Virtualization Concepts And Applications

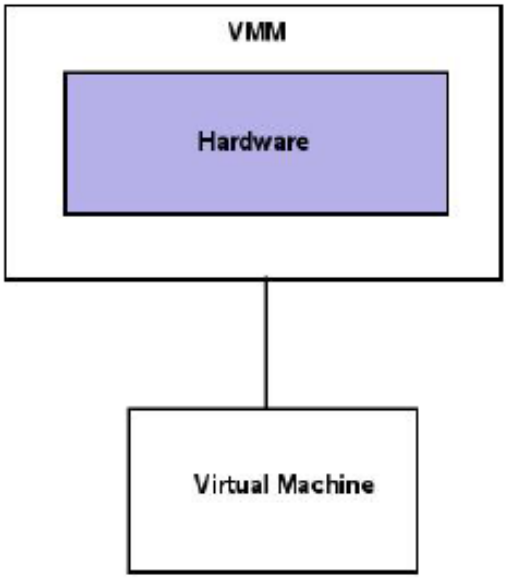
Yash Jain
DA-IICT
(DCOM Research Group)

Virtualization

- "Virtualization is a framework or methodology of dividing the resources of a computer into multiple execution environments, by applying one or more concepts or technologies such as hardware and software partitioning, time-sharing, partial or complete machine simulation, emulation, quality of service, and many others.."

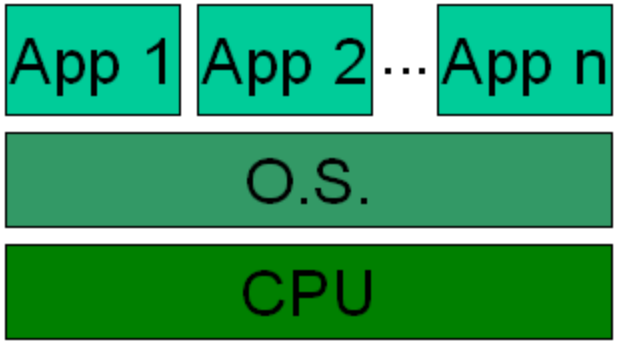
- We're used to a simple equation, one physical machine runs one OS at any given time.

- By virtualizing the machine, we are able to run several operating systems (and all of their applications) at the same time.

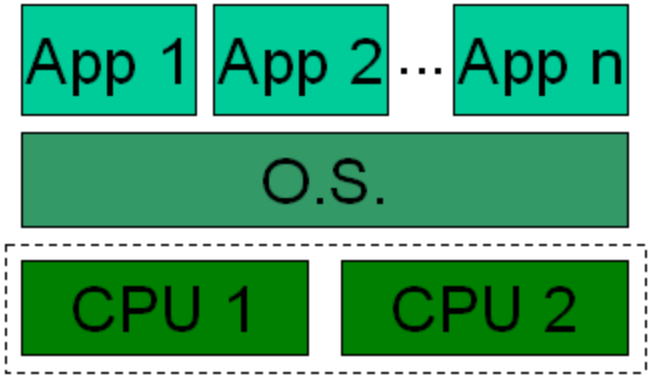


The Virtual Machine Monitor

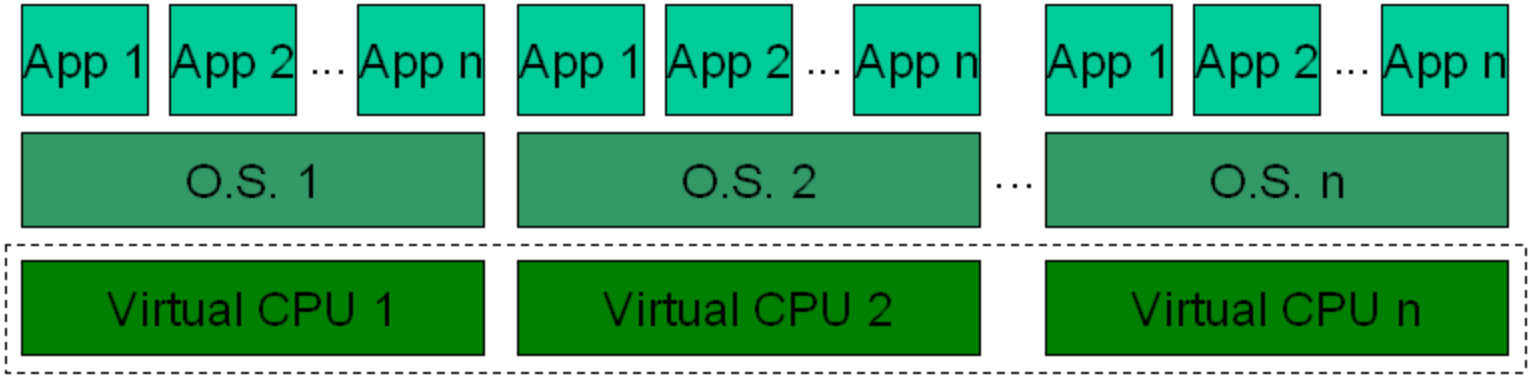
Multi-tasking, Multi-threading And Virtualization



