



# Systems Administration

## Introduction to Virtualization



These materials are licensed under the Creative Commons *Attribution-Noncommercial 3.0 Unported* license  
(<http://creativecommons.org/licenses/by-nc/3.0/>)



# Objectives

- To revise the core concepts
- To ensure we are using the same terminology

# What is it?

- Virtualization -- the abstraction of the a resource from the actual physical instance of that resource.
- What Computing/Network resources can be virtualized?
  - Virtually anything! :)

# Anything?

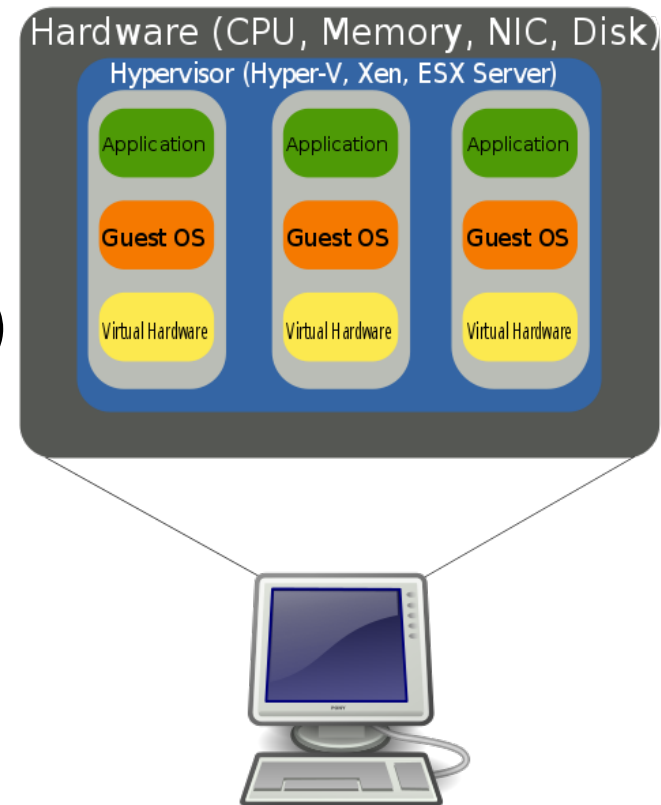
- In the context of this course. We're interested in virtualization along two dimensions:
  - Services
  - Hosts

# Resource/Service virtualization

- Examples:
  - Load-balancers
  - DNS Based GLB
  - HTTP(S) Virtual Hosting
  - MX records
  - Virtual Switches
  - Virtual Routers
  - Virtual Firewalls

# Host Virtualization

- Examples
  - Vmware
  - Virtual-Box (used in class)
  - KVM
  - XEN
  - FreeBSD and Linux Jails
  - Windows Hyper-V
  - Solaris Zones



# What problem are we attempting to solve with host virtualization.

- Problem 1 – Idle capacity.
  - Most of the machines in your datacenter are idle most of the time.
  - Capacity you're not using:
    - Cost money up front
    - Cost money to operate
    - Reduces your return on capital
  - Packing discreet systems into a smaller number of servers provides savings along virtually every dimension.

