College of Computer Science and Technology
Zhejiang University, P. R. China

Virtualization Overview

Jianhai Chen (陈建海)

chenjh919@zju.edu.cn

+86 13958011808

Intelligence Computing and System Lab

College of Computer Science

Zhejiang University

Hangzhou, Zhejiang, P.R. China

2015/09/07

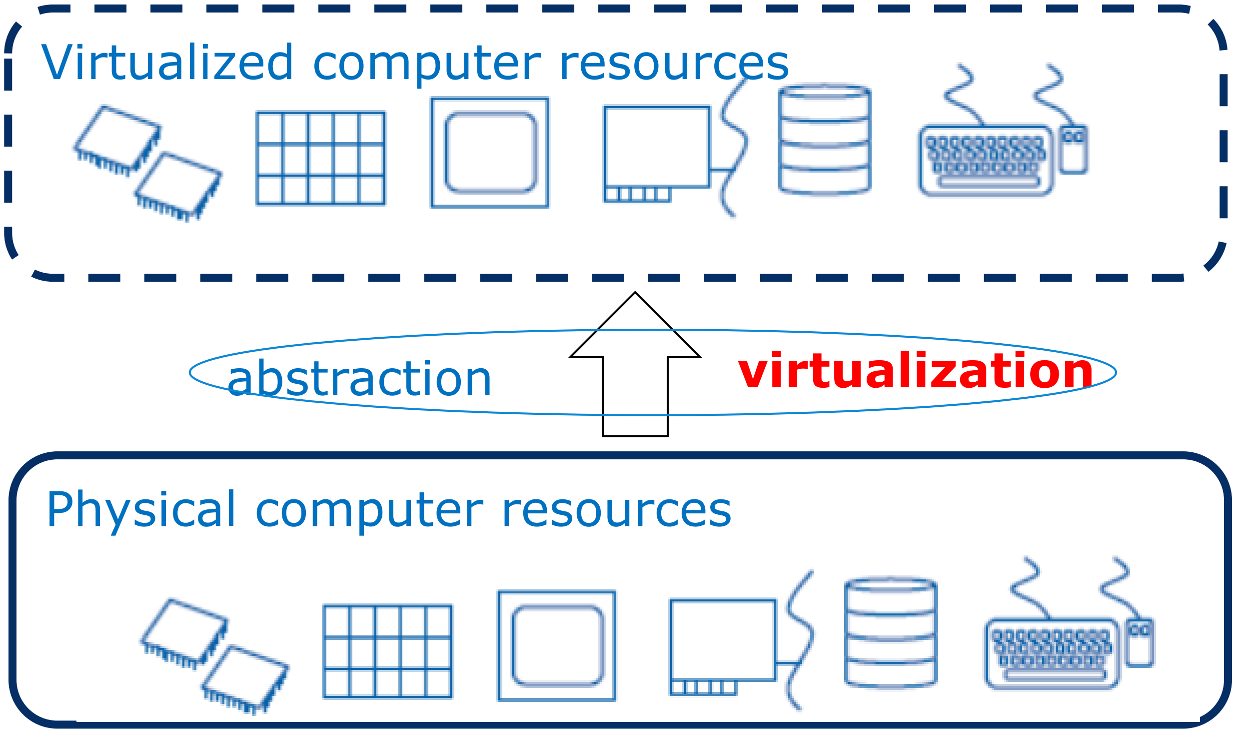
IHEP SUMMER SHOOL

Agenda

- Introduction
- Virtualization Technologies
  - CPU
  - Memory
  - IO
- Commercial virtualization tools
- Problems in using virtualization
  - How to create and handle VM images
  - Virtual network concepts and configuration
- Summary

What is virtualization

- Virtualization is a term that (**wikipedia**)
  - refers to the **abstraction** of computer resources, or **the act of creating a virtual (rather than actual) version of something**, including virtual computer hardware platforms, operating systems, storage devices, and computer network resources.



https://en.wikipedia.org/wiki/Virtualization

History

- In the 1960s
  - Firstly appeared in IBM mainframe computer OS.
  - Used as a method of dividing system resources

V · T · E		Virtualization
Application-level (Sandbox)		Ceedo · Citrix XenApp · Dalvik · InstallFree · Microsoft App-V · Spoon · Symantec Workspace Virtualization · VMware ThinApp · ZeroVM
Environment-level Containers		cgroups-based (CoreOS · lmctfy · LXC · Docker · OpenVZ (Virtuozzo)) · Linux-VServer · FreeBSD jail · iCore Virtual Accounts · Solaris Containers · Workload Partitions
OS-level Hypervisors	Microkernel	Hyper-V · LynxSecure · Oracle VM Server for SPARC · VMware ESX/ESXi · Adeos · Xen · XtratuM · z/VM
	Monolithic	bhyve · KVM · L4Linux · Mac-on-Linux · Mac-on-Mac · Microsoft Virtual Server · Parallels Workstation · Parallels Desktop for Mac · Parallels Server for Mac · PearPC · QEMU · VirtualBox · Vi… · Virtual Machine Manager · VMware Fusion · VMware Player · VMware Server · VMware Workst… · Windows Virtual PC · Win4Lin
Networking		DOVE · Open vSwitch · Virtual security switch · VXLAN

more academia research on virtualization.

- 2007, KVM, a linux OS kernel-based VM
- 2007, hardware support: Intel VT, AMD-V
- Now ~50 virtualization tools or types

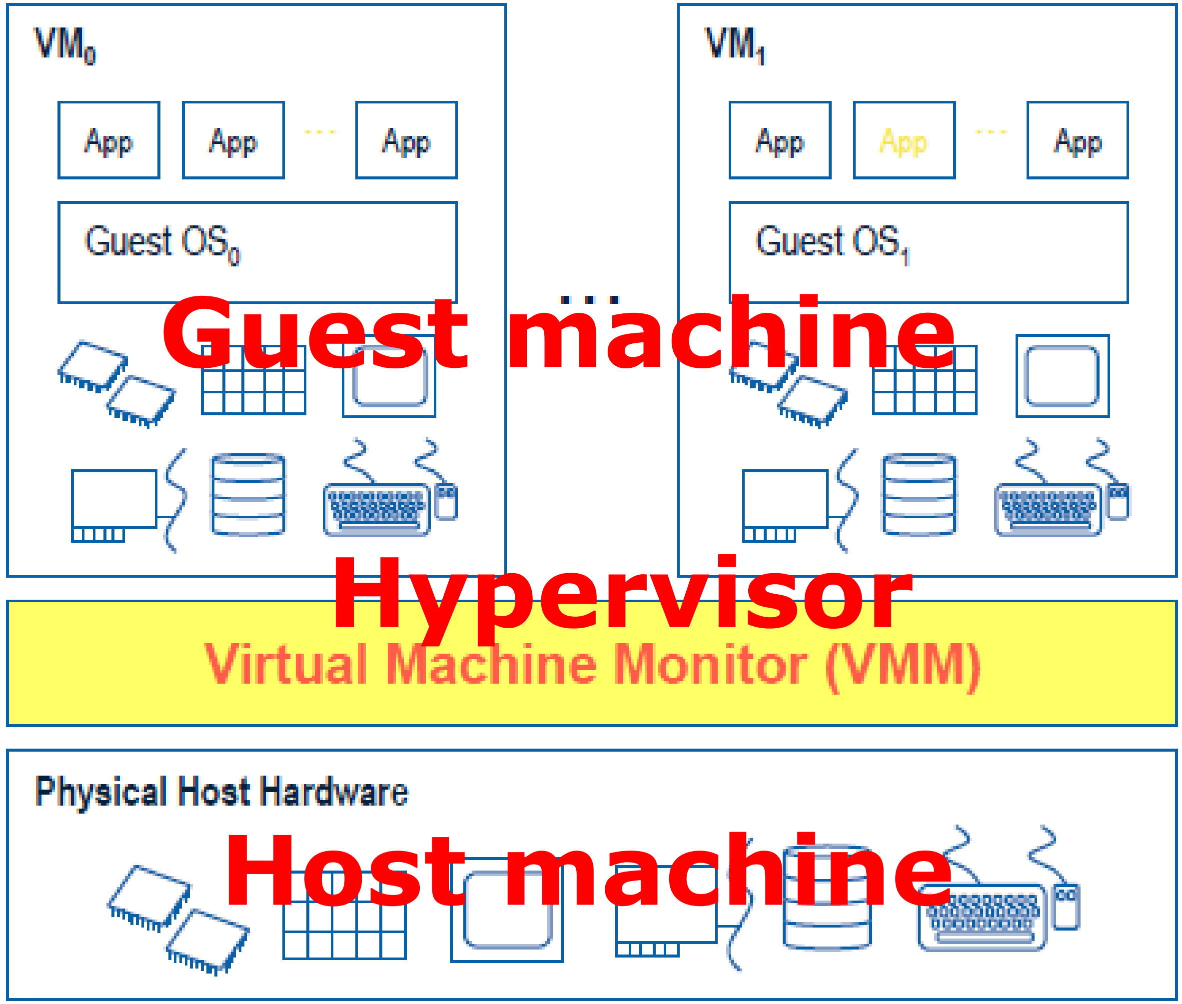
Virtualization technology types

- Hardware virtualization
- Software virtualization
- Other virtualization
  - Desktop virtualization
  - Nested virtualization

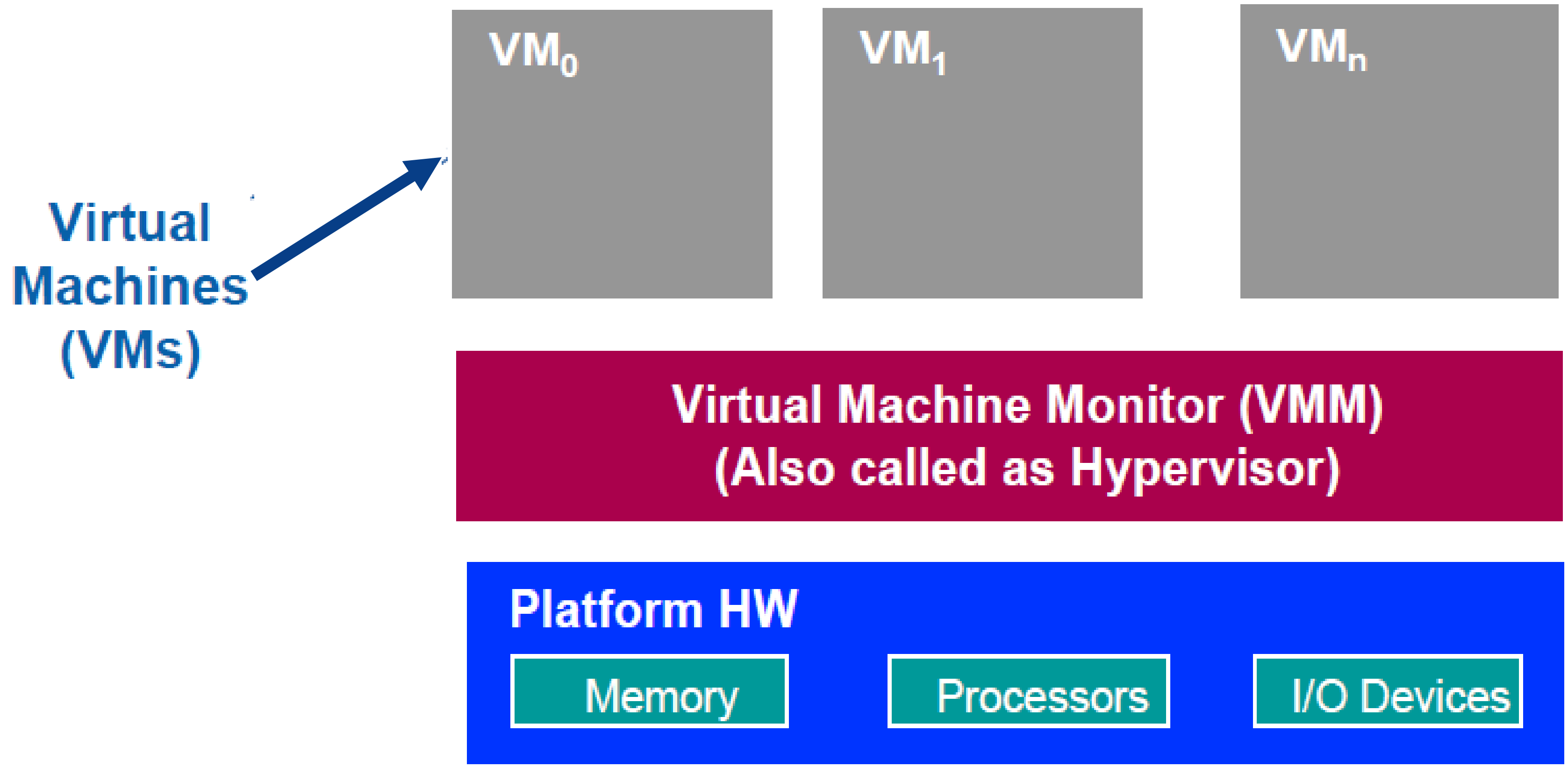
Hardware virtualization (HV)