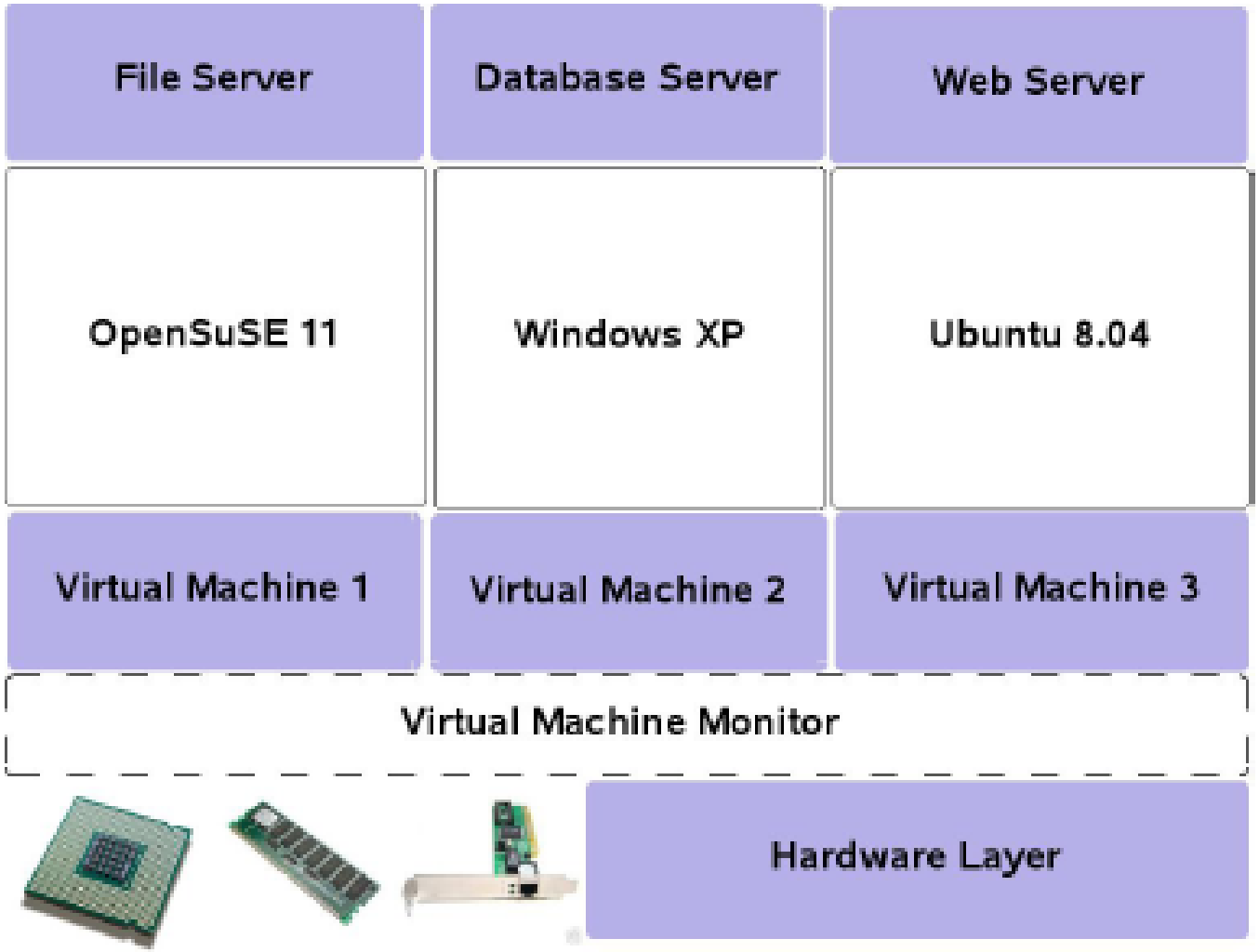
Multi-tasking



HyperThreading



Virtualization



A Virtualized Server Hosting Three Different Applications

Definitions

Virtual Machines

- A representation of a real machine using software that provides an operating environment which can run or host a guest operating system. Virtual machines are created and managed by virtual machine monitors.

Guest Operating System

- Operating system which is running inside the created virtual machine.

Definitions (cont..)

Hypervisor

- A thin layer of software that generally provides virtual partitioning capabilities, which runs directly on hardware, but underneath higher-level virtualization services, sometimes referred to as a “bare metal” approach.

Definitions (cont…)

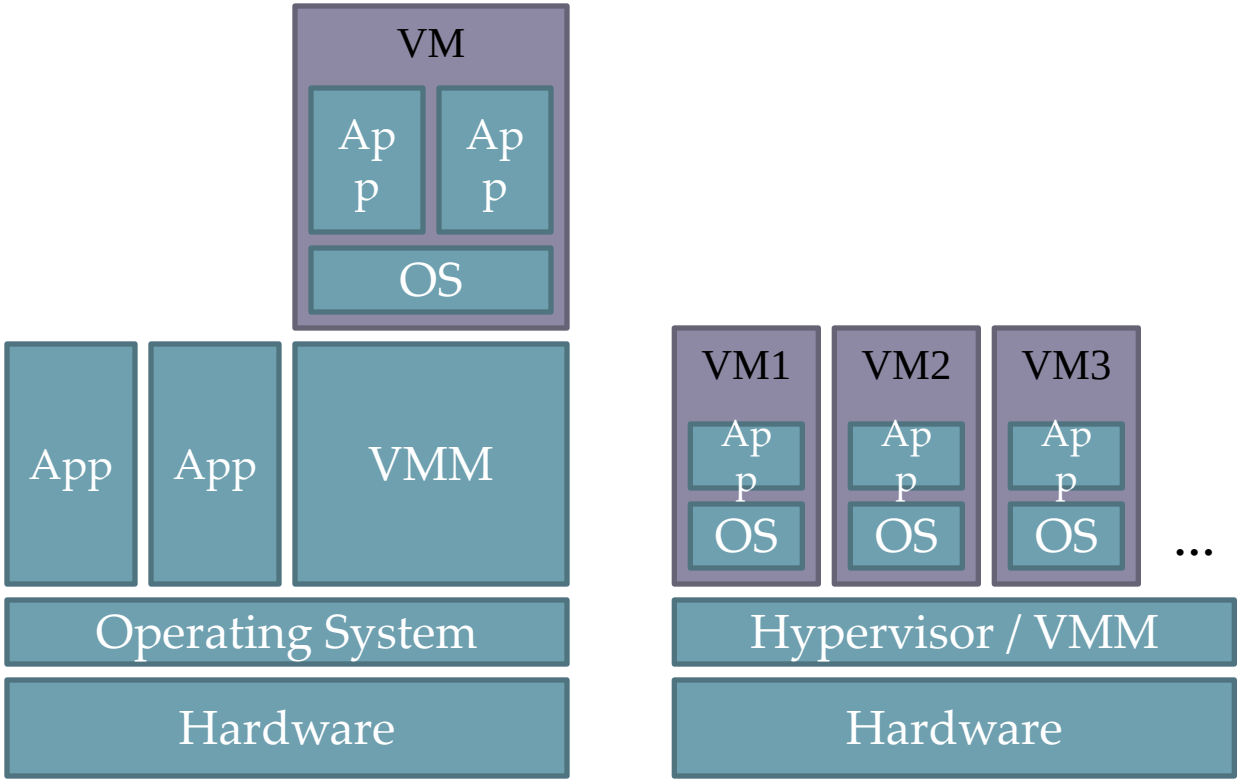
Virtual Machine Monitor

- Recent articles use the terms like hypervisor and virtual machine monitor interchangeably, but they are separated two in conceptual model.
- Software that runs in a layer between host operating system and one or more virtual machines that provides the virtual machine abstraction to the guest operating systems.
- With full virtualization, the virtual machine monitor exports a virtual machine abstraction identical to a physical machine, so that standard operating systems can run just as they would on physical hardware.

Virtualization

- Virtualization enables enterprises to
  - consolidate multiple servers without sacrificing application isolation,
  - scale their infrastructure as their needs grow,
  - increase availability through dynamic provisioning and relocation of critical systems.
- Examples of Virtual Machine Manager
- Xen, KVM, VMWare.