# Problems - Continued

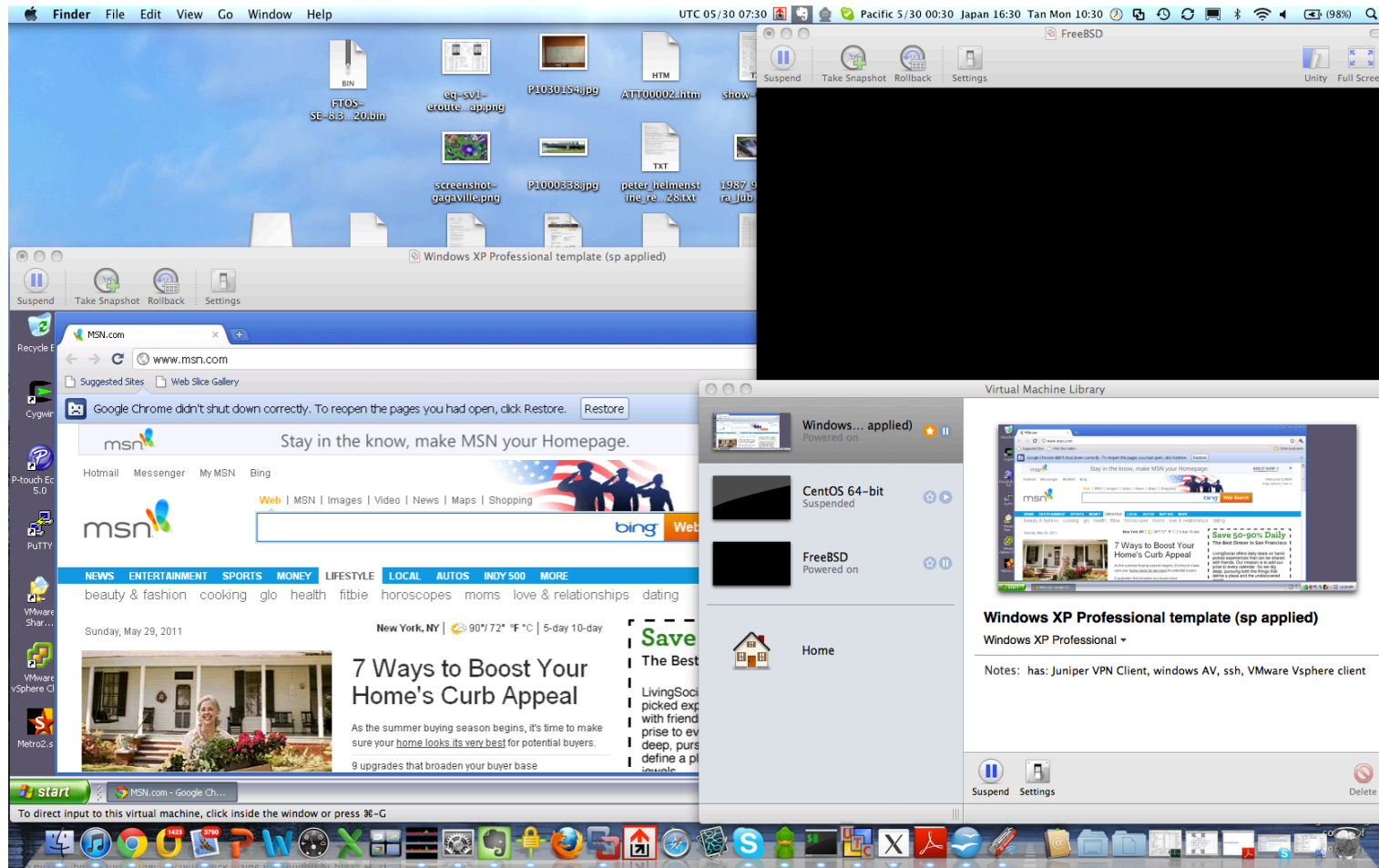
- Problem 2 – Provisioning
  - Spinning up a new service involves:
    - Acquiring the hardware
    - Building the server
    - Integration with existing services
  - With virtualization we're aiming to short-circuit that
    - Capacity is a resource
    - Machine instances may be cloned or provisioned from common basic images
    - Resources are purchased in bulk and assigned to applications as necessary



# Problems - Continued

- Problem 3 – Hardware abstraction
  - Operating systems, servers, and applications evolve at different rates.
  - Providing a common set of infrastructure resources means, virtualized systems are portable across servers
  - Hardware failure can more easily be managed.