- Hardware virtualization (most fundamental)
  - or platform virt… machine (VM) t… operating syste…
  - Software execu… underlying hard…
    - Host with Mic…

- Concepts
  - Host/guest
  - Hypervisor or *Virtual Machine Manager (VMM)*



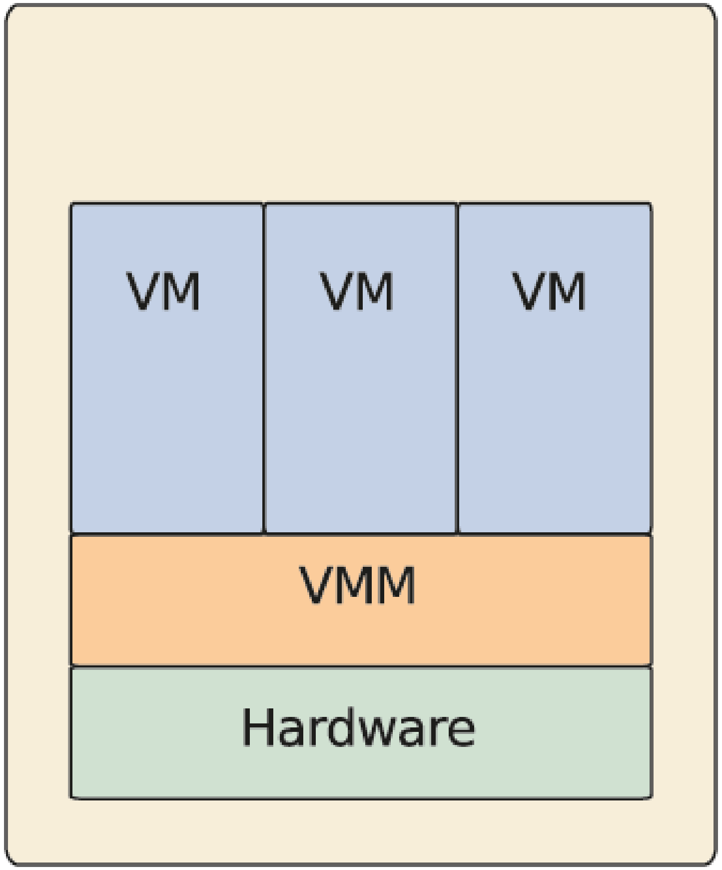
Virtual machine monitor (VMM)



VMM transforms the single machine interface into the illusion of many interfaces. Each of these interfaces (virtual machines) is an efficient replica of the original computer system, complete with all of the processor instructions.

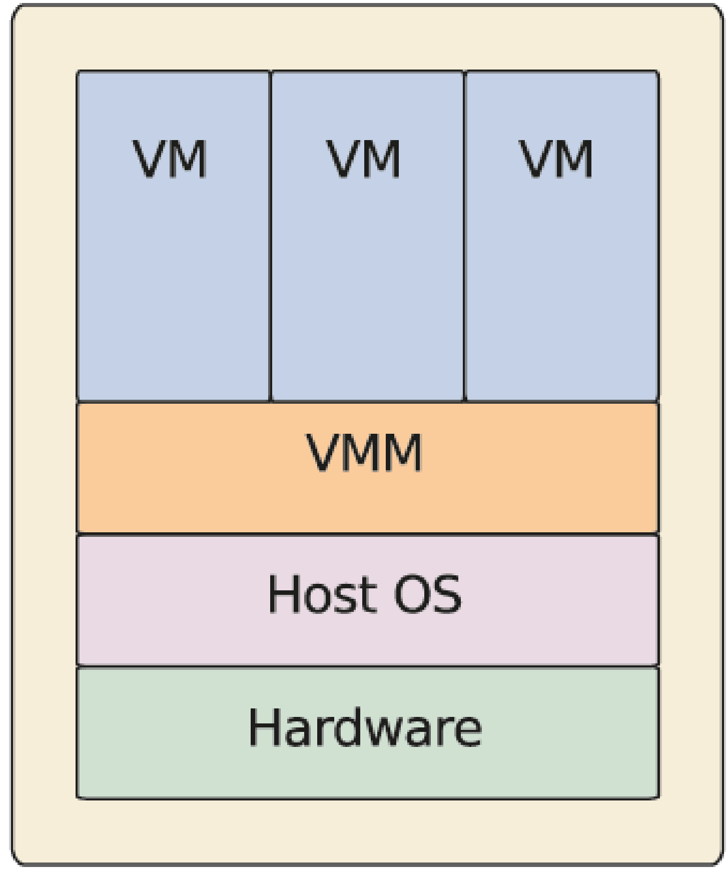
Robert P. Goldberg.

Two Hypervisor (VMM) Types



Hypervisor Type I

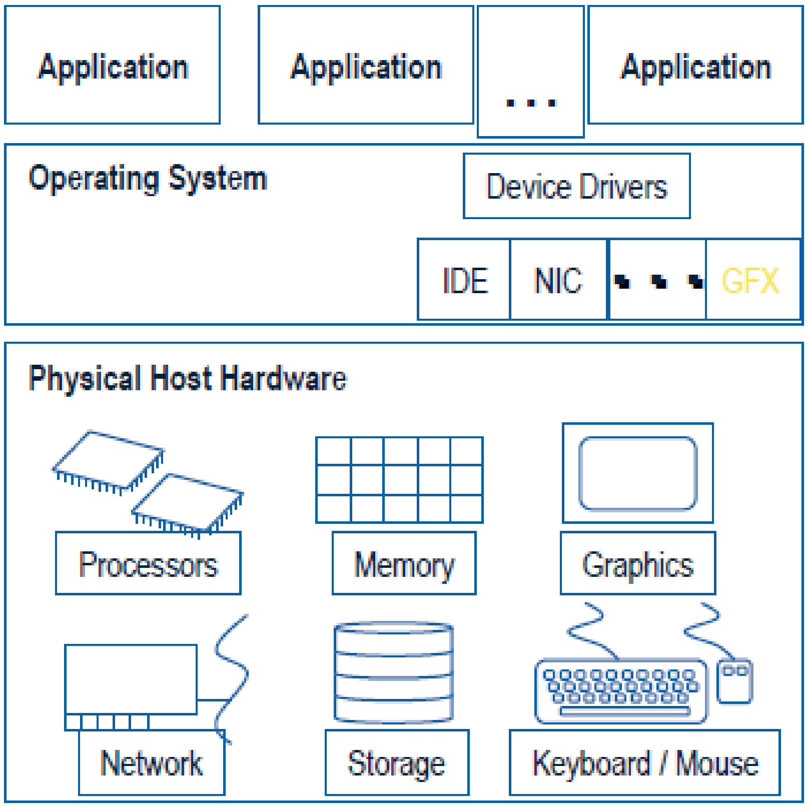
Vmware Esx server
Xen ctrix server



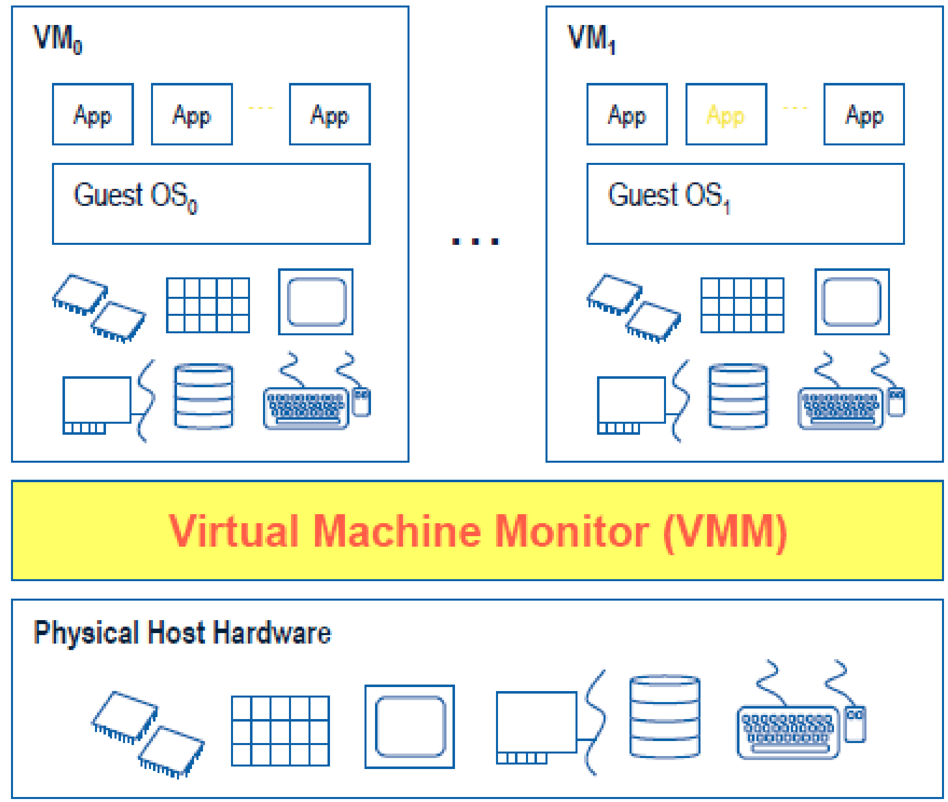
Hypervisor Type II

VMware Esx workstation
Xenserver

What does a VM look like



Without VMs: Single OS owns all hardware resources



With VMs: Multi-OSes share hardware resources

A virtual machine is implemented by adding software to an execution platform to give it the appearance of a different platform, or for that matter, to give the appearance of multiple platforms.

Essential characteristics of VMM

- **Equivalence**
  - Essentially identical virtual platform, except
    - Differences caused by the availability of system resources
      - E.g. amount of memory available to VM
    - Differences caused by timing dependencies.
- **Isolation, or resource control**
  - VMM is in complete control of system resources
    - VM can't access any resources not explicitly allocated to it
    - VMM may regain control of resources already allocated
- **Efficiency**
  - At worst only minor decreases in speed
  - Rules out traditional emulators and complete software interpreters (simulators)

Source from Gerald J. Popek & Robert P. Goldberg, "Formal Requirements for Virtualizable Third Generation Architectures"

Types of hardware virtualization

- Full virtualization
  - almost complete simulation of the actual hardware to allow software to run an unmodified guest OS.
  - Hardware-assisted virtualization
  - Binary Translation etc.
- Partial virtualization
  - some but not all of the target environment attributes are simulated.
    - As a result, some guest programs may need modifications to run in such virtual environments.
- Paravirtualization
  - Modify guest OS to cooperatively work with hypervisor to solve virtualization issues or for performance.
- Hardware-assisted virtualization
  - is a way of improving overall efficiency of virtualization.
    - CPUs provide support for virtualization in hardware
    - other hardware components that help improve the performance of a guest environment.

