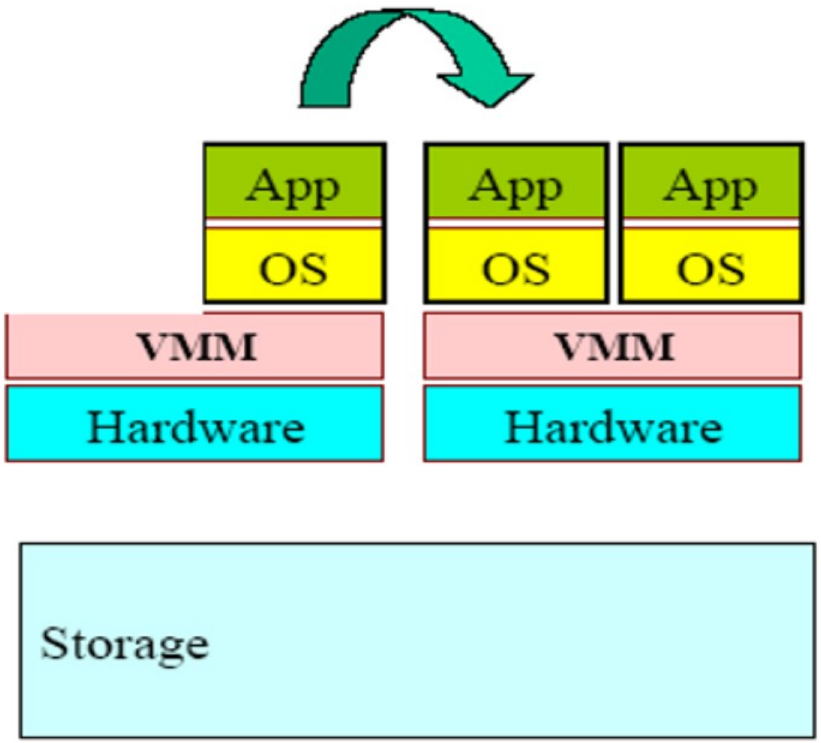
Low-Level VMM Operations (3)



- Multiplex
- Suspend
- Resume (Provision)

(Courtesy of Mendel Rosenblum, 2006)

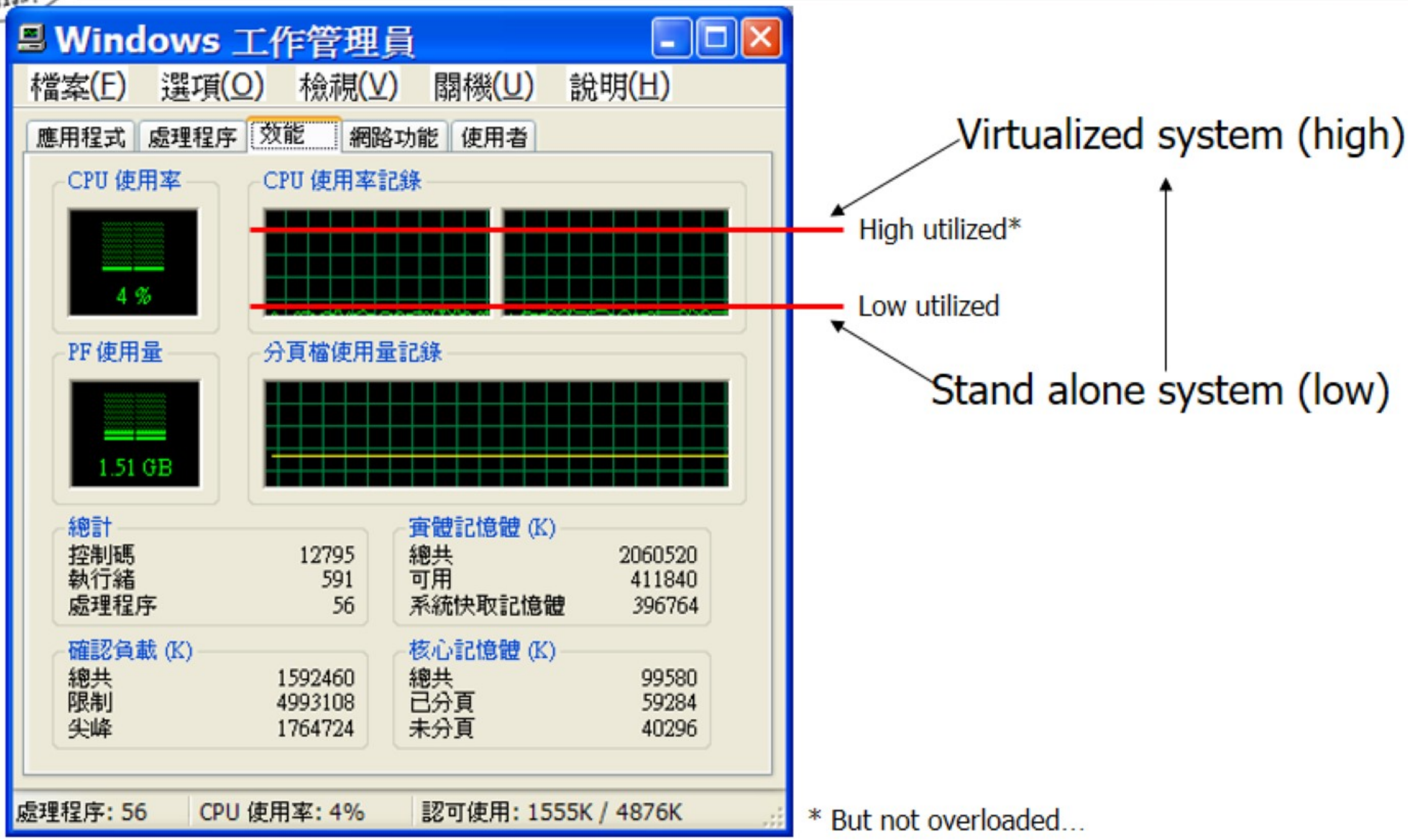
Low-Level VMM Operations (4)