# Examples – Desktop Virtualization



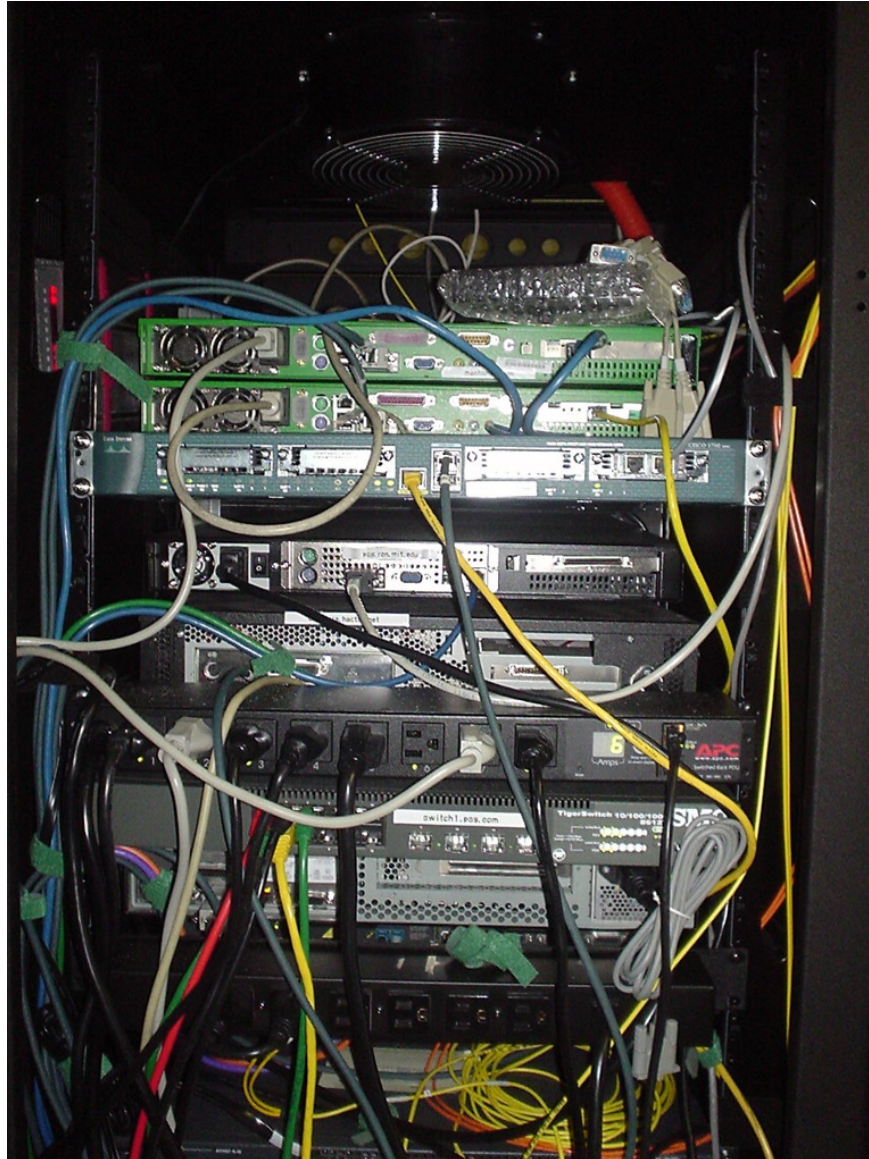
# Desktop Virtualization

- Uses
  - Prototyping services or applications before deployment
  - Utilities that don't run on your operating system
  - Isolation of sandbox environments from your desktop
  - Maintaining multiple versions of an environment for support purposes.
  - Staying familiar with unix while running windows (consider compared to the alternative (dual-booting))
- Issues
  - Emulating multiple computers on your laptop/desktop is somewhat resource intensive
- Vmware player and VirtualBox are free.
  - <http://www.virtualbox.org/wiki/Downloads>
  - [http://downloads.vmware.com/d/info/desktop\\_downloads/vmware\\_player/3\\_0?ie=UTF-8&q=vmwareplayer](http://downloads.vmware.com/d/info/desktop_downloads/vmware_player/3_0?ie=UTF-8&q=vmwareplayer)

# VirtualBox Extensions

- VMWare has similar “VMWare Guest Tools”
- These are extra drivers installed in the guest to support added functionality from within the VM
- VirtualBox Extension Pack adds:
  - USB 2.0 Support
  - RDP Support (remote desktop)
  - RDP Remote Media via local USB
  - PXE Boot Support

# Examples – Server Virtualization



# Server Virtualization - Continued

The screenshot displays the vSphere Client interface for a host named 'vm0.sea.rg.net VMware ESXi, 4.1.0, 260247'. The left sidebar shows a tree view of the inventory, including a folder 'vm0.sea.rg.net' and several virtual machines. The main pane shows the 'Virtual Machines' tab with a table of VMs and their resource allocation.