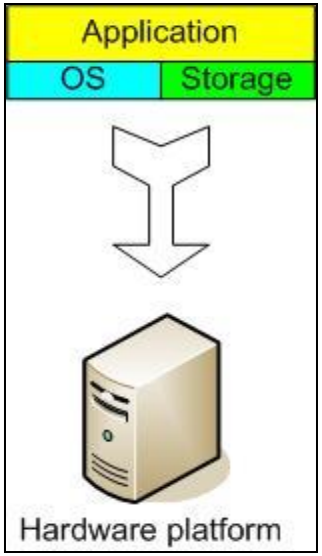
Virtualization

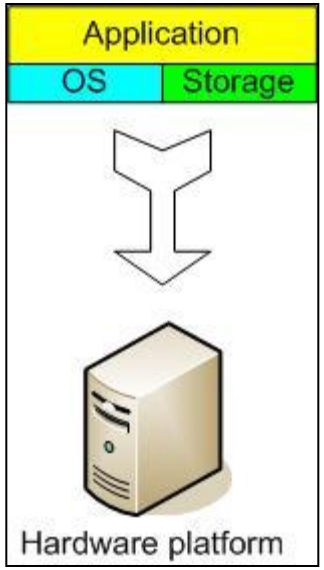


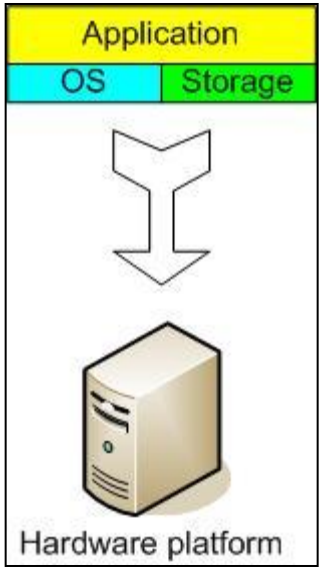
Concept 1

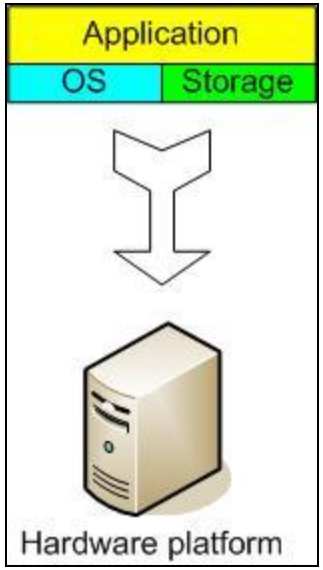
Concept 2

The Traditional Server Concept









Web Server

Windows

IIS

App Server

Linux

Glassfish

DB Server

Linux

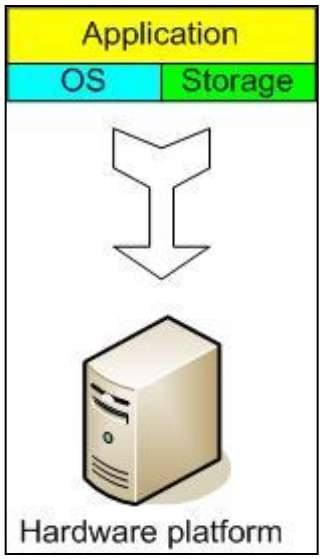
MySQL

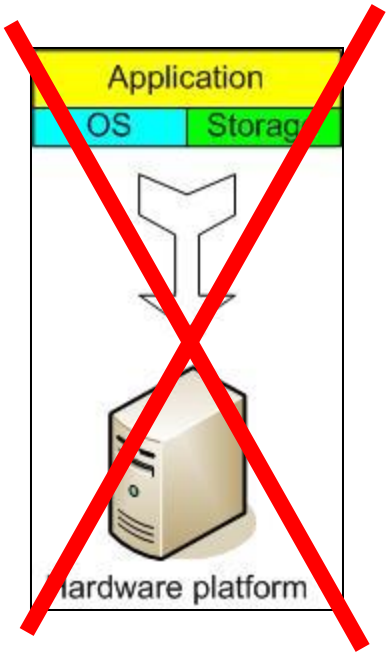
EMail

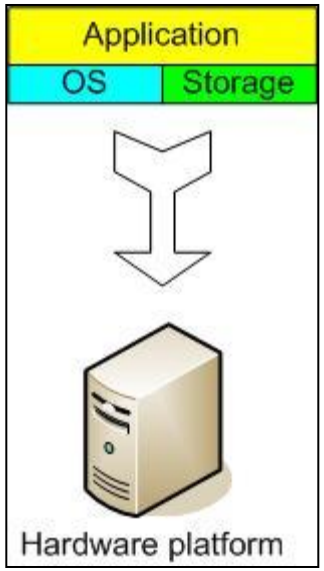
Windows

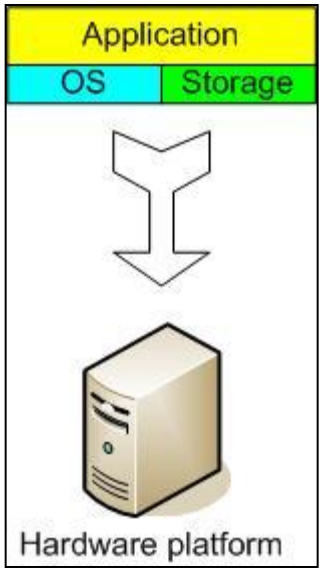
Exchange

And if something goes wrong ...









Web Server

Windows

IIS

App Server

DOWN!

DB Server

Linux

MySQL

EMail

Windows

Exchange

The Traditional Server Concept

- System Administrators often talk about servers as a whole unit that includes the hardware, the OS, the storage, and the applications.
- Servers are often referred to by their function i.e. the Exchange server, the SQL server, the File server, etc.
- If the File server fills up, or the Exchange server becomes overtaxed, then the System Administrators must add in a new server.

The Traditional Server Concept

- Unless there are multiple servers, if a service experiences a hardware failure, then the service is down.
- System Admins can implement clusters of servers to make them more fault tolerant. However, even clusters have limits on their scalability, and not all applications work in a clustered environment.