- Multiplex
- Suspend
- Resume (Provision)
- Migration

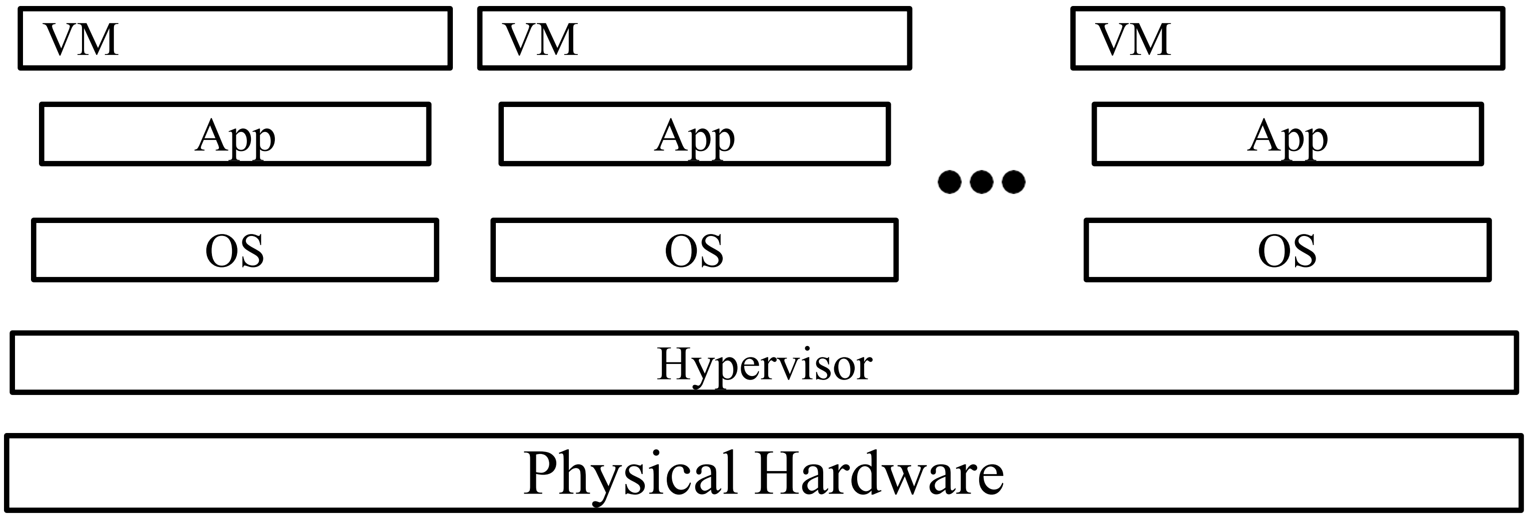
(Courtesy of Mendel Rosenblum, 2006)

Capacity Utilization



* But not overloaded…

Problems with Virtualization



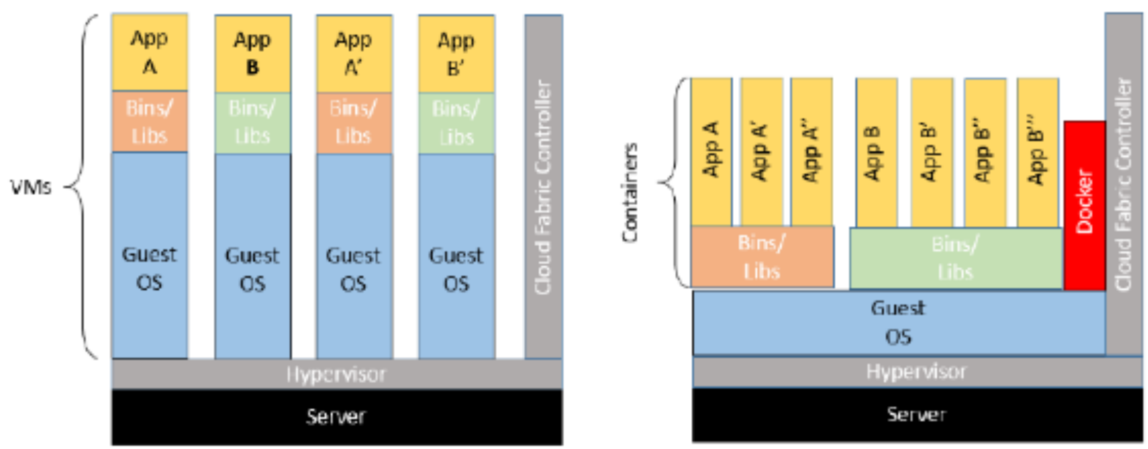
- Each VM requires an operating system (OS)
  - Each OS requires a license ⇒ CapEx
  - Each OS has its own compute and storage overhead
  - Needs maintenance, updates ⇒ OpEx
  - VM Tax = added CapEx + OpEx