Software virtualization

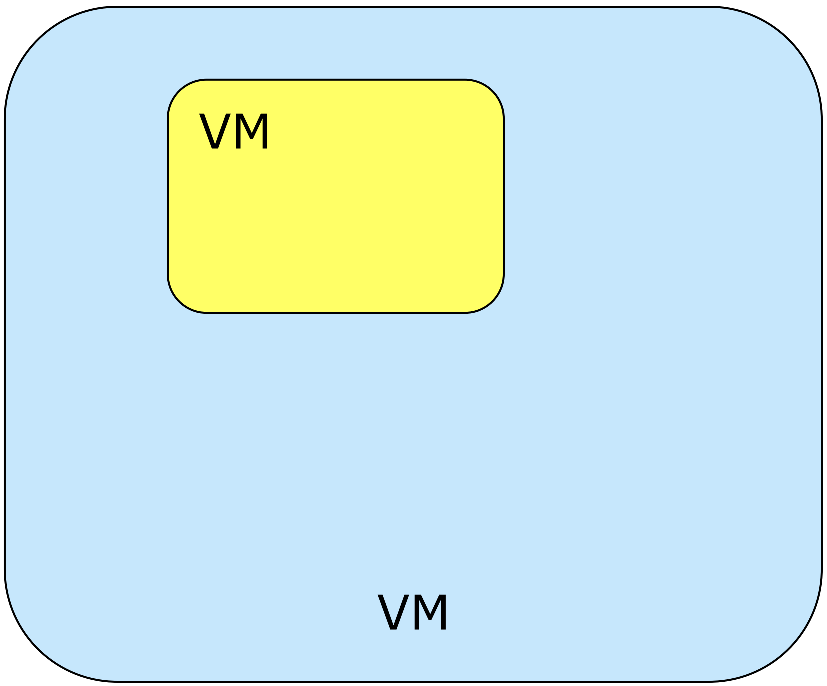
- **Operating system-level virtualization**
  - hosting of multiple virtualized environments within a single OS instance.
  - An OS allows for multiple isolated user-space instances (based on cgroup)
  - Often called containers
- **Application virtualization and workspace virtualization**
  - the hosting of individual applications in an environment separated from the underlying OS.
  - Application virtualization is closely associated with the concept of portable applications.
- **Service virtualization**
  - emulating the behavior of dependent (e.g., third-party, evolving, or not implemented) system components that are needed to exercise an application under test (AUT) for development or testing purposes.

Desktop virtualization

- Desktop virtualization
  - is the concept of separating the logical desktop from the physical machine.
    - One form of DV is virtual desktop infrastructure (VDI).
    - it can be thought of as a more advanced form of hardware virtualization.
  - Is used to support desktop office.

Nested virtualization

- Nested virtualization
  - refers to the ability of running a virtual machine within another, having this general concept extendable to an arbitrary depth.
  - Can be used to deploy a cloud platform in VMs.



Agenda

- Introduction
- Virtualization Technologies
  - CPU
  - Memory
  - IO
- Commercial virtualization tools
- Problems in using virtualization
  - How to create and handle VM images
  - Virtual network concepts and configuration
- Summary

Virtual Platform

- Simulates a hardware platform and all its software platforms.
- Hardware platform
  - CPU, Memory,I/O
- Software platforms/components layer
  - BIOS
  - OS
  - Run time library
  - Applications

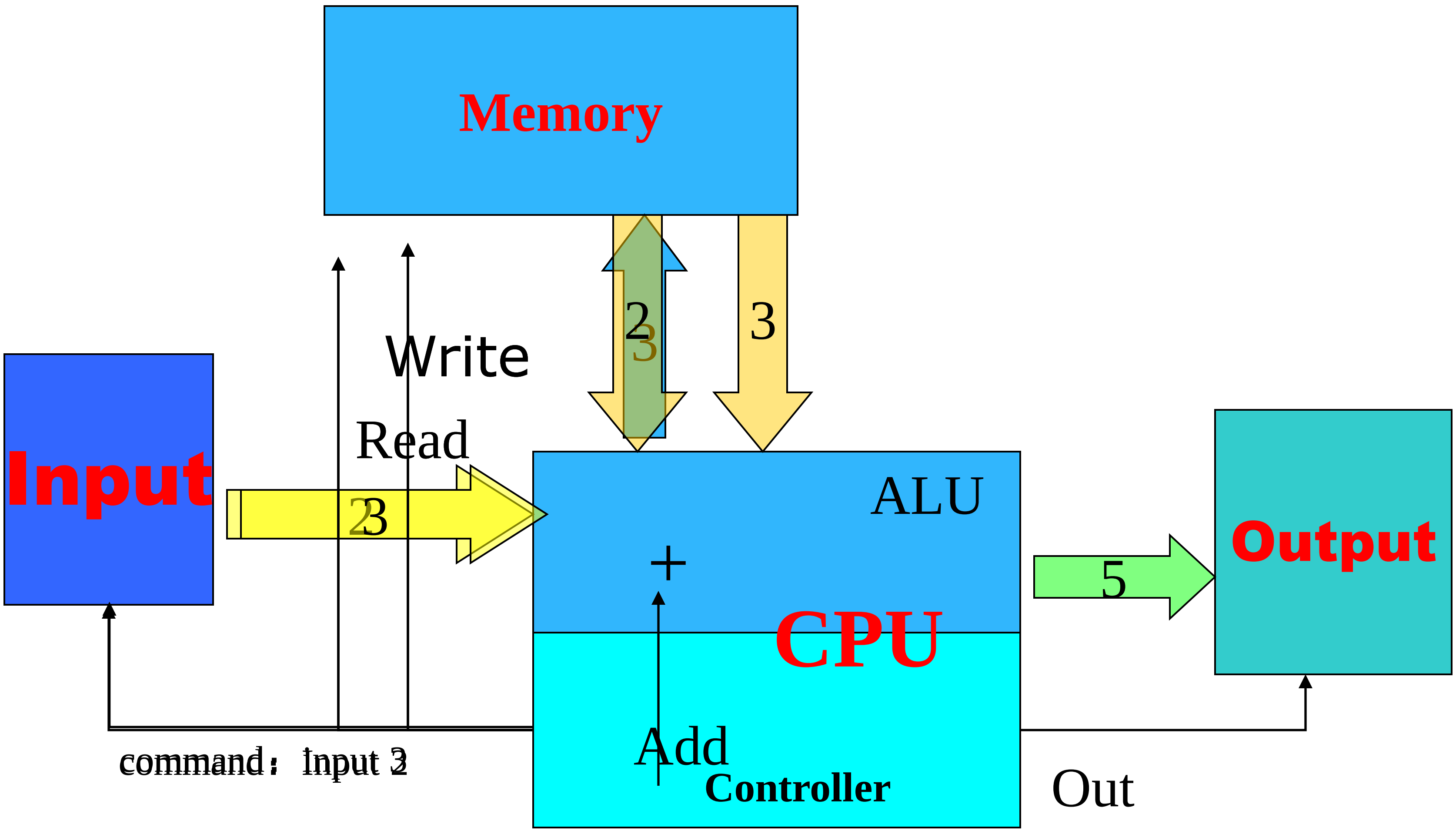
Agenda

- Introduction
- Virtualization Technologies
  - CPU
  - Memory
  - IO
- Commercial virtualization tools
- Problems in using virtualization
  - How to create and handle VM images
  - Virtual network concepts and configuration
- Summary

Instruction and application program

- CPU is responsible for running applications.
- CPU includes an instruction set system.
- An instruction
  - is a basic command of computer CPU operation.
- An application program
  - is compiled into a binary code initially.
  - is a set of sequence instructions.
- The running of application denotes
  - CPU run a set of instructions according to a special order.

For Example：2+3=5

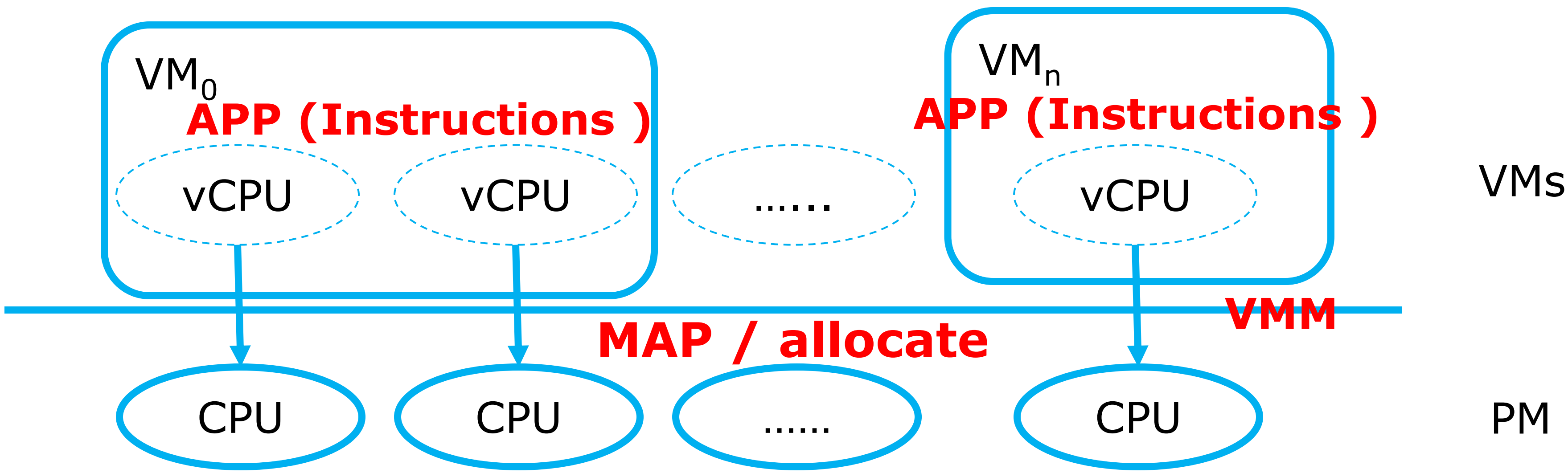


Program with instructions

- Input 2
- Write 2 to memory
- Input 3
- Write 3 to memory
- Read 2
- Read 3
- Add 2 and 3
- Output 5

CPU virtualization

- How to virtualize CPU resource?
  - VM is set one or more vCPUs and each vCPU is allocated to run applications of Guest OS.
  - Each application program is denoted by an ordered set of **instructions** and finally executed by physical CPU.
  - During the running of application, app instructions will firstly submitted to vCPU, then the vCPUs are mapped to or allocated with physical CPU to run instructions through **VMM**.

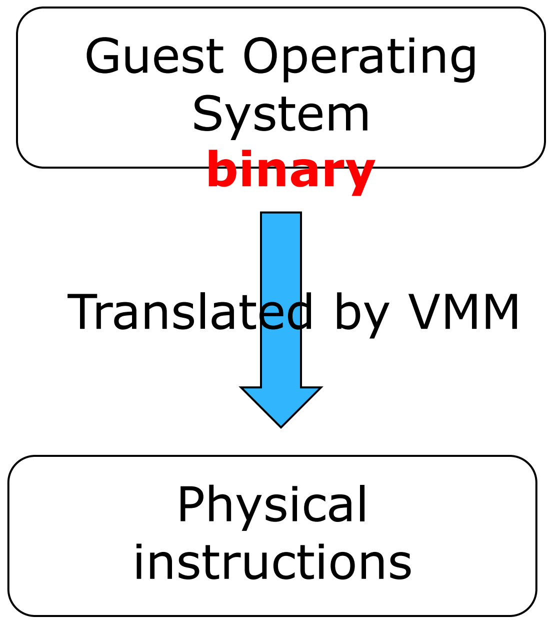


CPU virtualization approaches

- **Binary Translation (BT)**
  - Translate guest OS binary on the fly to solve virtualization issue
- **Paravirtualization (PV)**
- **Hardware-assisted virtualization**

1)Binary Translation (BT)

- Translate guest OS binary
  - OS binary is still visible to guest, but the executed guest code is actually in translation cache.
  - Need to steal guest address space to hold the translated cache.
    - be able to run **unmodified guest OS**
- All the instructions are virtualized for translation.
  - Frequent translations impact on performance
  - Benefit: not modify guest OS



2)Paravirtualization (PV)