The Traditional Server Concept

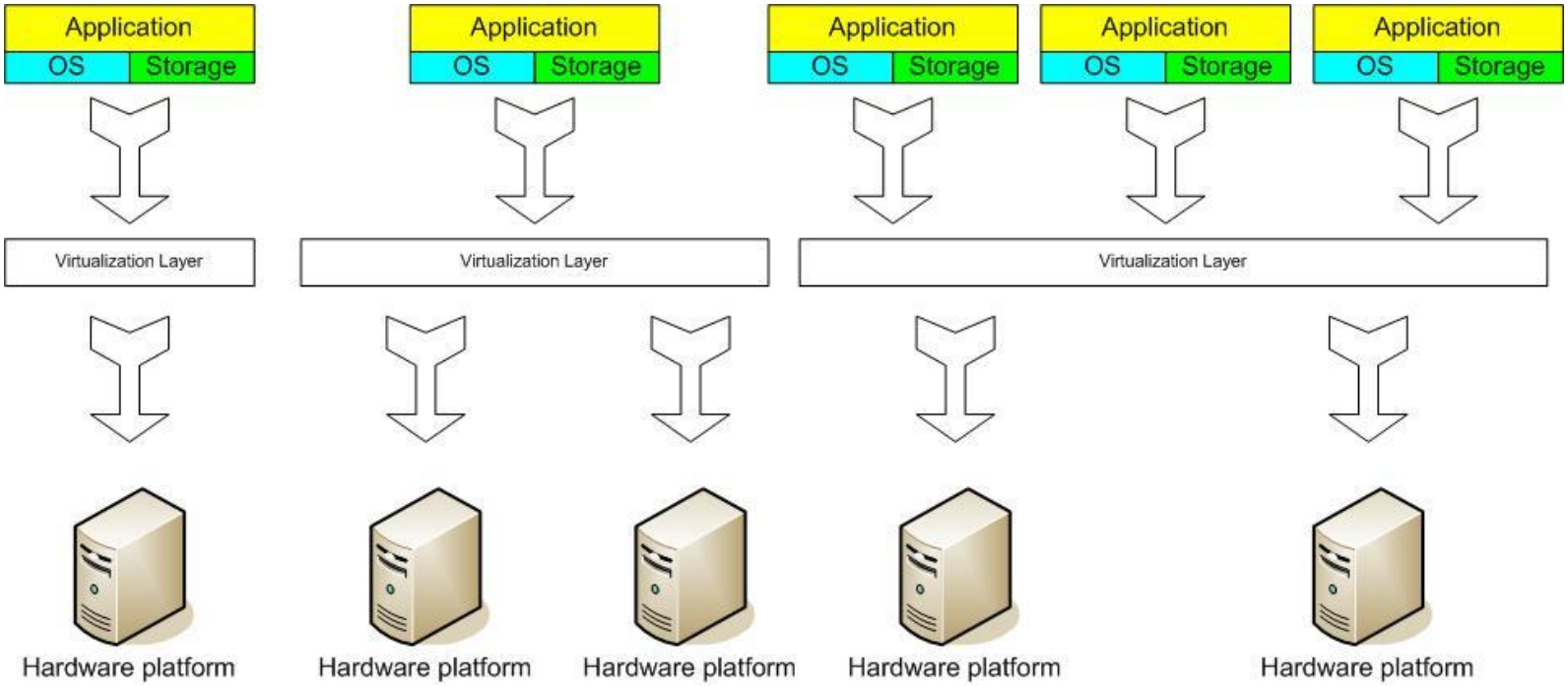
Pros

- Easy to conceptualize
- Fairly easy to deploy
- Easy to backup
- Virtually any application/service can be run from this type of setup

Cons

- Expensive to acquire and maintain hardware
- Not very scalable
- Difficult to replicate
- Redundancy is difficult to implement
- Vulnerable to hardware outages
- In many cases, processor is under-utilized

The Virtual Server Concept



Virtual Machine Monitor (VMM) layer

between

***Guest OS* and hardware**

The Virtual Server Concept

- Virtual servers seek to encapsulate the server software away from the hardware
  - This includes the OS, the applications, and the storage for that server.
- Servers end up as mere files stored on a physical box, or in enterprise storage.
- A virtual server can be serviced by one or more hosts, and one host may house more than one virtual server.

The Virtual Server Concept

- Virtual servers can still be referred to by their function i.e. email server, database server, etc.
- If the environment is built correctly, virtual servers will not be affected by the loss of a host.
- Hosts may be removed and introduced almost at will to accommodate maintenance.

The Virtual Server Concept

- Virtual servers can be scaled out easily.
- If the administrators find that the resources supporting a virtual server are being taxed too much, they can adjust the amount of resources allocated to that virtual server
- Server templates can be created in a virtual environment to be used to create multiple, identical virtual servers
- Virtual servers themselves can be migrated from host to host almost at will.

The Virtual Server Concept

Pros

- Resource pooling
- Highly redundant
- Highly available
- Rapidly deploy new servers
- Easy to deploy
- Reconfigurable while services are running
- Optimizes physical resources by doing more with less

Cons

- Slightly harder to conceptualize
- Slightly more costly (must buy hardware, OS, Apps, and now the abstraction layer)

Why to virtualize?

- **Flexibility:** more than one instance
- **Availability:** temporary migration, if physical node is down
- **Scalability:** very easy to insert a physical node with the basic cluster
- **Hardware utilization:** virtual machines utilize hardware resources that are left idle
- **Security:** Using multiple virtual machines, it is possible to separate services by running one service on each virtual machine. This approach is also called jailing of services.

Types of Virtualization

- Two kinds of virtualization approaches available according to VMM
- **Hosted** When VMM runs in a operating system,
- **Bare-metal** approach runs VMM on top of hardware directly.
  - Fairly complex to implement but good in performance.

Types of Virtualization (cont)

Emulation

- A virtual machine simulates the entire hardware set needed to run unmodified guests for completely different hardware architectures.
- Used to create new Operating Systems for the hardware which is in design phase and not in physical form
- Examples: Bochs and QEMU

Types of Virtualization (cont)

Full Virtualization

- Native virtualization.
- It is similar to emulation except it is designed to simulate the underlying hardware which is physically available.
- Runs unmodified guests on a physical machine.
- It gives the flexibility to move entire virtual machines from one host to another host very easily, but for the cost of performance due to the overhead added by the emulator Layer
- Examples: Virtual PC and VMware Workstation

Types of Virtualization (cont)

Full Virtualization

- VMware is the first commercial virtualization product provider for x86 architecture.
- It has a bare metal product, ESX server.
- It enables the execution of unmodified guest operating systems through on-the-fly translation of x86 instructions that can not be virtualized.
- Vmware Player is a free hosted VMM by Vmware
- VirtualPC from Microsoft
- Hyper-V, a stand alone product and as a feature for Windows Server 2008, windows edition translates guest kernel mode and real mode into x86 user mode

Types of Virtualization (cont)

Para Virtualization

- Used by Xen, Denali and VMware ESX
- The hypervisor exports a modified version of the underlying physical hardware.
- It provides better performance than full virtualization.
- Examples: user Mode Linux and Xen.

Types of Virtualization (cont)

