Virtual Machines

CS2043 - Spring 2019
February 27

quiz-02-27-19

- You know the drill :)

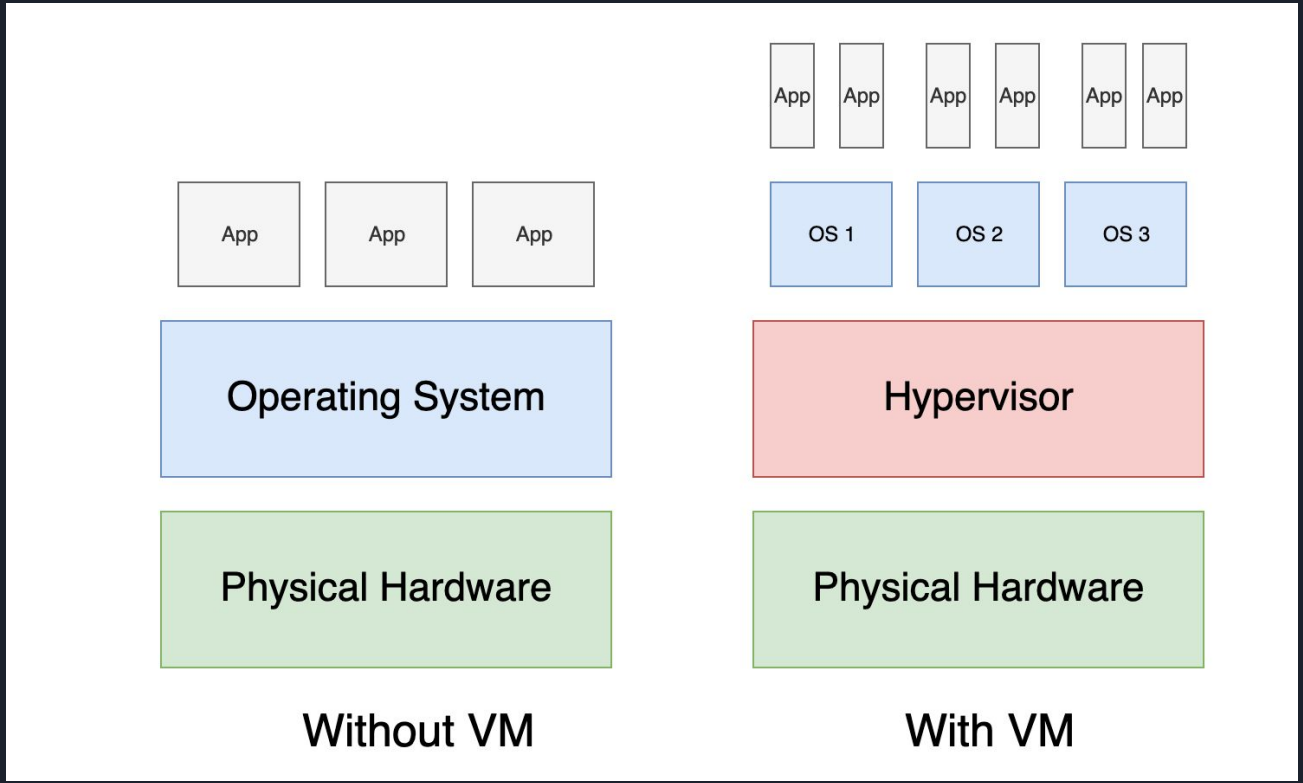
Virtualization

- Virtualization is the creation and use of a virtual (logical) version of a resource
  - Storage, network card, runtime environment, operating system etc.
- Think of it as a layer of indirection
  - Indirection - The ability to reference something using a name, reference, or container instead of the value itself
    - An indirect way to access
    - Basically a middle person, an intermediary step
  - “All problems in Computer Science can be solved by another level of indirection, ” (David Wheeler, first ever Computer Science PhD)

Virtual Machines (VMs)

- System VMs
  - Allow you to run an entire operating system on top of real hardware
  - For example, run a copy of MacOS within Windows 10
  - Oracle VirtualBox, Microsoft Hyper-V, VMware ESXi Server
- Process VMs
  - Allow you to run a single process in a platform-independent environment
  - Similar to what Docker allows you to do
  - Java Virtual Machine (JVM), .NET VM
- Generally when people refer to VMs colloquially they mean system VMs

System VMs, Pictorially



Use Cases

- Security/Testing
  - VMs provide an isolated environment that can easily be deleted
- Backwards compatibility
  - For example, run an older VM to use legacy software
- Cross-platform compatibility
  - Run Xcode without a Mac
- Reduce hardware costs
  - Can run multiple systems without buying new a computer
- Datacenter load balancing
  - Move applications onto fewer physical machines during off-peak hours
- Experiment!!!