Containers

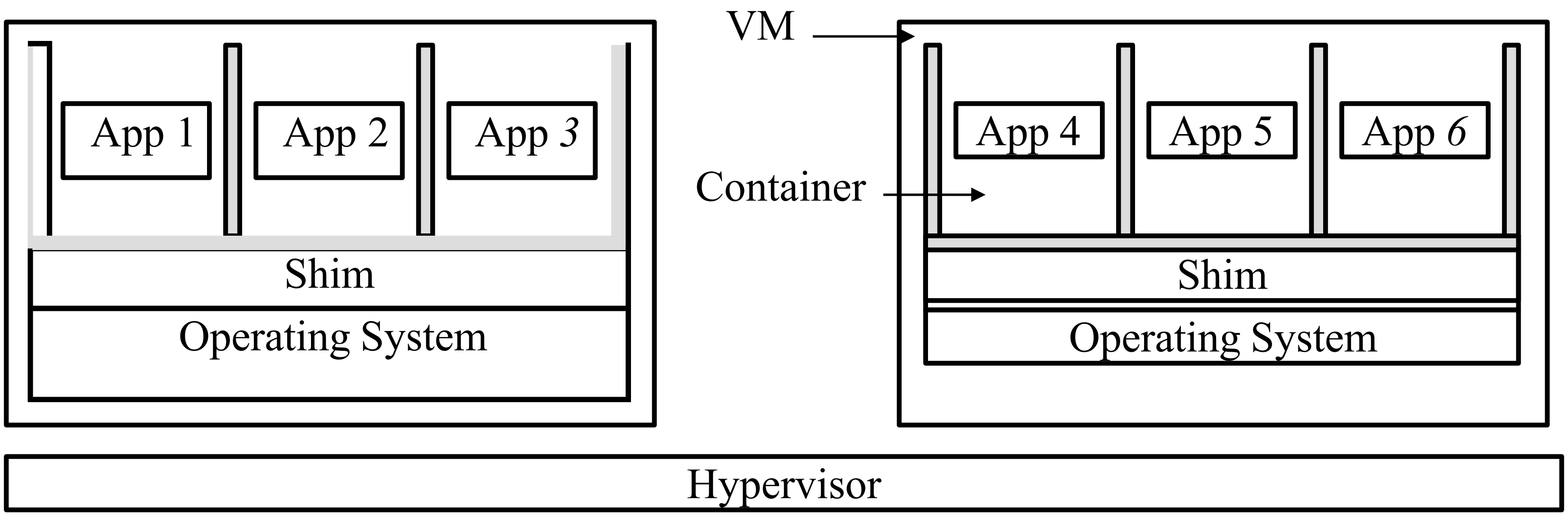
- Run many apps in the same virtual machine
  - These apps share the OS and its overhead
  - But these apps can't interfere with each other
  - Can't access each other's resources without explicit permission
  - Like apartments in a complex
  - ⇒ Containers

Containers

- Containers are similar to VMs but are based on a different technology and serve a slightly different purpose.
- Rather than run a full OS, a container is layered on top of the host OS and uses that OS's resources in a clever way.
- VM on the other hand facilitates the use of heterogeneous operating systems so comprise a **complete OS copy** with libraries and application
- Containers allow you to package up an application and all of its library dependencies and data into a single, easy-to-manage unit



Containers



- Multiple containers run on one operating system on a virtual/physical machine
- All containers **share the operating system** ⇒ CapEx and OpEx
- Containers are isolated ⇒ cannot interfere with each other
  - Own file system/data, own networking ⇒ Portable

Containers

- Containers have all the good properties of VMs
  - Come complete with all files and data that you need to run
  - Multiple copies can be run on the same machine or different machine ⇒ Scalable
  - Same image can run on a personal machine, in a data center or in a cloud
  - Operating system resources can be restricted or unrestricted as designed at container build time
  - Isolation: For example, "Show Process" (ps on Linux) command in a container will show only the processes in the container
  - Can be stopped. Saved and moved to another machine or for later run