- Modify guest OS source to cooperatively work with hypervisor for performance, simplicity etc.
  - Hypercall to request for hypervisor service
    - Share the global resource
  - Event channel to receive asynchronous notification from hypervisor
  - Share memory for massive information communication
  - Virtio for directly access IO device
- Widely used in ***device driver*** by commercial VMMs
  - Vmware
  - Xen/KVM
  - Hyper-V

3)Hardware-assisted virtualization

- Hardware extension
  - Intel® Virtualization Technology (Intel® VT)
  - AMD-V
- Hardware support to enable run
  - Guest OS runs in de-privileged mode to execute instructions
  - Guest access to privileged resources triggers **exit** from VM to VMM
- Be able to *run unmodified guest OS*

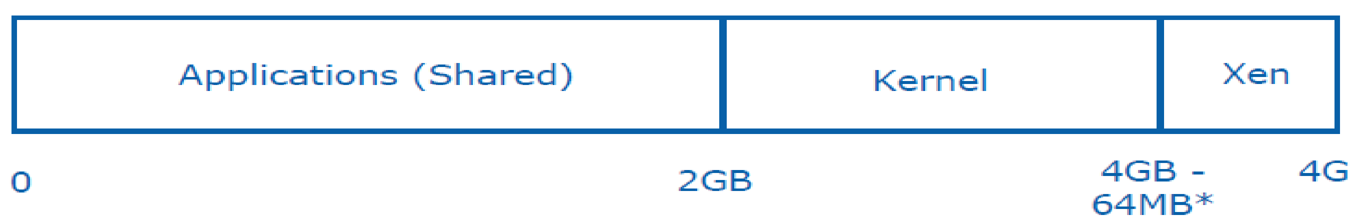
Agenda

- Introduction
- Virtualization Technologies
  - CPU
  - Memory
  - IO
- Commercial virtualization tools
- Problems in using virtualization
  - How to create and handle VM images
  - Virtual network concepts and configuration
- Summary

Memory virtualization

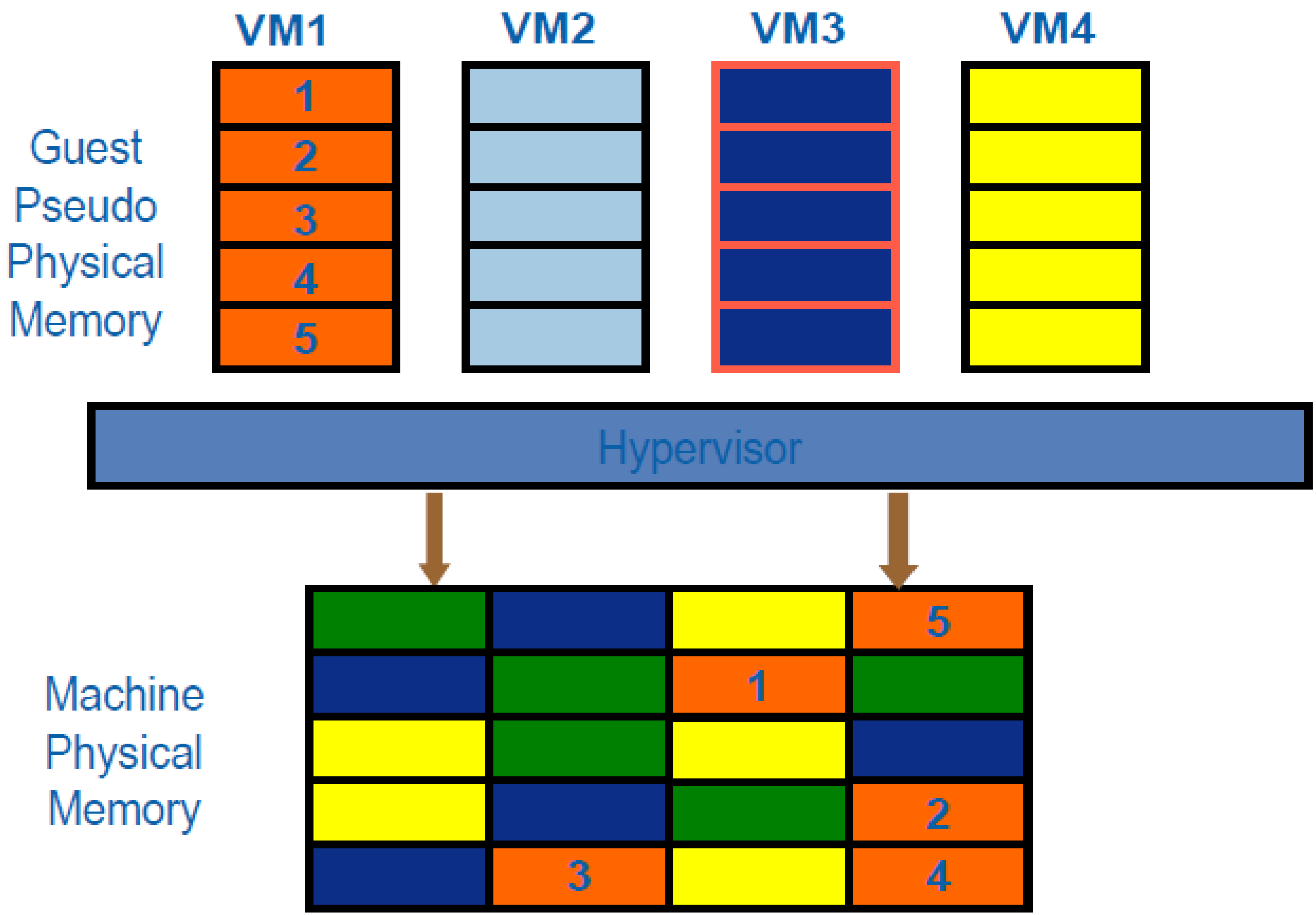
- In OS, memory management has the rules
  - OS expect to see physical memory starting from 0
    - BIOS/Legacy OS are designed to boot from address low 1M
  - OS expect to see contiguous memory in address space
    - OS kernel requires minimal contiguous low memory
      - OS, such as Linux, requires minimal contiguous low memory
    - OS components may require certain level of contiguous memory
      - For example, DMA pages, identical mapped kernel data structure…
    - Efficient page frame management
      - Less management overhead
    - Efficient TLB (Translation lookaside buffer)
      - Super page TLB

- **Example of XenLinux32 address space**



VMM has to remap guest physical address to host (or machine) physical address

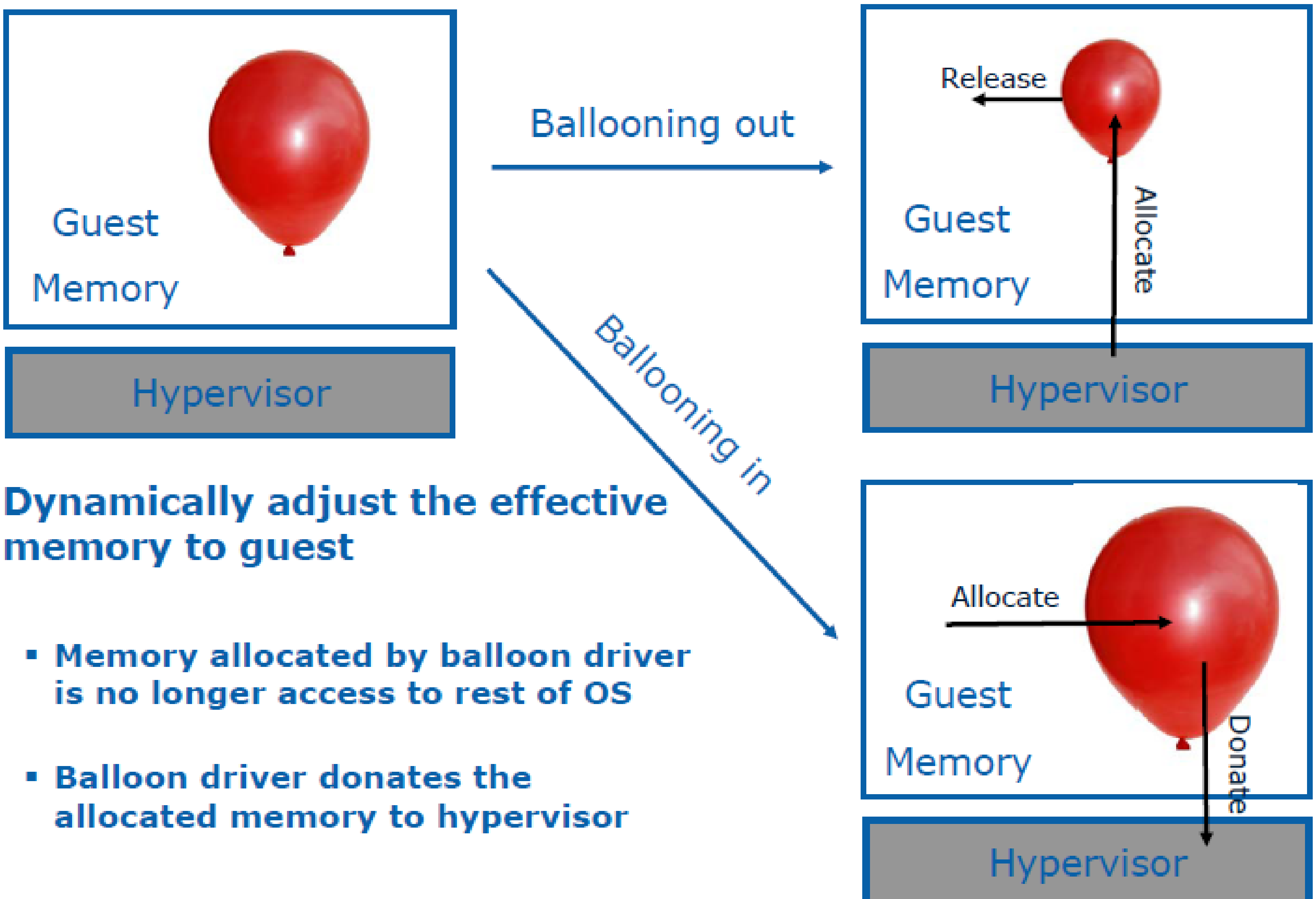
Physical page frame redirection



Page frame allocation

- Partitioning
  - Simple and high performance
  - $ to buy more memory
- Contents based sharing
  - Pages with same contents can share the physical frame
    - Such as zero page, guest code pages etc.
    - Can improve memory usage
- Ballooning
  - A VMM-aware balloon driver running in guest OS to dynamically allocate memory from OS and release them to VMM, and vice versa
- Host swapping
  - The physical frames for guest pages may be swapped out
- Shadow page table
  - Guest memory <--→ physical machine memory

Ballooning



- **Dynamically adjust the effective memory to guest**
  - **Memory allocated by balloon driver is no longer access to rest of OS**
  - **Balloon driver donates the allocated memory to hypervisor**

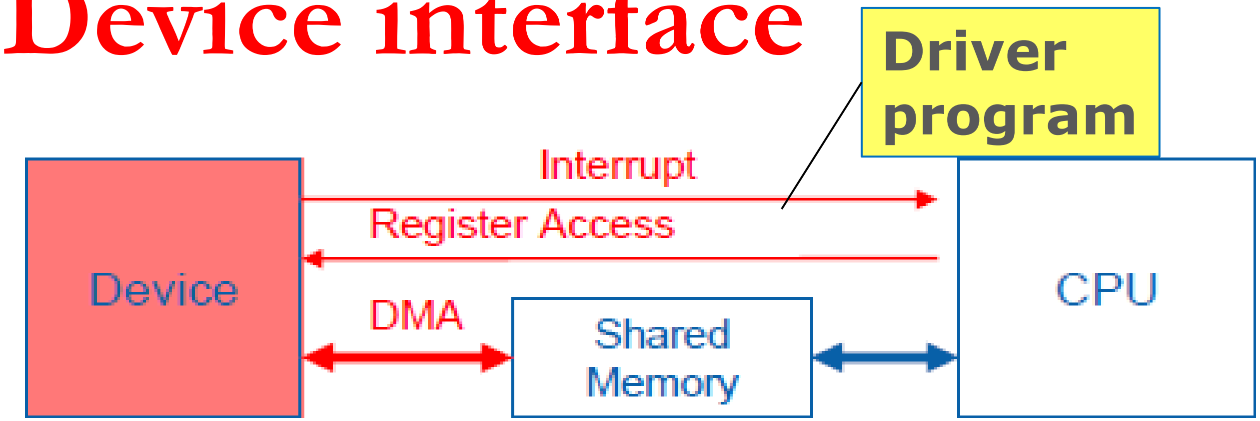
Agenda

- Introduction
- Virtualization Technologies
  - CPU
  - Memory
  - IO
- Commercial virtualization tools
- Two problems in using virtualization
  - How to create and handle VM images
  - Virtual network concepts and configuration
- Summary