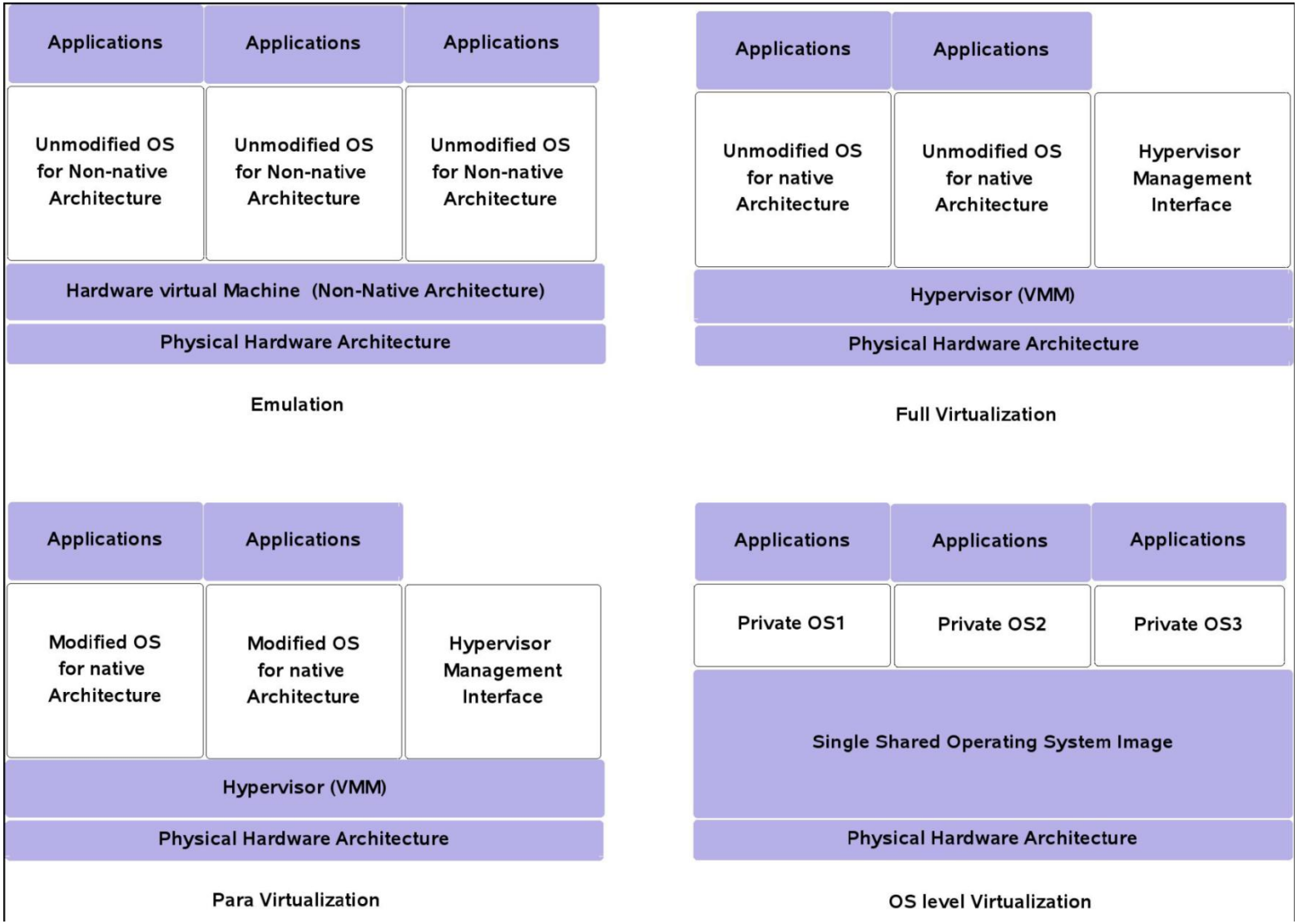
OS level Virtualization

- No requirement of virtual machine monitor software
- Single OS image handles all the guest images in different isolated containers
- OS level virtualization does not support running different operating systems (Specifically, different kernel) at a time
- Examples: Virtuozzo, Linux VServers and OpenVZ.

Types of Virtualization (cont)

Application Virtualization

- Referred as process virtualization.
- Application virtualization is the approach of running applications inside a virtual execution environment (Managed Run-time).
- The virtual execution environment provides a standard API for cross platform execution and manages the consumption of application's local resources, e.g.
- Threading model, environment variables, user interface libraries and objects.



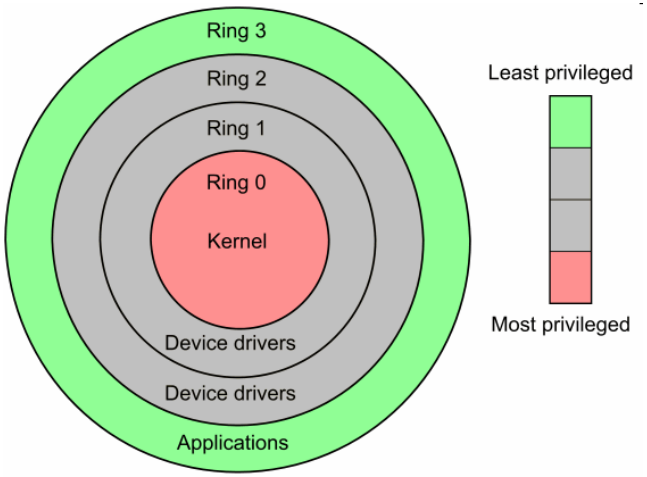
Types of Virtualization

Hardware Support

- Much of the virtualization overhead today are due to processors not being designed with virtualization in mind
- Seminal paper by Popek, Goldberg (1974) provides basic guidelines
- Efficient VMM can be designed if:
- Processor has protection mechanisms, and
- Privileged instructions that read/write system status must cause exceptions if run at non-privileged level
- Modern microprocessors support protection (condition 1)
- However, second condition rarely satisfied
- E.g. Intel 'x86': 17 problematic instructions, E.g.: POPF instruction for setting interrupt flag.
- SGDT, SIDT, PUSHF, POPF.....
- VMs must use several software 'tricks' to circumvent problematic instructions in processors such as Intel/AMD
- Processors are now being redesigned to support more efficient VMs
- Intel VT in dual-core "Yonah"; AMD "Pacifica"

x86 Privilege Rings

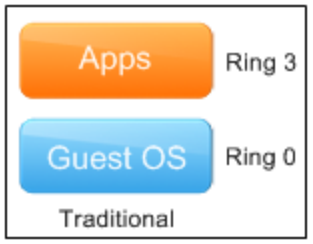
- x86 CPUs provide a range of protection levels also known as rings in which code can execute. Ring 0 has the highest level privilege and is where the operating system kernel normally runs. Code executing in Ring 0 is said to be running in system space, kernel mode or supervisor mode. All other code such as applications running on the operating system operate in less privileged rings, typically Ring 3.



Rings in virtualization

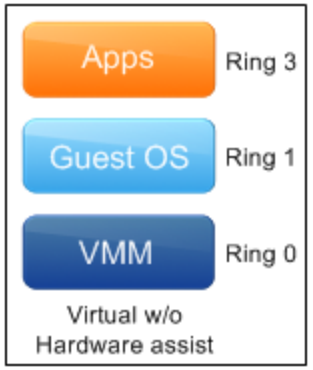
Traditional systems

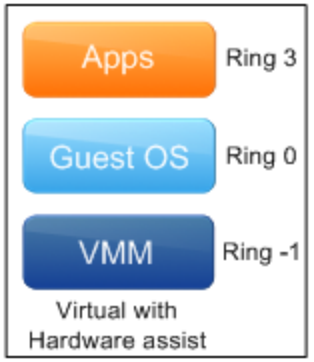
- Operating system runs in privileged mode in Ring 0 and owns the hardware
- Applications run in Ring 3 with less privileges runs in privileged mode in Ring 0



Virtualized systems

- VMM Guest OS inside VMs are fooled into thinking they are running in Ring 0, privileged instructions are trapped and emulated by the VMM
- Newer CPUs (AMD-V/Intel-VT) use a new privilege level called Ring -1 for the VMM to reside allowing for better performance as the VMM no longer needs to fool the Guest OS that it is running in Ring 0.