Name	State	Provisioned Space	Used Space	Host CPU - MHz	Host Mem - MB	Guest Mem - %	Notes
ssh.rpki.net	Powered On	104.00 GB	104.00 GB	39	4165	3	
turing.wps.com	Powered On	102.00 GB	102.00 GB	5	2105	0	
ran.psg.com	Powered On	154.00 GB	154.00 GB	39	4187	3	
cent0.psg.com	Powered On	102.00 GB	102.00 GB	51	1453	1	
apnicrpki.rand.apnic.net	Powered On	5.29 GB	5.29 GB	57	717	9	
rip1.psg.com	Powered On	38.00 GB	38.00 GB	41	2085	4	
hiroshima.bogus.com	Powered Off	254.00 GB	250.00 GB	0	0		
nagasaki.bogus.com	Powered On	1023.00 GB	439.14 GB	124	4257	9	

Below the table, the 'Recent Tasks' section is visible, showing a table with columns: Name, Target, Status, Details, Initiated by, Requested Start Time, Start Time, and Completed Time. The table is currently empty.

# Server Virtualization

The screenshot displays the vSphere Client interface for a host named 'vm0.sea.rg.net'. The interface is divided into several sections:

- Left Panel:** A tree view showing the host 'vm0.sea.rg.net' and its associated virtual machines: 'apnicrpk1.rand.apnic.net', 'cent0.psg.com', 'hiroshima.bogus.com', 'nagasaki.bogus.com', 'ran.psg.com', 'rip1.psg.com', 'ssh.rpk1.net', and 'turing.wps.com'.
- Top Bar:** A navigation bar with tabs for 'Getting Started', 'Summary', 'Virtual Machines', 'Resource Allocation', 'Performance', 'Configuration', 'Local Users & Groups', 'Events', and 'Permissions'. The 'Summary' tab is currently selected.
- General Section:** Displays host information including Manufacturer (Supermicro), Model (X7DWU), CPU Cores (8 CPUs x 2 GHz), Processor Type (Intel(R) Xeon(R) CPU E5405 @ 2.00GHz), License (vSphere 4 Essentials), Processor Sockets (2), Cores per Socket (4), Logical Processors (8), Hyperthreading (Inactive), Number of NICs (2), State (Connected), Virtual Machines and Templates (8), vMotion Enabled (N/A), VMware EVC Mode (N/A), Host Configured for FT (N/A), Active Tasks (N/A), Host Profile (N/A), and Profile Compliance (N/A).
- Resources Section:** Shows CPU usage (750 MHz) and Memory usage (20163.00 MB). It also includes a table for Datastore usage:

Datastore	Capacity	Free	Last Update
datastore1	3.63 TB	2.45 TB	4/29/2011

- Network Section:** Shows the VM Network connected to a Standard switch network.
- Fault Tolerance Section:** Displays Fault Tolerance Version (2.0.1-2.0.0-2.0.0) and a link to 'Refresh Virtual Machine Counts'. It also shows the number of Total Primary VMs (0), Powered On Primary VMs (0), Total Secondary VMs (0), and Powered On Secondary VMs (0).
- Recent Tasks Section:** A table with columns for Name, Target, Status, Details, Initiated by, Requested Start Time, Start Time, and Completed Time. The table is currently empty.