IO virtualization

- Device interface
- Software Emulation
- Paravirtualization
- Direct assignment

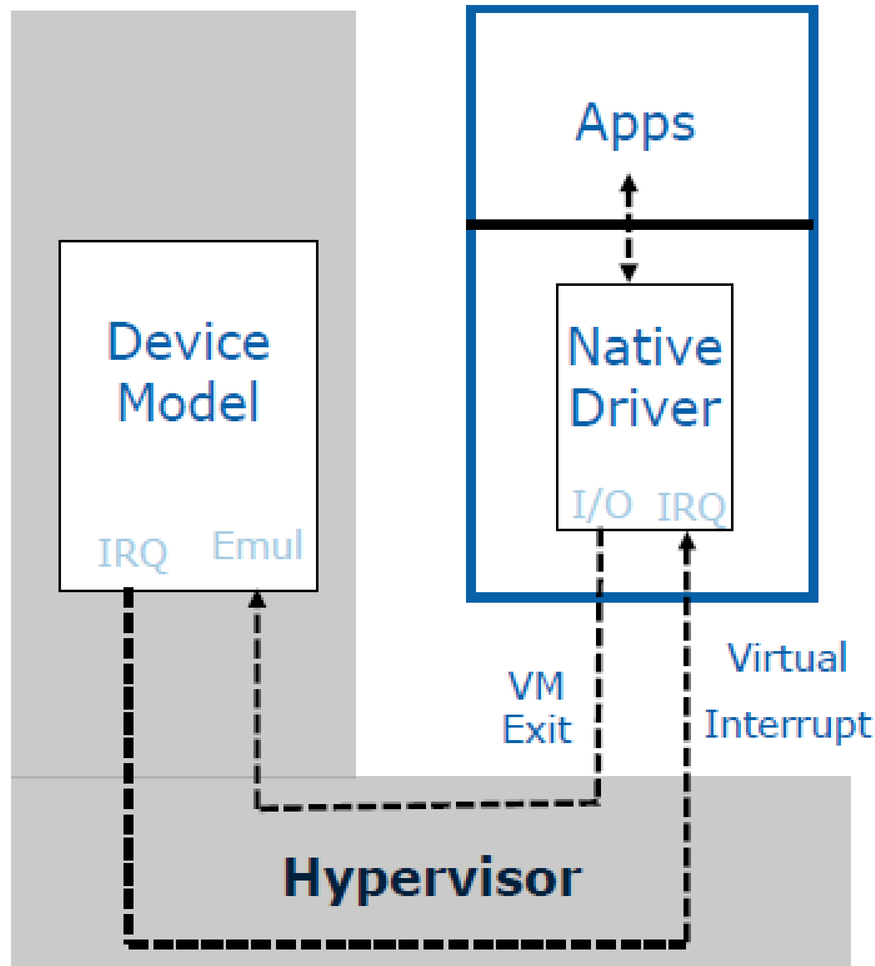
Device interface



- **Interaction between device and driver:**
  - Driver programs device through register access
  - Device notifies driver through interrupt
  - Device could DMA for massive data movement
- ***I/O Virtualization*** **requires the hypervisor to present guest a complete device interface**
  - Presenting an existing interface
    - **Software Emulation**
    - **Direct assignment**
  - Presenting a brand new interface
    - **Paravirtualization**

Software Emulation

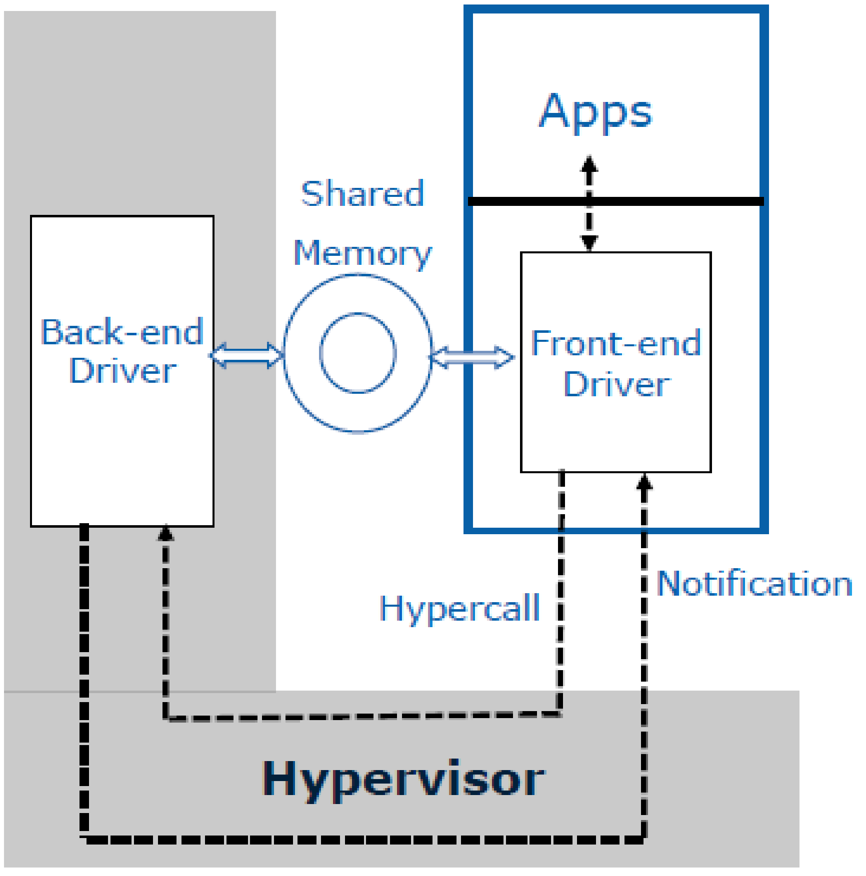
- **Guest runs native device driver, e.g. NE2000**
  - I/O access is trap-and-emulated by device model in hypervisor
    - Translation for MMIO is zapped
  - Virtual Interrupt is signaled by device model per se'mantics
- **Excessive trap and emulation**



Excessive hypervisor intervention for performance data movement

Paravirtualization

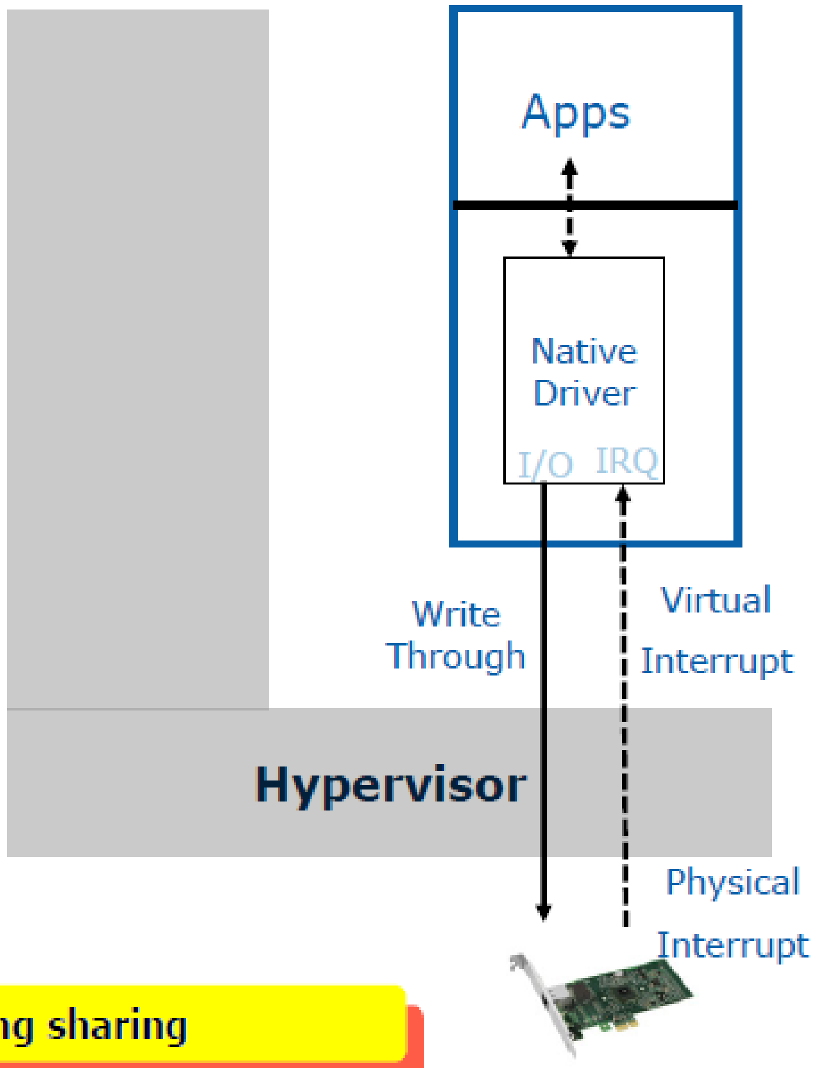
- **A new front-end driver (FE driver) is run in guest**
  - Optimized request through hypercall
- **Hypervisor runs a back-end driver (BE driver) to service request from FE driver**
  - Notify guest for processing
- **Shared memory is used for massive data communication**
  - To reduce guest/hypervisor boundary crossing
  - E.g. Xen VNIF, KVM Virtio-Net



Runtime Hypervisor intervention is largely reduced

Direct assignment

- Guest runs native driver
- I/O is written through
  - Translation for MMIO is presented
- Interrupt
  - Physical interrupt is captured by hypervisor (pIRQ)
  - Virtual interrupt is signaled for guest (vIRQ)
  - Remapping from pIRQ->vIRQin hypervisor



Maximizing performance, but sacrificing sharing

Agenda

- Introduction
- Virtualization Technologies
  - CPU
  - Memory
  - IO
- Commercial virtualization tools
- Problems in using virtualization
  - How to create and handle VM images
  - Virtual network concepts and configuration
- Summary

Commercial virtualization tools

- Xen
- KVM/Qemu
- Vmware
- virtualbox
- LXC & Docker
- ……

Xen

- ***Produced in 2003.***
- ***Open-source virtualization tool***
- ***The University of Cambridge*** Computer Laboratory developed the first versions of Xen
- Xen is currently available for the IA-32, x86-64 and ARM instruction sets.
- Xen supports five different approaches to running the guest operating system:
  - HVM (hardware virtual machine),
  - HVM with PV drivers,
  - PVHVM (HVM with PVHVM drivers),
  - PVH (PV in an HVM container) and PV (paravirtualization)

