

# CS370 Operating Systems

Colorado State University

Yashwant K Malaiya

Fall 2021 L26

Virtualization



## Slides based on

- Text by Silberschatz, Galvin, Gagne
- Various sources

## Why do we need to study .....

- Why do we need to study advanced topics like HDFS, Virtualization, Containers etc.? Because ...
- Why did we need to study several simple schemes with toy examples for:
  - CPU Scheduling (FCFS etc.)
  - Synchronization, deadlocks
  - Page replacement algorithms
  - Disk space allocation
- So that you can understand real and advanced schemes.

# FAQ

**Kernel vs OS:** OS = {Kernel, UI, libraries/binaries}

- Include: header files, lib: runtime libraries, bin: executable programs.

Hypervisor Type 1 vs kernel:

- In a simple system, the kernel provides access to hardware resources (memory, secondary storage, ports/network)
- With Hypervisor Type 1, the kernel thinks it is providing access to the hardware resources. In reality, the hypervisor is managing the illusion.
- With Hypervisor Type 2, the guest kernel the kernel thinks it is providing access to the hardware resources. In reality, the hypervisor is managing the illusion, by passing the responsibility to the host kernel/

# FAQ

## Virtualization:

- How to allocate vCPUs to virtual machines?
  - Guidelines/experience. pCPU:vCPU is often 1:1 to 1:3 or more. A VM may not use a CPU all the time.
- How much memory should be allocated per VM? Depends. 0.5- 1GB min?
- Running a VM inside a VM?
  - Possible for modern CPUs. Restrictions possible for older.
- How can V machines run concurrently? How can two OSs run at the same time? Just like processes ..

# FAQ

## HDFS

- Hadoop Distributed File System: Large storage, large files, distributed and replicated storage
- 64MB blocks saved as 4KB blocks of underlying FS (ext4 etc), likely on different nodes, allowing access in parallel.
- Why 3 copies? 1 replica: MTDL  $\approx$  1 year, 2 replicas  $\approx$  10 years, 3 replicas  $\approx$  100 years assuming independent failures



- Final: comprehensive but questions will mostly be from the second half.
- Questions of various types
- Sec 001 (& local 801)
  - Tues Dec 14, 6:20-8:20pm
  - SDC: schedule with them for Dec 14 (4-8 PM window)
- Sec 801 (non-local)
  - 2 hours,
  - window Tues Dec 14, 6:20PM-Wed 8:20pm.
  - SDC: special arrangements

# Updates

- Project:
  - Final report: 12/2/21, 2-columns, citations
    - Slides/Videos also needed on Project Slides channel
    - Peer reviews needed
  - Specific requirements for Option A and Option B
  - Option B: 15 minutes demos each team
    - Sign-up sheet available on MS Team for Mon, Tu, Wed
    - Demos using MS Team recorded
- Project day: next Tuesday
- Course review: next Thursday