# Virtualized Servers as a Service (Amazon Web Services)

- Much as colocated servers, are available from a hosting provider, virtual servers are also available.
- Model is:
  - You pay for what you use.
  - Flexibility, need fewer servers today then you used, yesterday.
  - Leverage other amazon tools (storage/map-reduce/load-balancing/payments etc)



# AWS

The screenshot shows the AWS Management Console interface. The top navigation bar includes links to AWS, Products, Developers, Community, Support, and Account. The main content area is titled 'My Instances' and displays a table of instances. The table has columns for Name, Instance ID, AMI ID, Root Device, Type, Status, Security Groups, Key Pair Name, Monitoring, and Virtualization. Two instances are listed: one with ID i-7a255b3e (terminated) and one with ID i-ac1213e8 (running). Below the table, there are four monitoring graphs: Avg CPU Utilization (Percent), Avg Disk Reads (Bytes), Avg Disk Writes (Bytes), and Max Network In (Bytes). The console also features a navigation sidebar on the left with links to EC2 Dashboard, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORKING & SECURITY.

Name	Instance	AMI ID	Root Device	Type	Status	Security Groups	Key Pair Name	Monitoring	Virtualization
<input checked="" type="checkbox"/>	i-7a255b3e	ami-c7643482	ebs	t1.micro	terminated	default	joelja_amazon_key	basic	paravirtual
<input type="checkbox"/>	empty	i-ac1213e8	ebs	t1.micro	running	bare	amazon	basic	paravirtual

You have selected an instance that is not running. Amazon CloudWatch Detailed Monitoring can only be enabled for running instances.

**Avg CPU Utilization (Percent)**

Time	Utilization (%)
5/30 08:30	0
5/30 09:00	0

**Avg Disk Reads (Bytes)**

Time	Reads (Bytes)
5/30 08:30	0.0
5/30 09:00	0.0

**Avg Disk Writes (Bytes)**

Time	Writes (Bytes)
5/30 08:30	0.0
5/30 09:00	0.0

**Max Network In (Bytes)**

Time	Network In (Bytes)
5/30 08:30	0
5/30 09:00	0

# AWS Steps

- Select availability zone
- Launch new instance
- Select appropriate ami
- Associate with ssh key
- Launch instance
- Add ip
- SSH into new machine instance.
- t1-micro-instances run \$54 a year + bandwidth

# Try it for free...

- Free tier for the first Calendar year is (per month):
  - 750 hours of EC2 running Linux/Unix Micro instance usage
  - 750 hours of Elastic Load Balancing plus 15 GB data processing
  - 10 GB of Amazon Elastic Block Storage (EBS) plus 1 million IOs, 1 GB snapshot storage, 10,000 snapshot Get Requests and 1,000 snapshot Put Requests
  - 15 GB of bandwidth in and 15 GB of bandwidth out aggregated across all AWS services

# AWS - Continued

- For provisioning purposes cli interaction is possible:
  - <http://aws.amazon.com/developertools/351>
- Along with tools to support the provisioning and destruction of virtual machines.

# Provisioning and management

- Is the glue that makes virtualization usable
- In commercial virtualization environments the provisioning/management toolkits represent the bulk of the licensing cost (VMware) and the secret sauce (VMotion, disaster recovery, backup, etc)
- One end of the spectrum:
  - XEN tools – a collection of perl scripts for spinning <http://www.xen-tools.org/software/xen-tools/>
  - KVM tools - [http://www.linux-kvm.org/page/Management\\_Tools](http://www.linux-kvm.org/page/Management_Tools)
- The Other:
  - Rightscale - <http://www.rightscale.com/products/advantages/managing-systems-not-servers.php>

# Supporting Technology

- NIC teaming or Link aggregation
- Network attached storage and network centric filesystems
  - Example NFS
  - Hadoopfs
- Distributed databases
  - Example mysql cluster
  - OracleRAC

# QEMU-KVM

- **Qemu Emulator – the foundation of a number of virtualization products (including VirtualBox)**
- **Emulates the Entire Machine Environment**
  - **BIOS**
  - **CPU(s) SMP-capable**
  - **IDE Controller**
  - **NICs, many types**
  - **Graphics**
  - **USB, Sound, Etc.**
- **qemu-img used to generate Virtual Disks**
  - **supports RAW disks, sparse disks, copy-on-write, and VMDK**