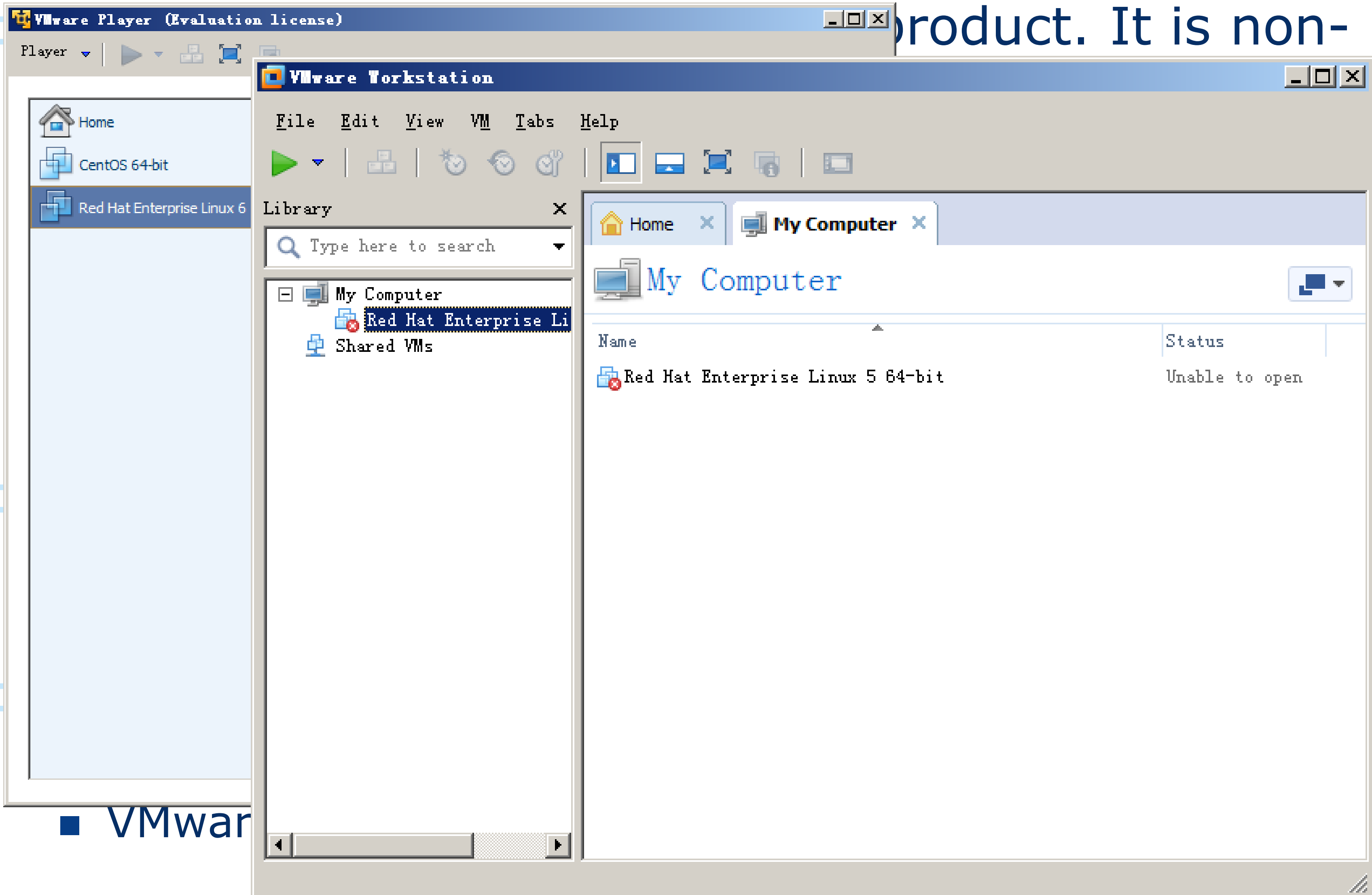
KVM

- Produced in 2007.
- KVM (Kernel-based Virtual Machine)
  - is a virtualization infrastructure for the Linux kernel that turns it into a hypervisor.
  - supports x86 processors
- Paravirtualization support for certain devices
  - is available for Linux, OpenBSD, FreeBSD, NetBSD, Plan 9 and Windows guests using the Virt-IO API.
  - This supports a para virtual Ethernet card, a para virtual disk I/O controller, a balloon device for adjusting guest memory usage, and a VGA graphics interface using SPICE or VMware drivers.

Qemu

- QEMU (short for Quick Emulator)
  - is a free and **open-source hosted hypervisor** that performs hardware virtualization (not to be confused with hardware-assisted virtualization).
- QEMU is a hosted virtual machine monitor
  - It emulates CPUs through dynamic binary translation and provides a set of device models, enabling it to run a variety of unmodified guest operating systems.
  - It also can be used together with **KVM** in order to run virtual machines at near-native speed (requiring hardware virtualization extensions (VT-x) on x86 machines).
  - QEMU can also be used purely for CPU emulation for user-level processes, allowing applications compiled for one architecture to be run on another.

VMware



roduct. It is non-

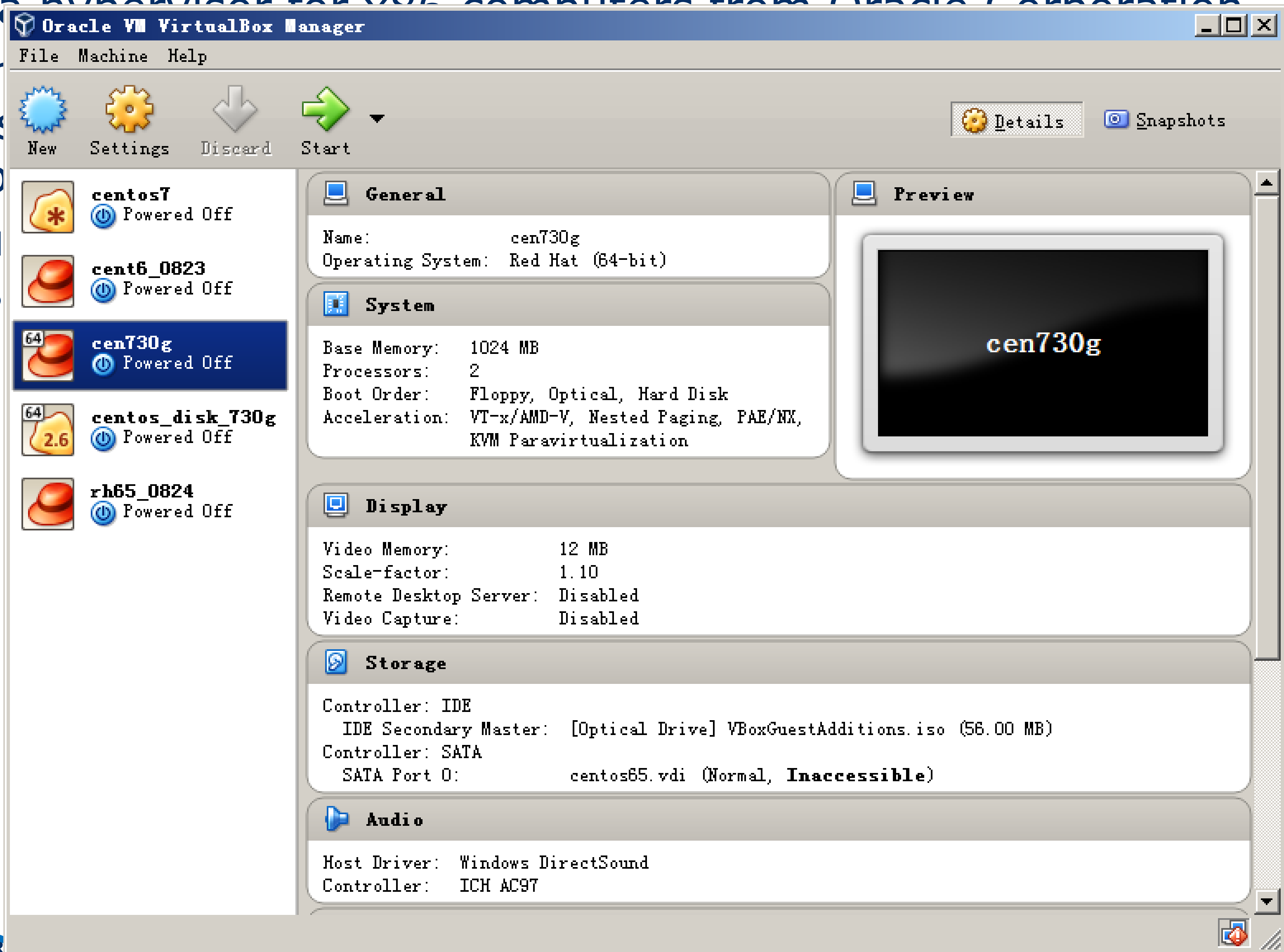
n

l

- VMwar

virtualbox

- is a hypervisor for x86 computers from Oracle Corporation
- Vir
  sys
  Op
- Su
  as



LXC (linux container) and Docker

- **LXC (Linux Containers)**
  - is an operating-system-level virtualization environment for running multiple isolated Linux systems (containers) on a single Linux control host.
  - The Linux kernel provides the cgroups functionality that allows limitation and prioritization of resources (CPU, memory, block I/O, network, etc.)
- **Docker**
  - a project automating deployment of applications inside software containers
  - can also use LXC as one of its execution drivers, enabling image management and providing deployment services.

Agenda

- Introduction
- Virtualization Technologies
  - CPU
  - Memory
  - IO
- Commercial virtualization tools
- Problems in using virtualization
  - How to create and handle VM images
  - Virtual network concepts and configuration
- Summary

Libvirt – API for managing VMs

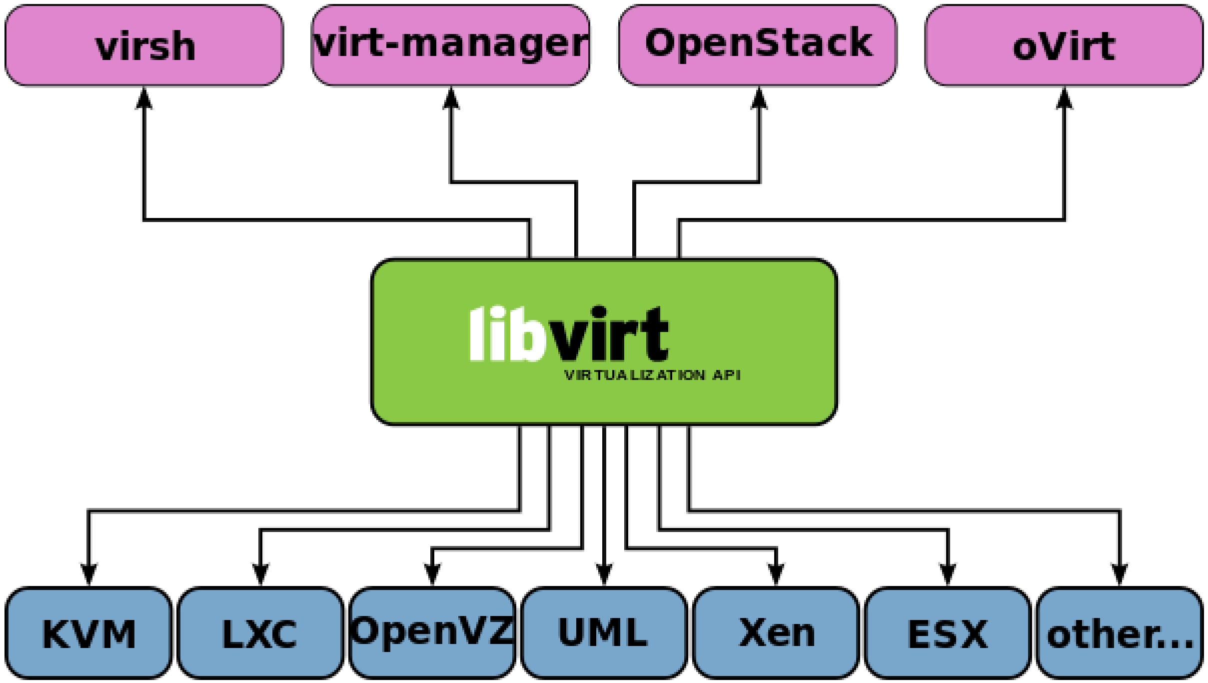
- is an open source API, daemon and management tool for managing platform virtualization.
  - itself is a C library.
  - Others, Python, Perl, OCaml, Ruby, Java, and PHP.
  - widely used in the orchestration layer of hypervisors in the development of a cloud-based solution.