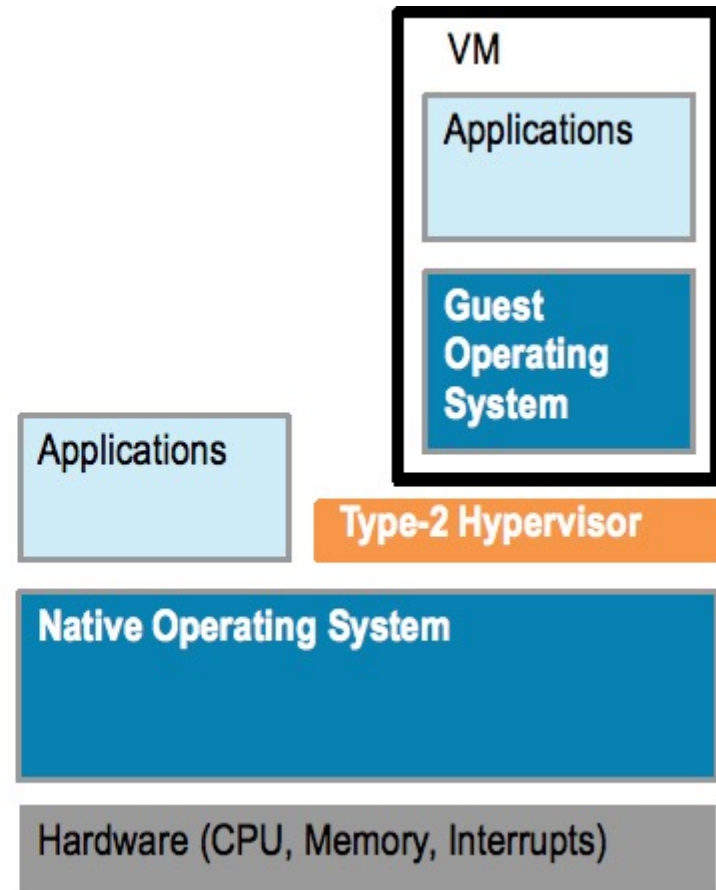
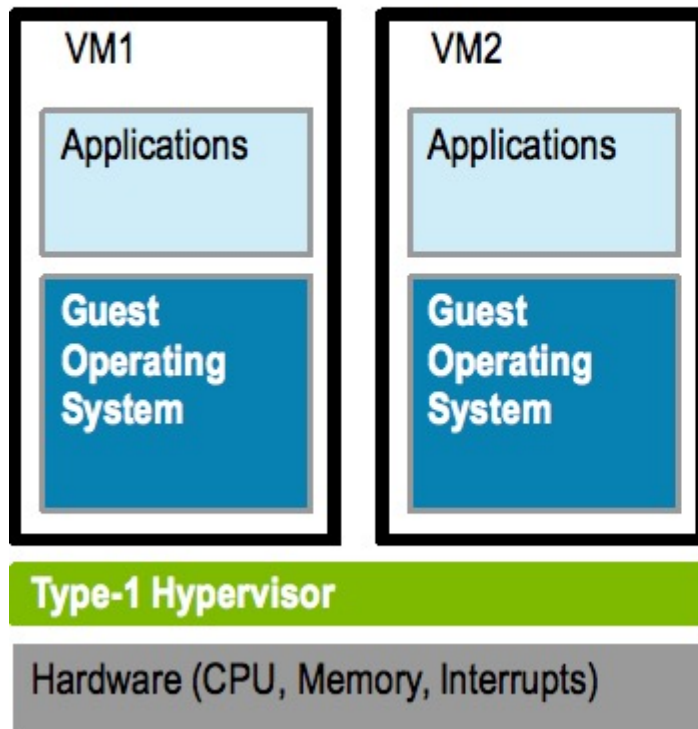
# Course Survey

- Will be available on Canvas.



# Implementation of VMMs

*A higher layer uses services of the lower layers.*



<https://microkerneldude.files.wordpress.com/2012/01/type1-vs-2.png>

# OS vs Hypervisor vs Containers

- **Isolation**
  - OS isolates processes from each other
  - Hypervisor isolate VMs from each other
  - **Container manager isolates containers from each other**
- **Resource allocation**
  - OS: allocates resources to processes
  - Hypervisor allocates resources to Virtual Machines
  - **Container manager allocates resources to containers**

Information on containers coming up.

# Memory Management

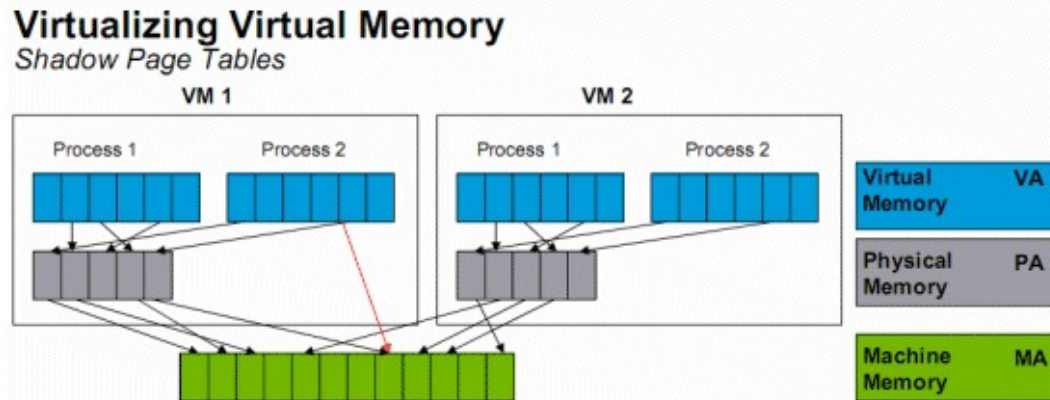
Memory mapping:

- On a bare metal machine: OS uses page table/TLB to map Virtual page number (VPN) to Physical page number (PPN) (physical memory is shared). Each process has its own page table/TLB.