# QEMU-KVM

- Why Qemu and not VMWare
  - 1) free open-source software
  - 2) supported by Redhat
  - 3) lots of features
  - 4) lots of support tools in development
- Why **NOT** QEMU-KVM
  - 1) documentation can be missing
  - 2) some features are buggy



# KVM

<http://www.linux-kvm.org>

- **KVM (Kernel Virtual Machine)**
- **Kernel modules for accelerating Virtualization**
- **Also provides additional services and I/O functionality**
- **Fully Integrated with current QEMU-KVM Distributions**
- **CPU-Specific, i.e. “kvm\_intel” or “kvm\_amd”**
- **% cat /proc/cpu | egrep 'vmx|svm'**
- **NOTE: Intel VM Extensions may or may not be enabled in your BIOS by default. Check this before you install a Hypervisor.**

# QEMU-KVM

- Can run in a number of display modes:
  - “SDL” VGA Graphics
  - curses” text graphics
  - VNC remote viewing
- Many Network NIC options
  - default is an internal DHCP with NO ICMP support
  - bridged mode support by “virtio” and “tap” interfaces

# QEMU Examples

```
qemu -hda /vms/myimg -cdrom /isos/ub10.iso -m 512
```

```
qemu -hda img1 -hdb img2 -hdc img3
```

```
qemu -hda qemu linux.img \
```

```
-net nic,vlan=0 -net tap,vlan=0,ifname=tap0
```

```
qemu -hda img.qcow2 -m 512 -daemonize -vnc :5
```

NOTE: the cdrom device can be an ISO file within the Host filesystem, or the Host CDROM drive itself

# qemu-img

- qemu-img is the tool used to generate qemu virtual disks
- qcow2 format
  - sparse disk storage
  - copy-on-write (c.o.w.), a.k.a. “snap-shot” support
  - copy-on-write: means freezing a disk image, and using a new file to hold any further writes to that disk. In this way the original disk image is preserved. To roll-back, throw away new file.
- Capable of reading/converting VirtualBox and VMWare Disks
- Examples:  
`qemu-img create myhd.qcow2 6G`  
`qemu-img convert old.vmdk -O qcow2 newimg`

# The Qemu Monitor

<http://en.wikibooks.org/wiki/QEMU/Monitor>

- builtin control console used to jump out of the guest OS and perform operations on the VM
- access with CTRL-ALT-1/CTRL-ALT-2

(Mac uses CTRL-OPT-1/CTRL-OPT-2)

- operations: stop, cont, system\_powerdown, change, usb\_add, vnc, etc.
- migration: live migration from one site to another  
On site B: % qemu -hda myimg -incoming tcp:0:4444  
On site A: (in monitor) migrate -b tcp:hostB:4444

# virsh/virt-manager

- libvirt toolkit – API used to interface with the qemu-kvm (and other vm platforms, xen, etc.)
- provides a uniform interface for controlling VMs
- provides a more consistent management console
- requires user added to groups: kvm, libvirtd
- Examples:
  - root/system-level: `virsh -c qemu:///system`
  - user/sessions: `virsh -c qemu:///session`
  - `virsh# list -all`
- virt-manager: GUI tool for building and controlling VMs

# Virtualization - Issues

- “All your eggs in one basket” – a poorly implemented virtual environment can create a large single point of failure
- Virtualization does not magically manufacture additional resources
- High-performance often requires dedicated hardware, ex. 10GB networking, massive Database I/O systems
- Sometimes the virtualized environment does not have all the features of the real one

# Virtualization - Summary

- Useful for creating and testing new OS's
- Excellent for creating a dual-head, fully redundant, highly-available set of services with live-migration for failover
- Considerable savings on physical resources: heating, cooling, rack space, etc.
- Copy-On-Write filesystems and Snapshots are useful as for de-duplication and as point-in-time versions of the OS
- Significantly reduces deployment time
- Provides a standard environment for services