- can be used to manage
  - KVM, Xen, VMware ESX, QEMU and other virtualization technologies.
- Two **User Interfaces**
  - Graphical Interface: virt-manager
  - Command line interface: virsh

Warnke, Robert; Ritzau, Thomas. *qemu-kvm & libvirt* (in German)

https://en.wikipedia.org/wiki/Libvirt

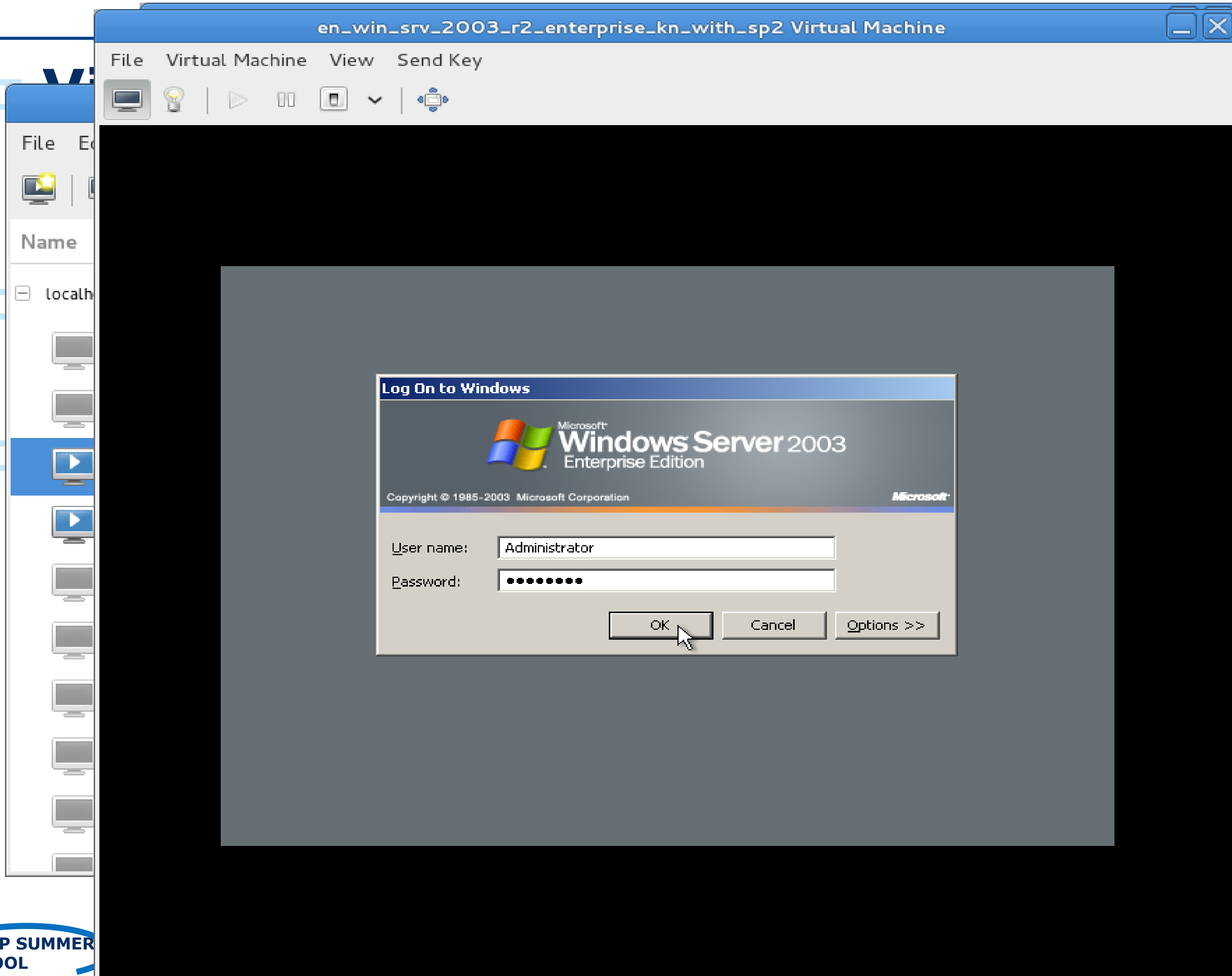
libvirt



- libvirt supports several Hypervisors and is supported by several management solutions

virt-manager

http://virt-manager.org/



virsh

http://blog.csdn.net/bravezhe/article/details/8465678

```
[root@localhost Desktop]# virsh
Welcome to virsh, the virtualization interactive terminal.

Type:  'help' for help with commands
       'quit' to quit

virsh #
```

- The virsh is used to manage domains. You must run the commands as the root user or by assuming the appropriate role account on the host operating system. The commands cannot be run in the guest domain.
- Common used commands:
  - virsh create #1: start a VM
    - virsh create vm01.xml
  - virsh destroy #1
  - virsh list
    - list all the running VMs
  - virsh console: connect to a VM
  - virsh reboot #1
  - virsh shutdown #1
  - …

VM image file

- VM Image file stores all contents of a VM.
- Two modes: **mirror and sparse mode**
  - The mirror mode image stores all byte data, includes invalid data for users. **Raw format** is a **mirror mode image**.
  - The sparse mode image stores the user or system valid data, only occupy a special necessary size of storage space. The **qcow2** and **vmdk** are **sparse mode file formats**.
- File format
  - General image: qcow, qcow2 & raw
  - Commercial image: vmdk, vdi, …
    - Each commercial virtualization has its own image format.

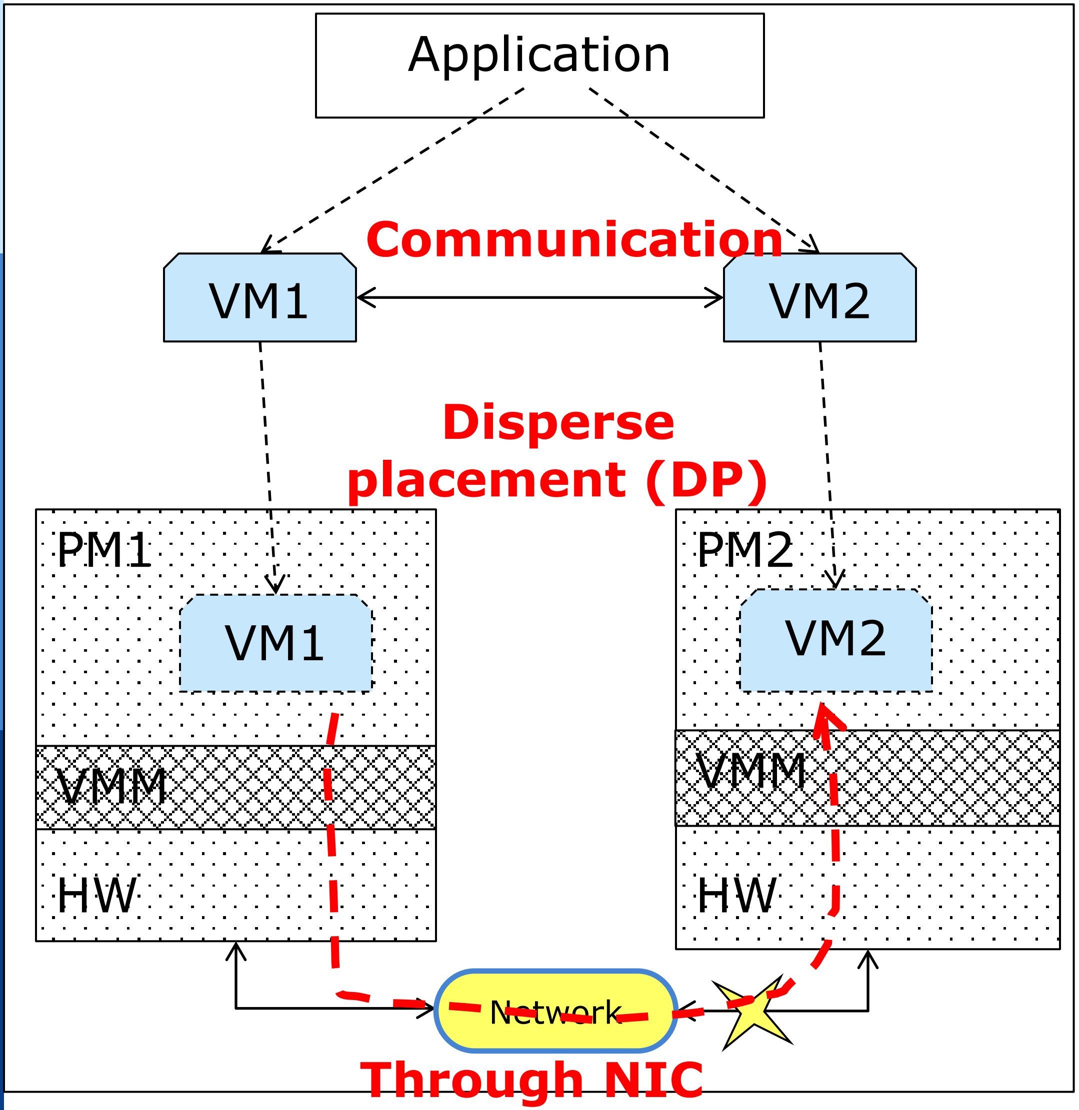
Choosing Image format

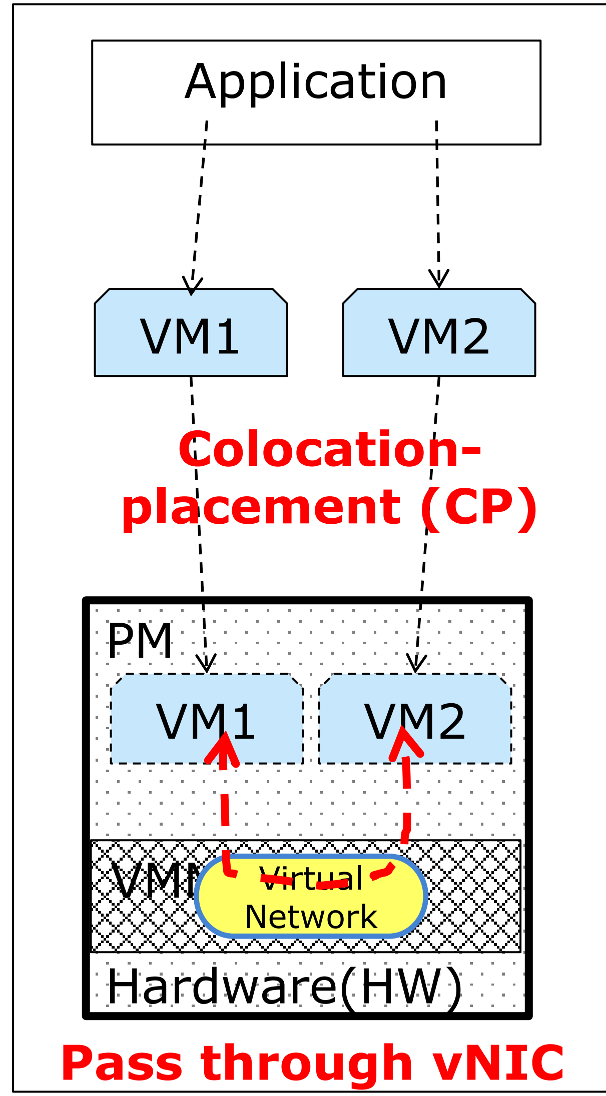
- raw format (KVM/Xen)
  - (**linux default**) the raw format is a plain binary image of the disc image, and is very portable.
  - On file systems that support sparse files, images in this format only use the space actually used by the data recorded in them.
- qcow2 (QEMU copy-on-write)
  - QEMU copy-on-write format with a range of special features, including the ability to take multiple snapshots, smaller images on file-systems that don’t support sparse files, optional AES encryption, and optional zlib compression.
- Other
  - vmdk: VMWare file format
  - vdi: virtualbox file format
- Image File format convert
  - In general a commercial VMM can convert its format file to the raw formant

Agenda

- Introduction
- Virtualization Technologies
  - CPU
  - Memory
  - IO
- Commercial virtualization tools
- Problems in using virtualization
  - How to create and handle VM images
  - Virtual network concepts and configuration
- Summary

Network Communication between VMs





Pass through vNIC but not NIC

Virtual network concepts

- Physical network
  - NIC, switch
- Virtual Network
  - vNIC, vSwitch
- Virtual network devices
  - TAP/TUN
  - Linux Bridge

http://www.ibm.com/developerworks/cn/linux/1310_xiawc_networkdevice/

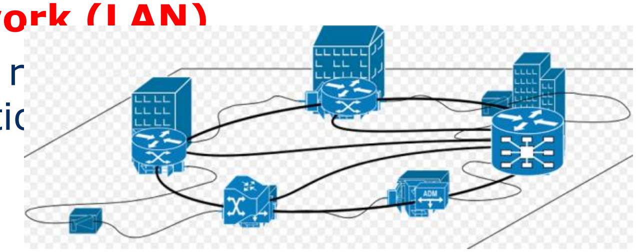
https://www.vmware.com/support/ws55/doc/ws_net_basics.html

http://blog.csdn.net/tantexian/article/details/45395075

Physical network

- Many computers with distinct geolocations are connected together, creating a physical network.
- Network connection needs network devices.
  - **Network interface controller(NIC)**
    - A computer or server must be network-capable with a working **NIC** or a **network card** installed.
    - The **NIC** enables the computer to interact with a network.