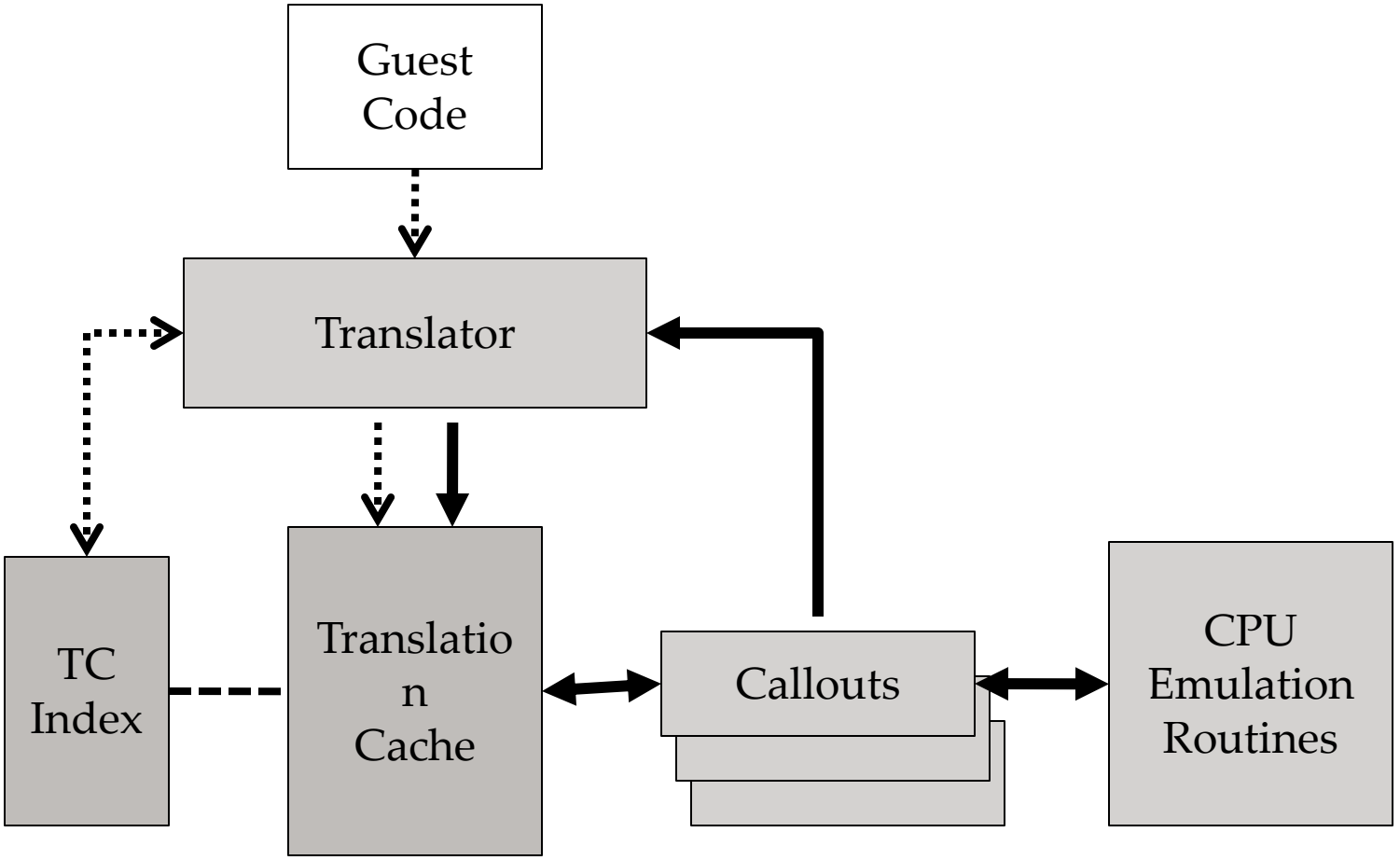




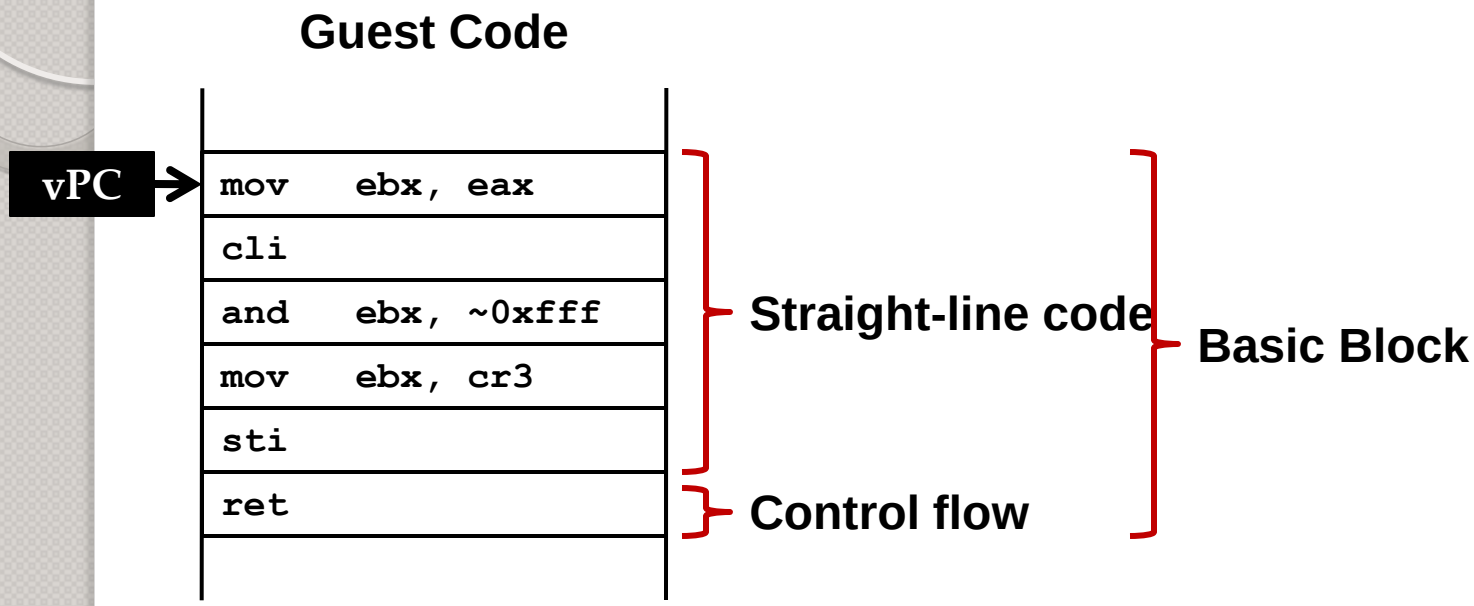
X86 Processor Virtualization

- x86 architecture is not fully virtualizable
  - Certain privileged instructions behave differently when run in unprivileged mode
  - Certain unprivileged instructions can access privileged state
- Instructions do not satisfy this, E.g.: POPF instruction for setting interrupt flag.
  - SGDT, SIDT, PUSHF, POPF.....
- Techniques to address inability to virtualize x86
  - Replace non-virtualizable instructions with easily virtualized ones statically (Paravirtualization)
  - Perform Binary Translation (Full Virtualization)