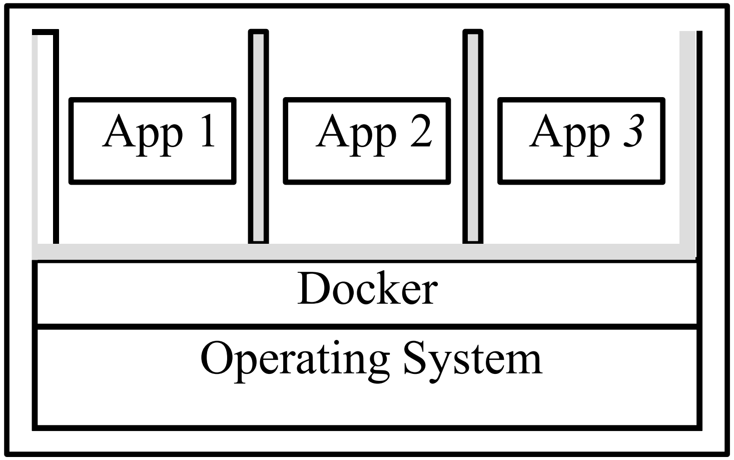
A Comparison

Criteria	VM	Containers
Image Size	3X	X
Boot Time	>10s	~1s
Computer Overhead	>10%	<5%
Disk I/O Overhead	>50%	Negligible
Isolation	Good	Fair
Security	Low-Medium	Medium-High
OS Flexibility	Excellent	Poor
Management	Excellent	Evolving
Impact on Legacy application	Low-Medium	High

Ref: M. K. Weldon "The Future X Network: A Bell Labs Perspective," CRC Press, 2016, 476 pp., ISBN:9781498779142

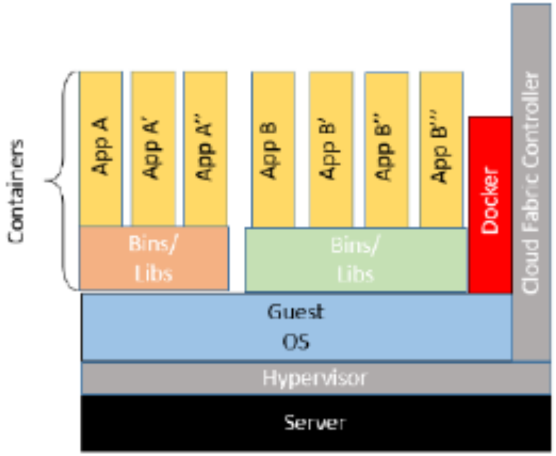
Docker

- Provides the isolation among containers
- Helps them share the OS
- Docker = Dock worker ⇒ Manage containers
- Developed initially by Docker.com
- Downloadable for Linux, Windows, and Mac from Docker.com
- Customizable with replacement modules from others



Containers

- When you launch the container, the application can be configured to start up, go through its initialization, and be running in seconds.
- For example, you can run a web server in one container and a database server in another; these two containers can discover each other and communicate as needed.
- Or, if you have a special simulation program in a container, you can start multiple instances of the container on the same host.
- ***Docker*** is a tool designed to make it easier to create, deploy, and run applications by using containers.



Containers Advantages

- Containers have the advantage of being extremely lightweight.
- Once you have downloaded a container to a host, you can start it and the application(s) that it contains quasi-instantly.
- Part of the reason for this speed is that a container instance can share libraries with other container instances.
- VMs, because they are complete OS instances, can take a few minutes to start up.
- You can run many more containers on a single host machine than you can effectively run the same number of VM

Containers Advantages

- Building a container to run a single application is simple compared with the task of customizing a VM to run a single application.
- All you need to do is create a script that identifies the needed libraries, source files, and data.
- You can then run the script on your laptop to test the container, before uploading the container to a repository, from where it can be downloaded to any cloud.
- Importantly, containers are completely portable across different clouds.
- Whereas, In general, VM images cannot be ported from one cloud framework to another.