— VPN -> PPN

- VMM: Real physical memory (*machine memory*) is shared by the OSs. Need to map PPN of each VM to MPN (Shadow page table)

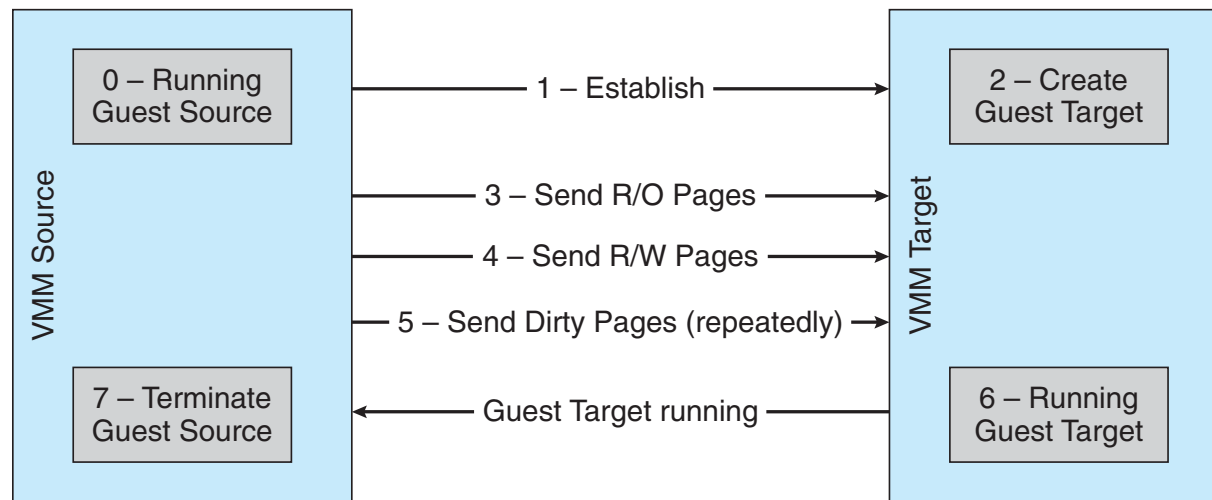
PPN -> MPN



# Virtual Machine (VM) as a set of files

- Configuration file describes the attributes of the virtual machine containing
  - server definition,
  - how many virtual processors (vCPUs)
  - how much RAM is allocated,
  - which I/O devices the VM has access to,
  - how many network interface cards (NICs) are in the virtual server
  - the storage that the VM can access
- When a virtual machine is instantiated, additional files are created for logging, for memory paging etc.
- Copying a VM produces not only a backup of the data but also a copy of the entire server, including the operating system, applications, and the hardware configuration itself

# Live Migration



- Migration from source VMM to target VMM
  - Source establishes a connection with the target
  - Target creates a new guest
  - Source sends all read-only memory pages to target
  - Source starts sending all read-write pages
  - Source VMM freezes guest, sends final stuff,
  - Once target acknowledge that guest running, source terminates guest.

# VIRTUAL APPLIANCES: “shrink-wrapped” virtual machines

- Developer can construct a virtual machine with
  - required OS, compiler, libraries, and application code
  - Freeze them as a unit ... ready to run
- Customers get a complete working package
- Virtual appliances: “shrink-wrapped” virtual machines
- Amazon’s EC2 cloud offers many pre-packaged virtual appliances examples of *Software as a service*
- *Question: do we really have to include a whole kernel in a shrink wrapped VM?*

# HW7 Review

# CS370 Operating Systems

Colorado State University

Yashwant K Malaiya

Fall 2021



## Containers

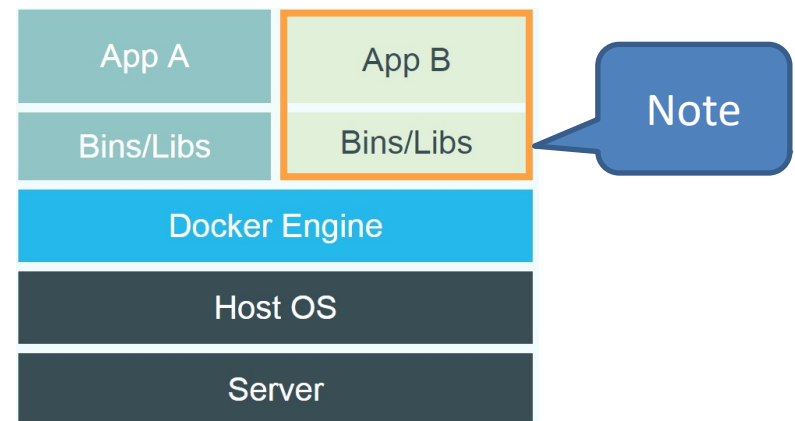
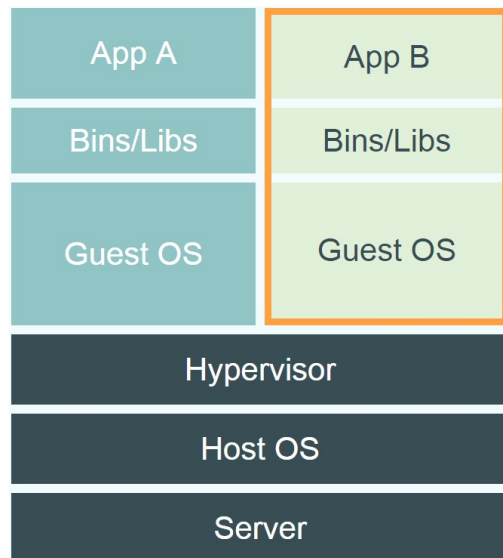
Slides based on