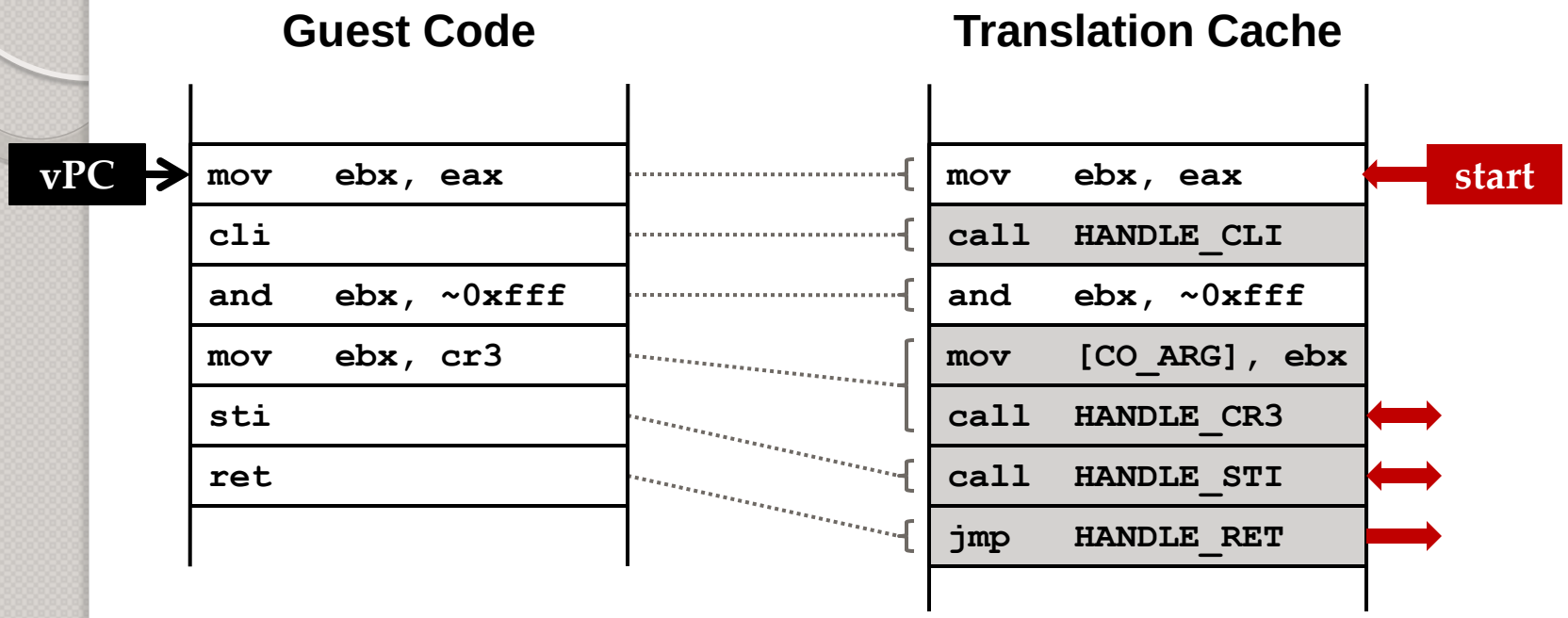
Binary Translator



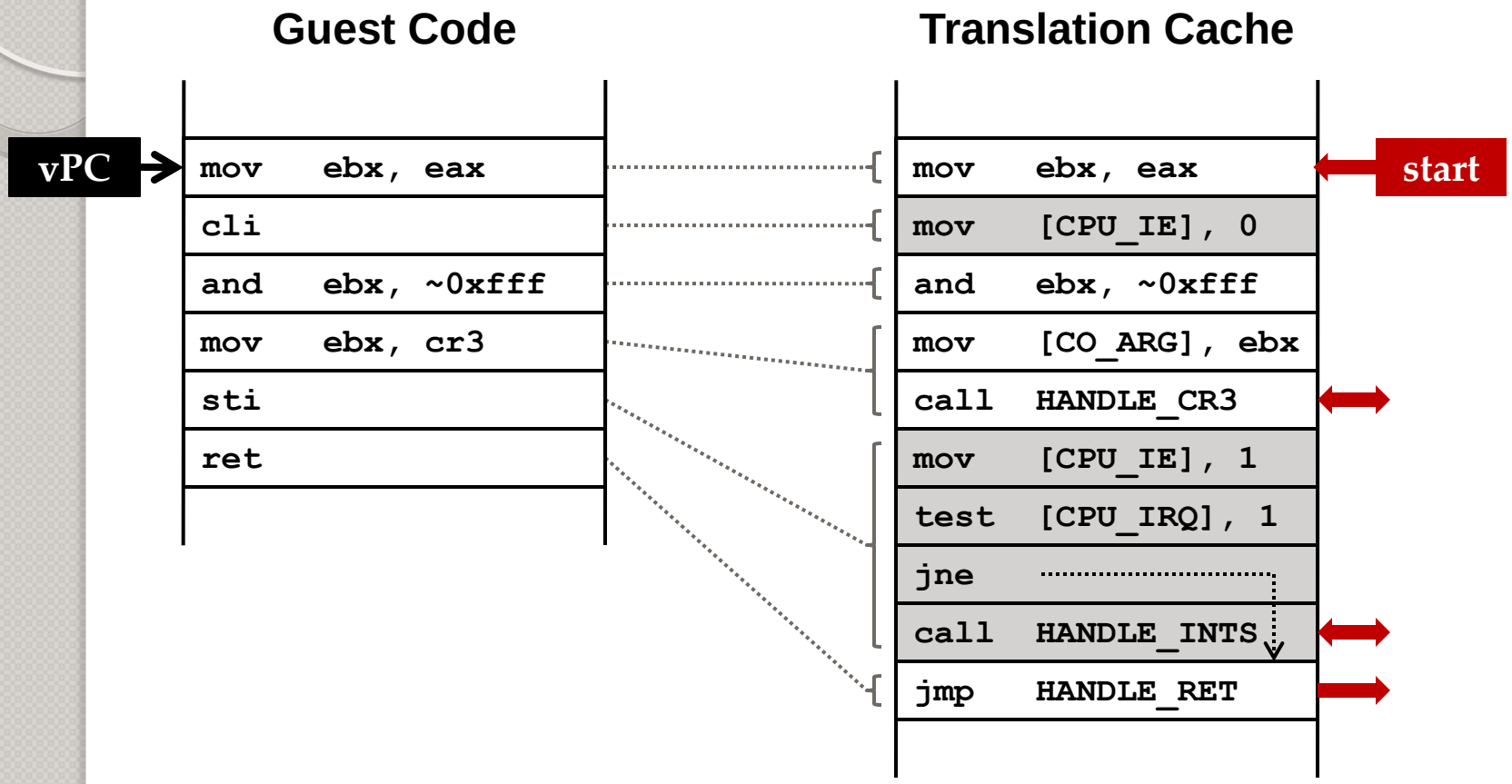
Basic Blocks