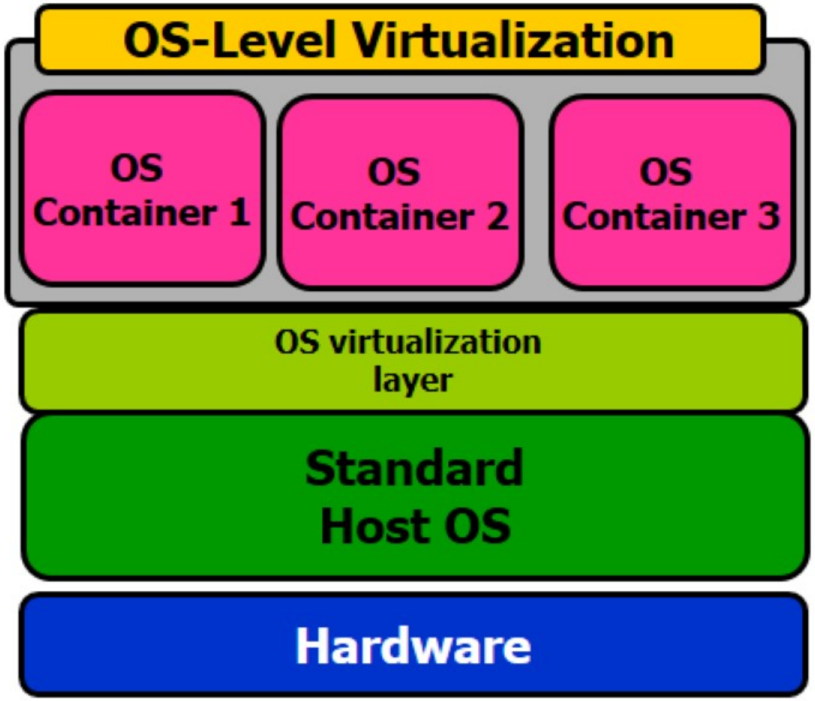
Containers Problems

- Containers also have downsides.
- The most serious issue is security.
- Because containers share the same host OS instance, two containers running on the same host are less isolated than two VMs running on that host.
- Managing the network ports and IP addresses used by containers can be slightly more confusing than when working with VMs.
- Furthermore, containers are often run on top of VMs, which can exacerbate the confusion .

Virtual machines	Containers
Heavyweight	Lightweight
Fully isolated; hence more secure	Process-level isolation; hence less secure
No automation for configuration	Script-driven configuration
Slow deployment	Rapid deployment
Easy port and IP address mapping	More abstract port and IP mappings
Custom images not portable across clouds	Completely portable

OS-Level Virtualization

- OS-level virtualization
  - kernel of an OS allows for multiple isolated user-space instances, instead of just one.
  - Each OS instance looks and feels like a real server
- OS virtualization virtualizes servers on the operating system (kernel) layer. This creates isolated containers on a single physical server and OS instance to utilize hardware, software, data center and management efforts with maximum efficiency.
- <u>OS-level virtualization implementations that are capable of live migration can be used for dynamic load balancing of containers between nodes in a cluster.</u>



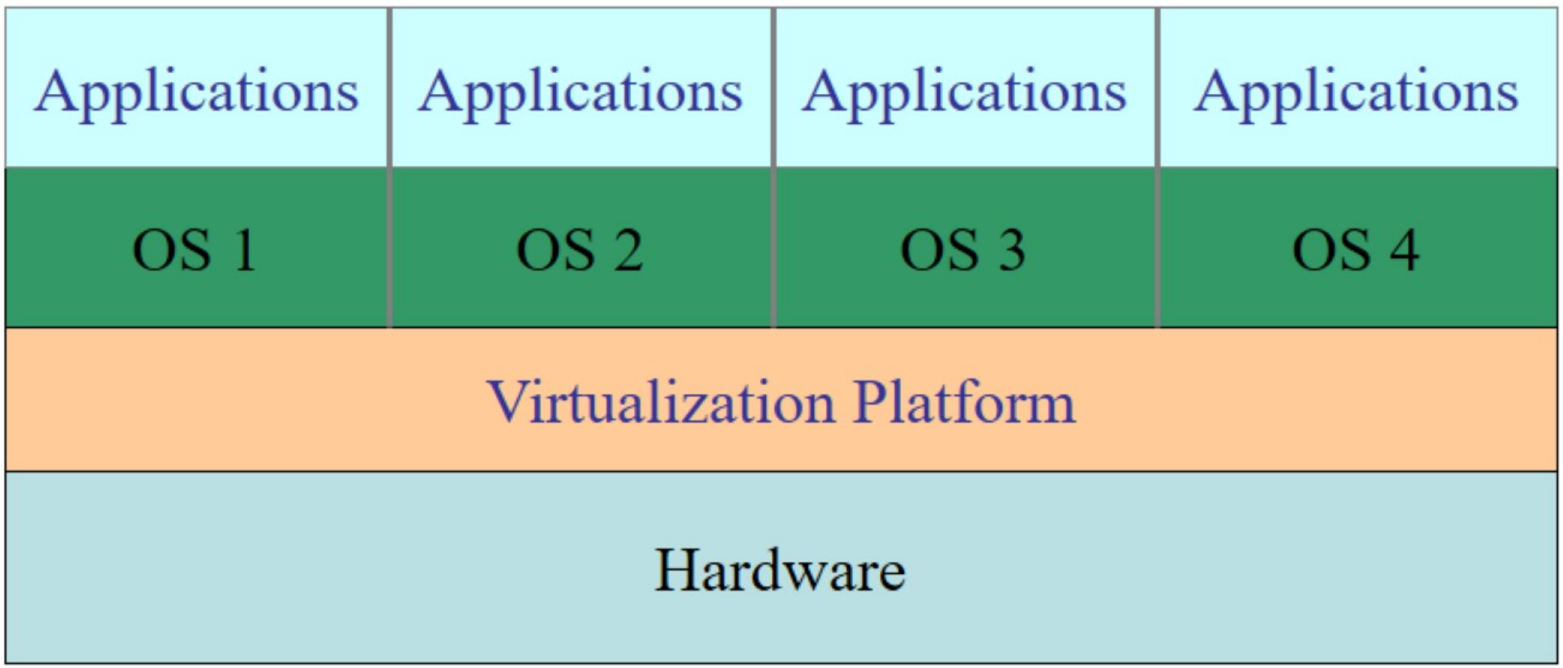
Operating Systems Limtations

- OSs provide a way of virtualizing hardware resources among *processes*
- This may help isolate *processes* from one another
- However, this does not provide a *virtual machine* to a user who may wish to run a different OS
- Having hardware resources managed by a single OS limits the flexibility of the system in terms of available software, security, and failure isolation
- Virtualization typically provides a way of relaxing constraints and increasing flexibility

