Virtualized Environments

Tyler Bletsch
Duke University
Virtualization

- Virtualize each layer of stack to pool resources; individual systems stop mattering
- Fundamental concept: **aggregate physically** and **separate logically**

**Aggregate**: Cluster disk-less interchangeable servers
**Separate**: Run virtual machines (VMs) that can freely migrate

**Aggregate**: Switches paired and interconnected with cables
**Separate**: Virtual LANs (VLANs) separate traffic flows

**Aggregate**: Disks combined with RAID and linear mapping
**Separate**: Logical volumes created on top
Server virtualization
Multiple VMs in One Machine

Adapted from "Virtualization Techniques" by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
History of Virtualization

1964 IBM CP-40
1972 IBM VM/370
1997 Virtual PC
1999 VMware
2003 Xen
2005 Intel VT
2006 AMD VT
2007 KVM-X86

Time Sharing
Virtual Memory

Mobile Virtualization

Adapted from "Virtualization Techniques" by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
Example: Server Virtualization

http://www.energystar.gov/index.cfm?c=power_mgt.datacenter_efficiency_virtualization

Adapted from "Virtualization Techniques" by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
Benefits of Server Virtualization

• Virtualization can reduce data center energy expenses by 10%–40%
  • Each physical machine has power overhead, so reducing boxes → reducing power

• Virtualization also improves scalability, reduces downtime, and enables faster deployments.
  • Shared storage means VMs can run on any host → easy failover
  • VM snapshots → faster recovery
  • VM cloning → faster deployment

• Reduce the data center footprint
  • Fewer machines

Adapted from “Virtualization Techniques” by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
Virtualization Techniques

- System Virtualization
  - CPU Virtualization
  - Memory Virtualization
  - I/O Virtualization
  - Hardware Support for Virtualization, e.g. Intel VT
- Storage Virtualization
  - LVM
  - RAID
- Network Virtualization
  - VLANs
  - Software Defined Network

Adapted from "Virtualization Techniques" by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
Types of Virtual Machine

- A virtual machine (VM) is a software implementation of a machine that executes programs like a physical machine. Virtual machines are separated into two major classifications:
  - A system virtual machine
    - Which provides a complete system platform which supports the execution of a complete operating system (OS)
  - A process virtual machine
    - Which is designed to run a single program, which means that it supports a single process.

Adapted from "Virtualization Techniques" by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
System Virtual Machine

- System virtual machine is controlled by a hypervisor or VMM (Virtual Machine Monitor)
- A hypervisor or VMM is a software to provide a hardware emulation interface including CPU, memory, I/O by multiplexing host resources

Adapted from "Virtualization Techniques" by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
In their 1974 article "Formal Requirements for Virtualizable Third Generation Architectures" Gerald J. Popek and Robert P. Goldberg classified two types of hypervisor:

- Type 1 hypervisor: bare metal type
- Type 2 hypervisor: hosted type

http://en.wikipedia.org/wiki/Hypervisor

Adapted from "Virtualization Techniques" by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
Purpose of Hypervisor

- **CPU Virtualization**
  - Handle all sensitive instructions by emulation

- **Memory Virtualization**
  - Allocate guest physical memory
  - Translate guest virtual address to host virtual address

- **I/O Virtualization**
  - Emulate I/O devices for guest
  - Ex: Keyboard, UART, Storage and Network

Adapted from "Virtualization Techniques" by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
Implementations of Hypervisor

- **Full Virtualization**
  - A wholly emulated virtual machine makes guest operating system binary can be executed directly without modifying guest source code
  - For efficiency, it can benefit from hardware-assisted virtualization

- **Para-Virtualization**
  - Hypercalls are defined and used in a guest operating system to make a virtual machine abstraction

- **Pre-Virtualization**
  - By compiling technique, guest operating system binary or source could be compiled for virtualization

Adapted from "Virtualization Techniques" by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
Hypervisor Case: KVM

- CPU and memory virtualization is handled in the Linux Kernel Space
- I/O virtualization is handled in the Linux User Space by QEMU
- It’s a type 2 virtual machine
- It’s a full virtualization implementation