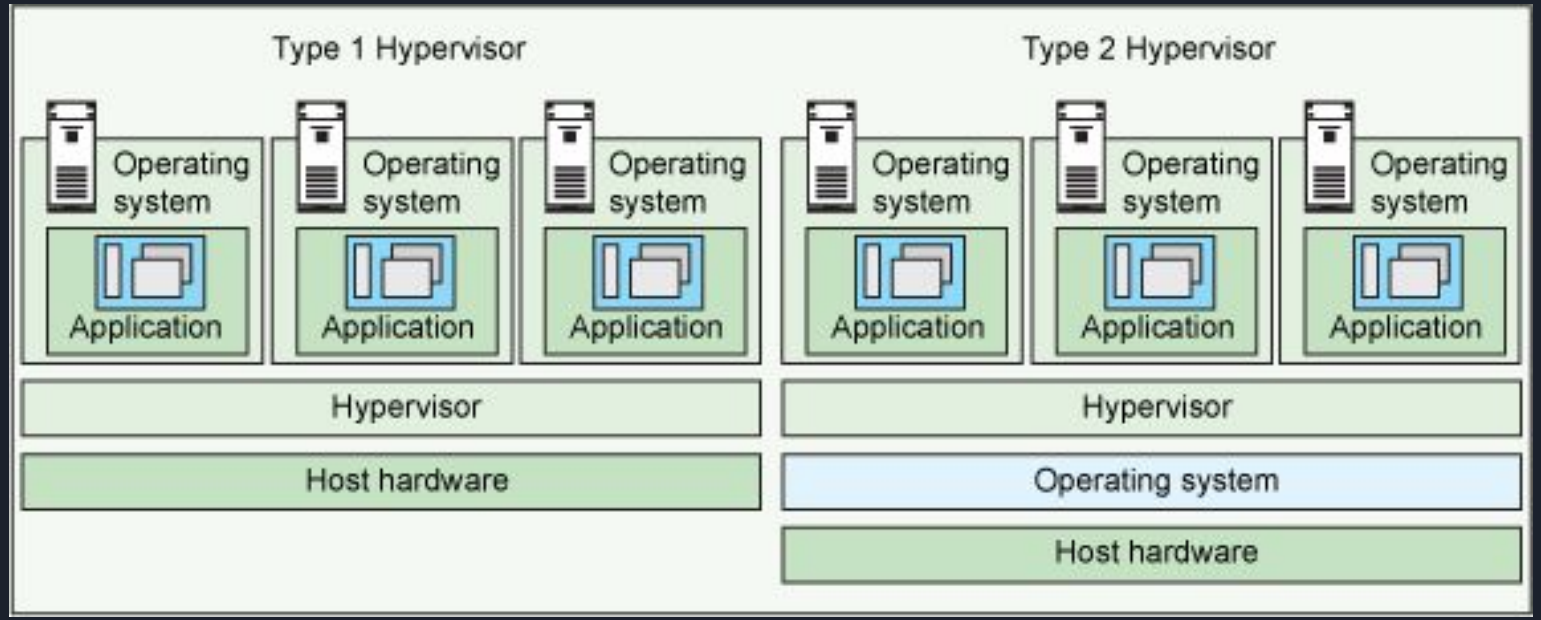
VM Terminology

- Host
  - The physical platform supporting all virtualized components
    - Underlying hardware and (often) operating system
- Guest(s)
  - Virtualized components running on the host
  - Hardware, operating system, applications, etc.
  - In the context of System VMs, refers to an operating system
- Hypervisor
  - Layer between the two that makes all of this possible

More on Hypervisors

- System VMs rely on a program called the “Hypervisor”
  - Also known as a “Virtual Machine Monitor”
- Creates and manages ***virtual*** versions of computer resources (processor, memory, network card, etc.)
  - These virtual versions are the layer of indirection - access physical resources via them
- Creates the illusion that the VM operating system has exclusive access to physical hardware

Type 1 vs Type 2, Pictorially



Type 1 vs Type 2 Hypervisors

- Type 1
  - Bare metal / Embedded / Native Hypervisor
  - Installed and ran directly on top of hardware
    - Means no operating system has direct access to hardware
  - More of a server solution
  - Microsoft Hyper-V, VMware ESXi Server, Xen
- Type 2
  - Hosted Hypervisor
  - Installed and ran as a program within a host operating system
  - More of a consumer/desktop solution
  - Oracle Virtualbox

Emulation and API Re-Implementation

- With System VMs, Guest OS runs on physical host hardware
- Emulation
  - Uses software to implement the functionality of some hardware
  - **Emulate** the hardware rather than running on a physical copy of it
  - Video game emulators, mobile app simulations
- API Re-implementation
  - Uses an identical interface but provides a different implementation
  - I.e., the caller is able to perform the exact same set of operations but under the hood things have changed
  - WINE - Allows you to run Windows programs on Linux by translating instruction set on the fly

Virtualbox Demo

- Download Virtualbox
  - https://www.virtualbox.org/wiki/Downloads
- Install Ubuntu on Virtualbox
  - https://www.lifewire.com/run-ubuntu-within-windows-virtualbox-2202098
- See references [4]

References

- Content partially inspired by G. Edward Suh, Cornell Computer Systems Lab (ECE 5770 Fall 2018 course material)
- Additional resources:
  1. https://www.redhat.com/en/topics/virtualization/what-is-virtualization
  2. http://techgenix.com/virtual-machines/
  3. https://developer.ibm.com/articles/cl-hypervisorcompare/
  4. https://www.virtualbox.org/manual/ch01.html