What is OS Virtualization?

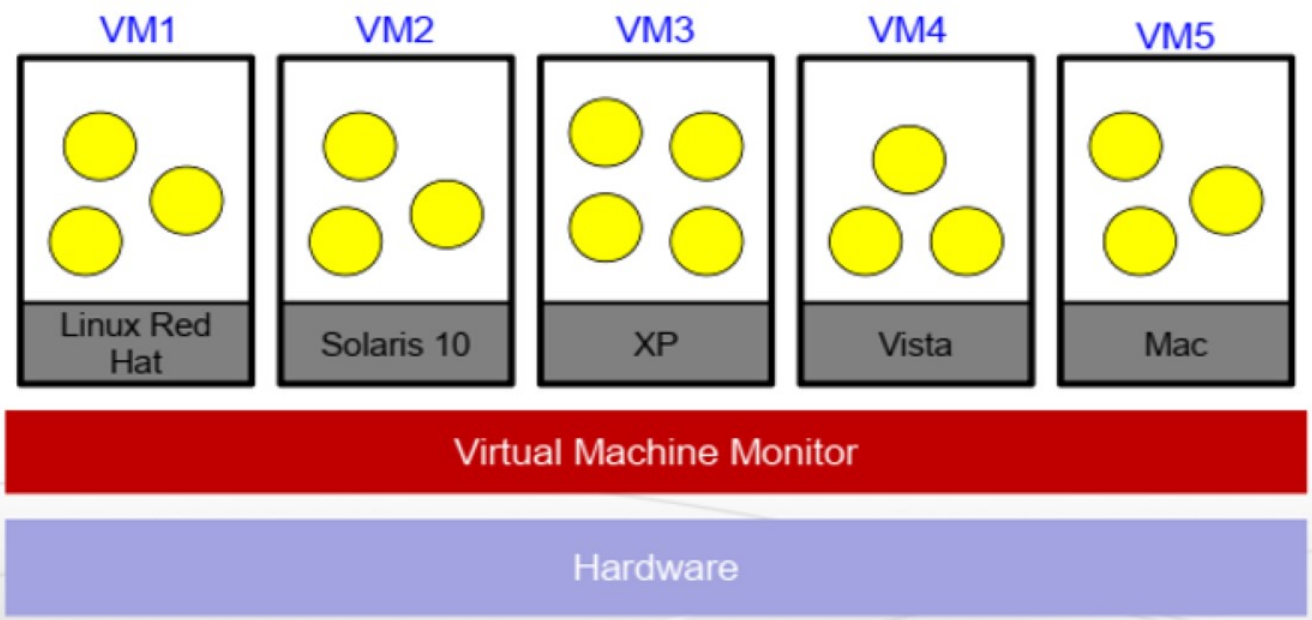
- OS virtualization
  - Create a platform that emulates a hardware platform and allow multiple instances of an OS to use that platform, as though they have full and exclusive access to the underlying hardware

OS Virtualization?



A Mixed OS Environment

- Multiple VMs can be implemented on a single hardware platform to provide individuals or user groups with their own OS environments



Confusion…

- **OS-Level Virtualization.** A type of server virtualization technology which works at the OS layer. The physical server and single instance of the operating system is virtualized into multiple isolated partitions, where each partition replicates a real server. The OS kernel will run a single operating system and provide that operating system functionality to each of the partitions.
- Operating system virtualization refers to the use of software to allow system hardware to run multiple instances of different operating systems concurrently, allowing you to run different applications requiring different operating systems on one computer system. The operating systems do not interfere with each other or the various applications.

WHY NOW???