- Text by Silberschatz, Galvin, Gagne
- Various sources



# Linux Containers and Docker

- Linux containers (LXC) are “lightweight” VMs
- Comparison between LXC/docker and VM



- Containers provide “OS-level Virtualization” vs “hardware level”.
- Containers can be deployed in seconds.
- Very little overhead during execution, even better than Type 1 VMM.

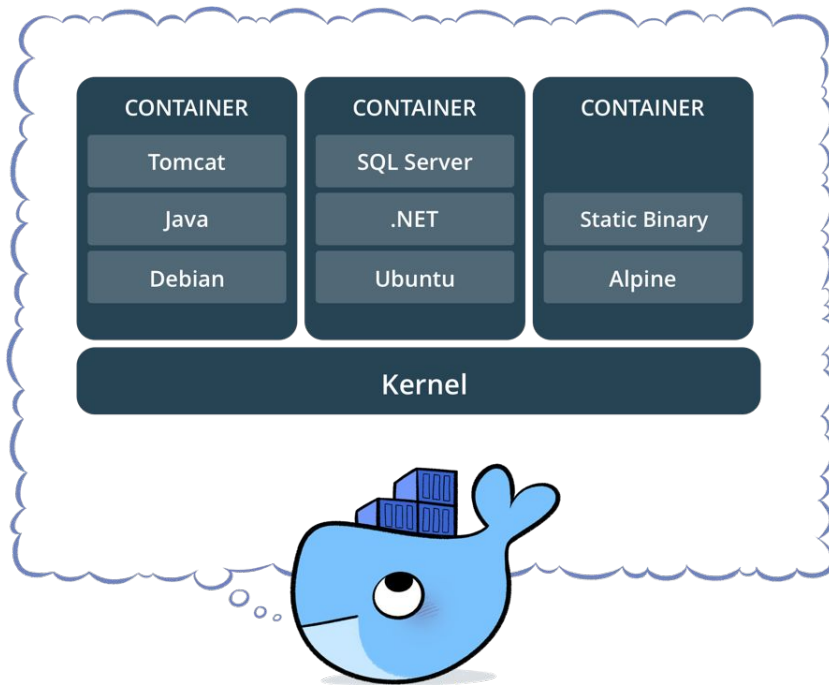
# Container: basis

Linux kernel provides

- “control groups” (cgroups) functionality for a set of processes
  - allows allocation and prioritization of resources (CPU, memory, block I/O, network, etc.) without the need for starting any VM
- “namespace isolation” functionality
  - allows complete isolation of an applications' view of the operating environment including Process trees, networking, user IDs and mounted file systems.
- Managed by Docker etc.
  - Docker: build, share, run and orchestrate containerized apps.
  - Kubernetes: orchestration platform for managing, automating, and scaling containerized applications

# Container

## What is a container?



- Standardized packaging for software and dependencies
- Isolate apps from each other
- Share the same OS kernel
- Works for all major Linux distributions
- Docker Desktop for Windows uses Windows-native Hyper-V virtualization (Win10)
- Containers native to Windows Server 2016
- Docker: a popular container management service technology

# CS370 Students: Future dreams and nightmares

- Man on Mars
- Other dreams
  - AI with human rights
  - More powerful phones
  - natural language models being used to preserve some of the 50%-90% of human languages that will otherwise disappear by the year 2100
  - I expect to see hoverboards in the next 20 years
  - Vaccination created dynamically from AI
  - Flying cars
- Nightmares
  - Cloud gaming virtually everywhere
  - working class be taken over by robots
  - resource wars in my lifetime caused by human greed and the inability to adapt to growing problems as a species
  - future will have less progression than the past decades
  - implants of small operating systems in human brains within the next 20-40 years. (2)
  - anything that does need human intellect to operate will be done by a computer