  - **Switch and local area network (LAN)**
    - Computers are usually connected to a device called a **switch**, which creates a **local area network (LAN)**
    - **Switches** are responsible for intelligently r network traffic to the appropriate destinatic



  - Network protocol: TCP/IP

Virtual network

- In virtualization network devices are virtualized, such as vNIC and vSwitch.
- Many VMs connected together with virtual network devices creates a virtual network.
  - **Virtual network interface card (vNIC)**
    - Hypervisor can create one or more vNICs for each VM.
    - The **vNIC** provides the networking capabilities of the VM.
    - Each vNIC is identical to a physical NIC.
  - **Virtual Switch(vSwitch)**
    - **Switch** also can be virtualized as a **virtual switch**.
    - **Each vNIC** is connected to the **vSwitch port**, and these vSwitch access external physical network through **the physical NIC** of Physical Server.
  - e.g., TAP/TUN

TAP/TUN

- **TAP / TUN**
  - is a pair of virtual network devices based on Linux kernel implementation.
  - is widely used to realize the connection between the VM and PM.
- **TAP**
  - is equivalent to an **Ethernet device**, which operates the **second-layer packets** such as Ethernet data frames. (can see in linux by ifconfig)
- **TUN**
  - simulates **the network layer devices**, and operate the **third layer packet** such as IP data packets. ➔router or switch (vSwitch)

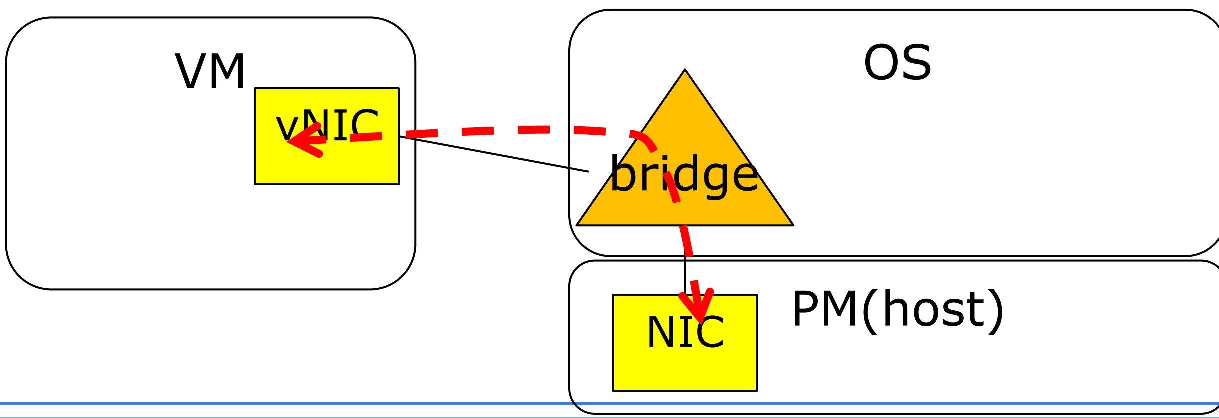
Tap devices in KVM machine

- One VM has a tap device

```
[root@compute2 ~]# ifconfig |grep tap
tap0920d05a-38: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tap1c52b862-e1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tap2efedf47-39: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tap45787583-0b: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tap5140c9f1-87: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tap56b1fa1f-e5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tap69383260-61: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tap6bbe47a7-f2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tap7b92e41b-a3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tap8213fc6e-f9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tap833640f0-24: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tap915224e1-da: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tap9bc4ac99-b9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tapb8e468e6-59: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tapbb4b88b7-1f: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tapcaf75f1d-9b: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tapdaa7ce9a-7c: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tapdea66e2e-63: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
tape28c2f3a-a5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
```

Linux bridge

- Linux Bridge
  - is a **virtual network device** working on **the second layer**
  - has the functions similar to the **physical switches**.
- Bridge can
  - bind other Linux network device as a slave device, and virtualize the slave device as a port.

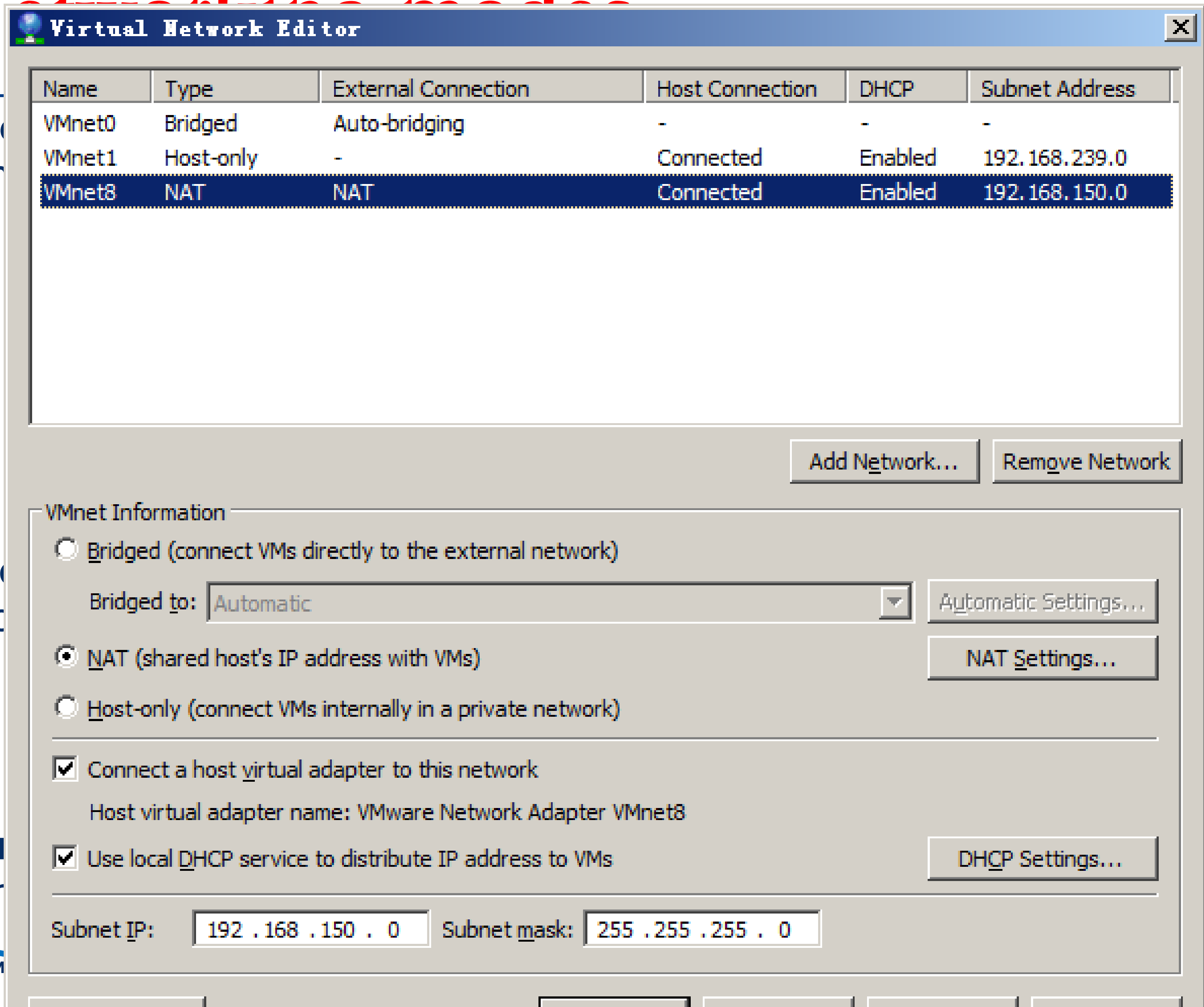


Network configuration

- Networking modes
- Example of virtual network configuration

N

- N
  sh
- H
  ac
- B
  or



Virtual Network Editor
Name
Type
External Connection
Host Connection
DHCP
Subnet Address
VMnet0
Bridged
Auto-bridging
-
-
-
VMnet1
Host-only
-
Connected
Enabled
192.168.239.0
VMnet8
NAT
NAT
Connected
Enabled
192.168.150.0
Add Network...
Remove Network
VMnet Information
Bridged (connect VMs directly to the external network)
Bridged to:
Automatic
Automatic Settings...
NAT (shared host's IP address with VMs)
NAT Settings...
Host-only (connect VMs internally in a private network)
Connect a host virtual adapter to this network
Host virtual adapter name: VMware Network Adapter VMnet8
Use local DHCP service to distribute IP address to VMs
DHCP Settings...
Subnet IP:
192 . 168 . 150 . 0
Subnet mask:
255 . 255 . 255 . 0

Example of bridge networking

- Hardware environment:
  - A host is running centos with virtualization KVM.
    - A VM is required to configure the virtual network in bridge mode.
    - The host is required to configure a bridge and bind the NIC to this bridge.
- Configuration steps
  - Step1. modify ifcfg-ethX in host
  - Step2. create a new bridge in host
  - Step3. modify VM configure file (.xml) in host
  - Step4. Connect VM and configure VM network

Step1: modify ifcfg-ethX

- 1. Modify /etc/sysconfig/network-scripts/ifcfg-ethX. *e.g.*

  ***vi* /etc/sysconfig/network-scripts/ifcfg-eth0**

  DEVICE="eth0"
  NM_CONTROLLED="no"
  ONBOOT="yes"
  TYPE=Ethernet
  BOOTPROTO=none
  BRIDGE="br0" *#Add this line bind br0*
  NAME="System eth0"
  HWADDR=44:37:E6:4A:62:AD

- --

(In host machine)

Step2: create a new bridge in host

Add a new file, /etc/sysconfig/network-scripts/ifcfg-br0.

```
#----------------------------------------
DEVICE="br0"
ONBOOT="yes"
TYPE="Bridge"                          #it's bridge
BOOTPROTO=static
IPADDR=10.0.112.39                     #ip addr.
NETMASK=255.255.255.0                  #netmask
GATEWAY=10.0.112.1                     #gateway
DEFROUTE=yes
#--------------------------------------------------------------
```

And restart network device and check bridge interface info.

#service network restart *#restart network service*

#brctl show *#show bridge list*

bridge name	bridge id	STP enabled	interfaces
br0	8000.4437e64a62ad	no	eth0 #bind eth0

Step3. Modify VM configure file (.xml)

Add the following code to VM configure file XXX.xml:

```
<devices>
    <interface type='bridge'>
        <source bridge='br0'/>
        <target dev='vnet0'/>
        <mac address="00:11:22:33:44:55"/>
    </interface>
</device>
```

Boot the VM to check network link valid or not.

```
#virsh create XXX.xml
#brctl show
bridge name     bridge id          STP enabled     interfaces
br0             0.4437e64a62ad     no              eth0 vnet0
```

Step4. Connect VM and configure VM network

- Use vnc to connect VM. Firstly use virsh dumpxml to view the vnc interface number. Then use virt-viewer to connect to the VM.

virt-viewer localhost:5901

- Login the VM and complete the network configuration. The operation in VM is the same as physical machine.

Summary

- What is virtualization
- Virtualization technology implementation principles
  - CPU, Memory, IO virtualization
- Virtualization tools
  - KVM, Xen, Docker
- Two problems in practical application
  - Manage VM images
  - Virtual network concepts

End

Thank You!