Hardware evolution

- Faster CPU clock than ever
  - Though almost hit its top
- More CPU cores in a single chip
  - 4-core CPUs already in the market
  - 6- or 8-core CPUs will be there soon
- Multi-core architectures make parallel processing more realizable
- Virtualization support on chip from CPU manufacturers (e.g., Intel, AMD)

Software maturity

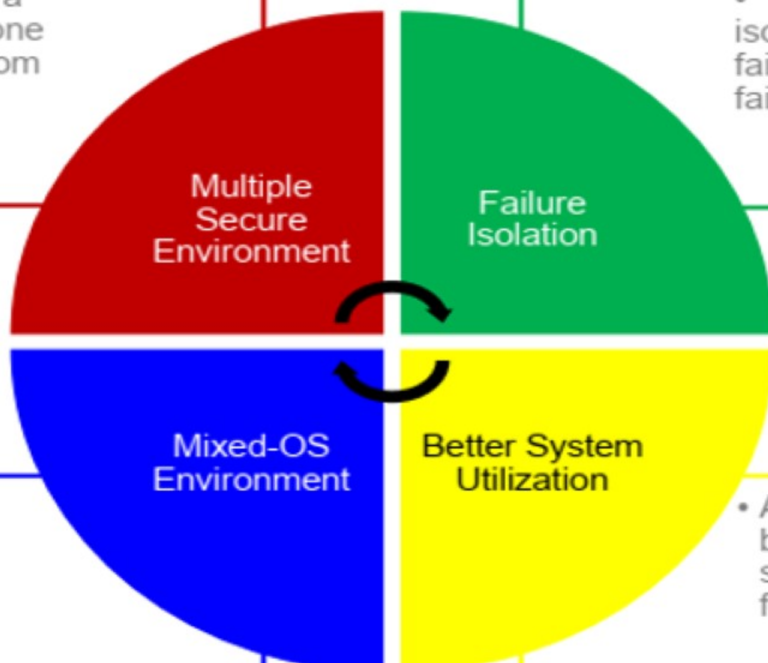
- More than one credible player in the market
- Available and stable open-sourced software
  - OS, DB, Web server, Java, PHP, gcc, etc.
- Established and mature software standards
  - Web service, XML, SOAP, COM, etc.

Benefits of Virtualization

- Here are *some* of the benefits that are typically provided by a virtualized system

- A system VM provides a sandbox that isolates one system environment from other environments

- Virtualization helps isolate the effects of a failure to the VM where the failure occurred



- A single hardware platform can support multiple operating systems concurrently

- A virtualized system can be (dynamically or statically) re-configured for changing needs

Virtualization Properties

- Fault Isolation
- Software Isolation
- Performance Isolation (accomplished through scheduling and resource allocation)

Isolation

1

- All VM state can be captured into a file (i.e., you can operate on VM by operating on file– cp, rm)
- Complexity is proportional to virtual HW model and independent of guest software configuration

Encapsulation

2

- All guest actions go through the virtualizing software which can inspect, modify, and deny operations

Interposition

3

What is Virtualization?

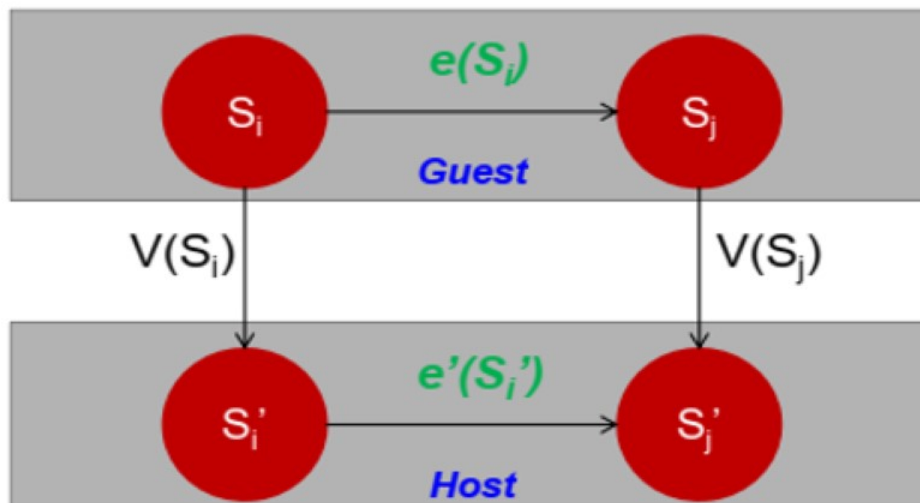
- Informally, a virtualized system (or subsystem) is a *mapping* of its interface, and all resources visible through that interface, to the interface and resources of a real system

- Formally, virtualization involves the construction of an isomorphism that *maps* a virtual ***guest*** system to a real ***host*** system (Popek and Goldberg 1974)

✓ Function V maps the guest state to the host state

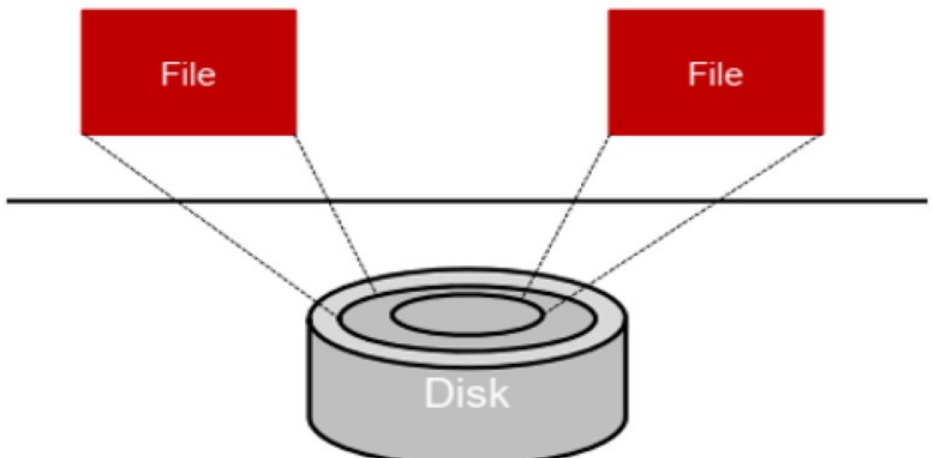
✓ For a sequence of operations, ***e***, that modifies a guest state, there is a corresponding ***e'*** in the host that performs an equivalent modification