Adapted from "Virtualization Techniques" by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
Hypervisor Case: VMware ESXi

- Without hardware assist, sensitive instructions are dynamically rewritten; with hardware assist, hardware helps trap sensitive instructions to VMM
- It’s a type 1 virtual machine
- It’s a full virtualization implementation

Adapted from “Virtualization Techniques” by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
Intel VT-x

- New CPU Operating Mode
  - VMX Root Operation
  - Non-Root Operation
- New Transitions
  - VM entry to Guest
  - VM exit to VMM
- VM Control Structure
  - Configured by VMM software

Adapted from “Virtualization Techniques” by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
ARM Virtualization Extension

- Secure world supports a single virtual machine
- New Non-secure level of privilege to hold Hypervisor
  - Hypervisor mode applies to normal world
  - Hyp Mode is used by the Hypervisor
  - Guest OS given same kernel/user privilege structure as for a non virtualized environment
- Monitor mode controls transition between worlds

Adapted from "Virtualization Techniques" by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
Storage virtualization
Storage virtualization

• It’s all the stuff we’ve covered so far:
  • RAID, file systems, etc.

• Only thing to add: **volume management**
  • Concatenate multiple block devices together (including RAID devices)
  • Decouples resulting block device from a single RAID topology
  • Example: Linux Logical Volume Manager (LVM)
Example: Linux Volume Manager (LVM)

- Separate *physical* block device boundaries from *logical* block device boundaries.
  - **Aggregate** into *volume groups*
  - **Split** into *logical volumes*
Network virtualization
VLANs

- Logically separate network
- Switch ports can be:
  - **Access ports**: can only see one VLAN, aren’t aware of VLAN concept
  - **Trunk ports**: end point includes a VLAN tag in packet header to indicate which VLAN it wants to talk to; interprets such headers on incoming packets

VLANs and System Virtualization

- Virtual switches provide virtual access ports
- Hypervisor’s physical NICS are trunk ports for uplink

“Software Defined Networking” (SDN): Overused and abused buzzword

Just means “the network config is done in software”.

Often translates to “connect everything with fat cables, split up traffic and configure network in software”.

Examples:
- Open vSwitch (for KVM/Xen environments)
- Cisco Nexus 1000V (virtual vSwitch)
• When it comes to virtualization, open vSwitch is attractive because it provides the ability for a single controller to manage your virtual network across all your servers.

Adapted from "Virtualization Techniques" by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
Putting it all together
“FlexPod for VMware”: VMware on Cisco+NetApp

Flexible platform built from unified compute, fabric, and storage

Simplified procurement and operation of cloud infrastructure

Integrated management enabling centralized and co-ordinated operations

Validated architectures and deployment services

Open Management Framework integrates easily with 3rd party infrastructure management tools

http://community.netapp.com/t5/Tech-OnTap-Articles/Running-Microsoft-Apps-on-FlexPod-for-VMware/tap/84887
VM Migration: Hypervisor-to-Hypervisor

• Active state of a VM is moved from one hypervisor to another
  ▶ Copies the contents of virtual machine memory from the source hypervisor to the target

• This technique requires source and target hypervisor access to the same storage
VM Migration: Array-to-Array

- VM files are moved from source array to remote array
- Can move VMs across dissimilar storage arrays
- Balances storage utilization by redistributing VMs to different storage arrays
Common use case: Virtual Desktop Infrastructure (VDI)
Virtual Desktop Infrastructure (VDI)

- Virtual desktop Infrastructure (VDI) is a desktop-centric service that hosts users desktop environments on remote servers, which are accessed over a network using a remote display protocol.

Adapted from “Virtualization Techniques” by Dr. Yeh-Ching Chung, National Tsing Hua University, Taiwan.
• User’s physical machine is just a “thin client”; just shows remote desktop of VM
  • User does all work in VM
  • VM can be monitored and managed much easier than physical laptop

• Example: NetApp’s Virtual Engineering Desktop and “Dome” architecture for intellectual property security
  • Engineering VLAN separated from internet
Common use case: Multi-tenant environments

(Covered in more detail in the Cloud lecture)