✓ How can this be managed?



Abstraction

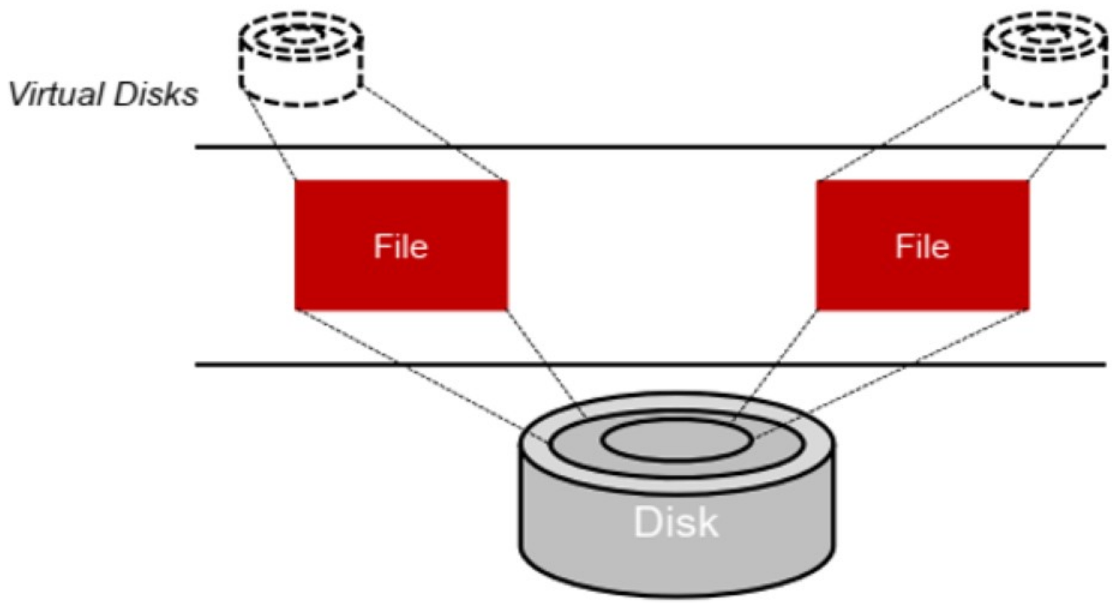
- The key to managing complexity in computer systems is their division into ***levels of abstraction*** separated by ***well-defined interfaces***
- Levels of abstraction allow implementation details at lower levels of a design to be ignored or simplified



- ✓ **Files are an abstraction of a Disk**
- ✓ **A level of abstraction provides a simplified interface to underlying resources**

Virtualization and Abstraction

- Virtualization uses abstraction but is different in that it doesn't necessarily hide details; the level of detail in a virtual system is often the same as that in the underlying real system



✓ *Virtualization provides a different interface and/or resources at the same level of abstraction*

Virtualization Requirements

- Virtualization requirements (by Popek and Goldberg) :
  - Equivalence: a program running under the VMM should exhibit a behavior essentially identical to that demonstrated when running on an equivalent machine directly;
  - Resource control (safety): the VMM must be in complete control of the virtualized resources;
  - Efficiency: a statistically dominant fraction of machine instructions must be executed without VMM intervention.

VMM: Virtual Machine Monitor

Types of Virtualization

- Virtual memory
- Desktop virtualization
- Platform virtualization
  - Full virtualization
  - Paravirtualization
  - Hardware-assisted virtualization
  - Partial virtualization
  - OS-level virtualization
  - Hosted environment (e.g. User-mode Linux)
- Storage virtualization
- Network virtualization
- Application virtualizationPortable application
  - Cross-platform virtualization
  - Emulation or simulation
  - Hosted Virtual Desktop

Category in Wiki

Additional Resources

- Amazon Boto3 `aws.amazon.com/sdk-for-python/`
- Azure `azure.microsoft.com/en-us/develop/python/`
- Google's Cloud `cloud.google.com/sdk/`
- Openstack CloudBridge `cloudbridge.readthedocs.io/en/latest/`
- The Globus Python SDK `github.com/globus/globus-sdk-python` and the Globus CLI `github.com/globus/globus-cli`

https://www.cse.wustl.edu/~jain/cse570-18/m_21cdk.htm