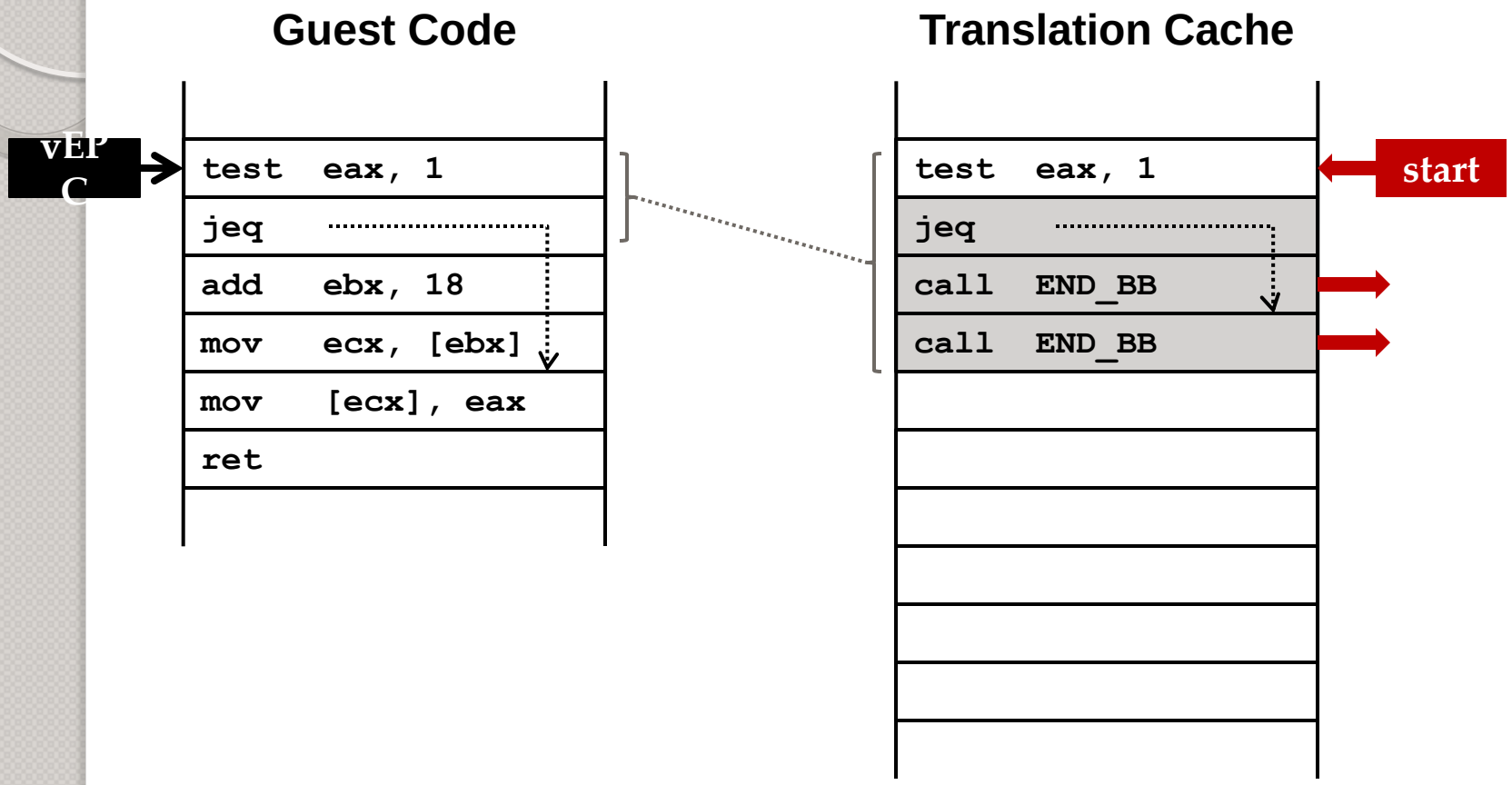
Binary Translation



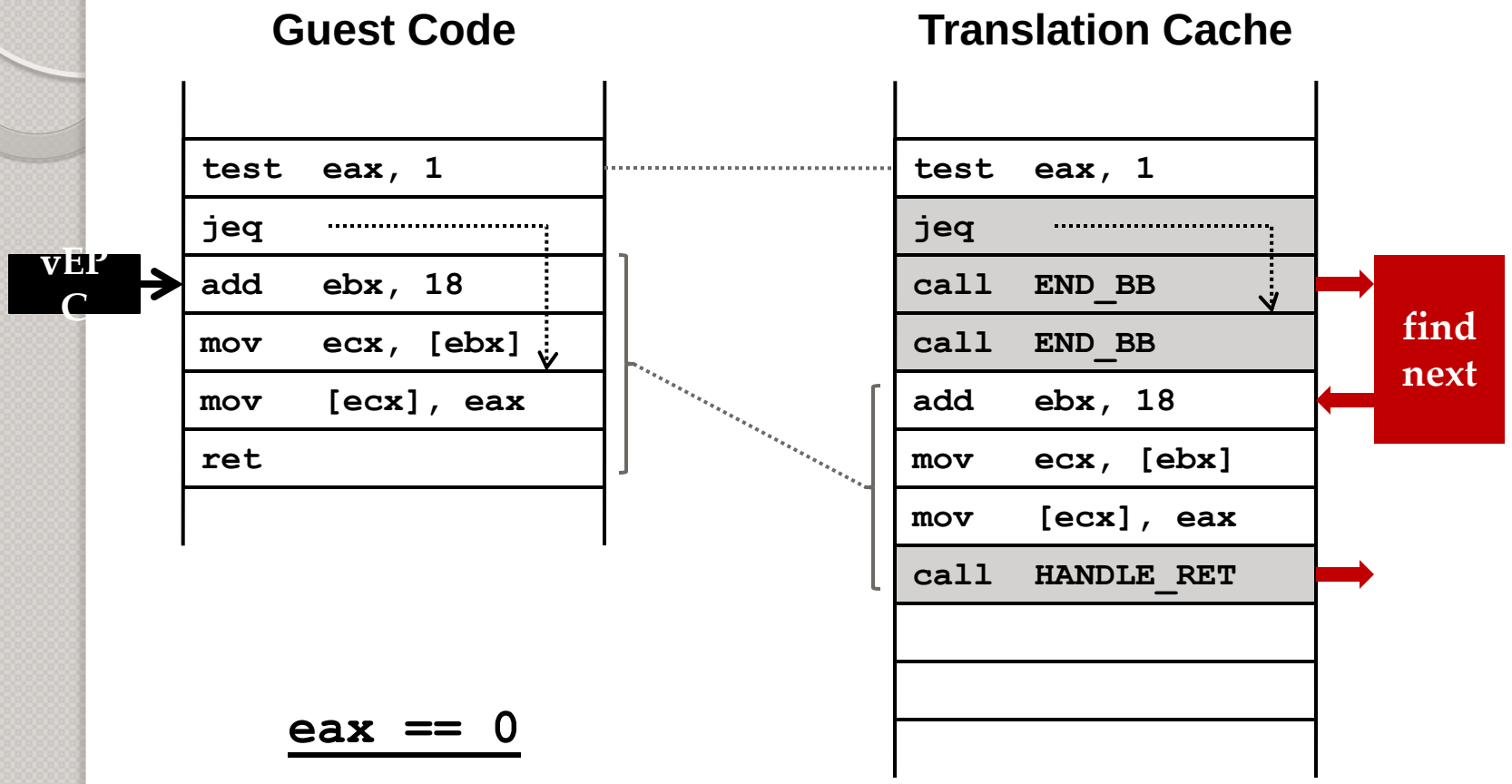
Binary Translation



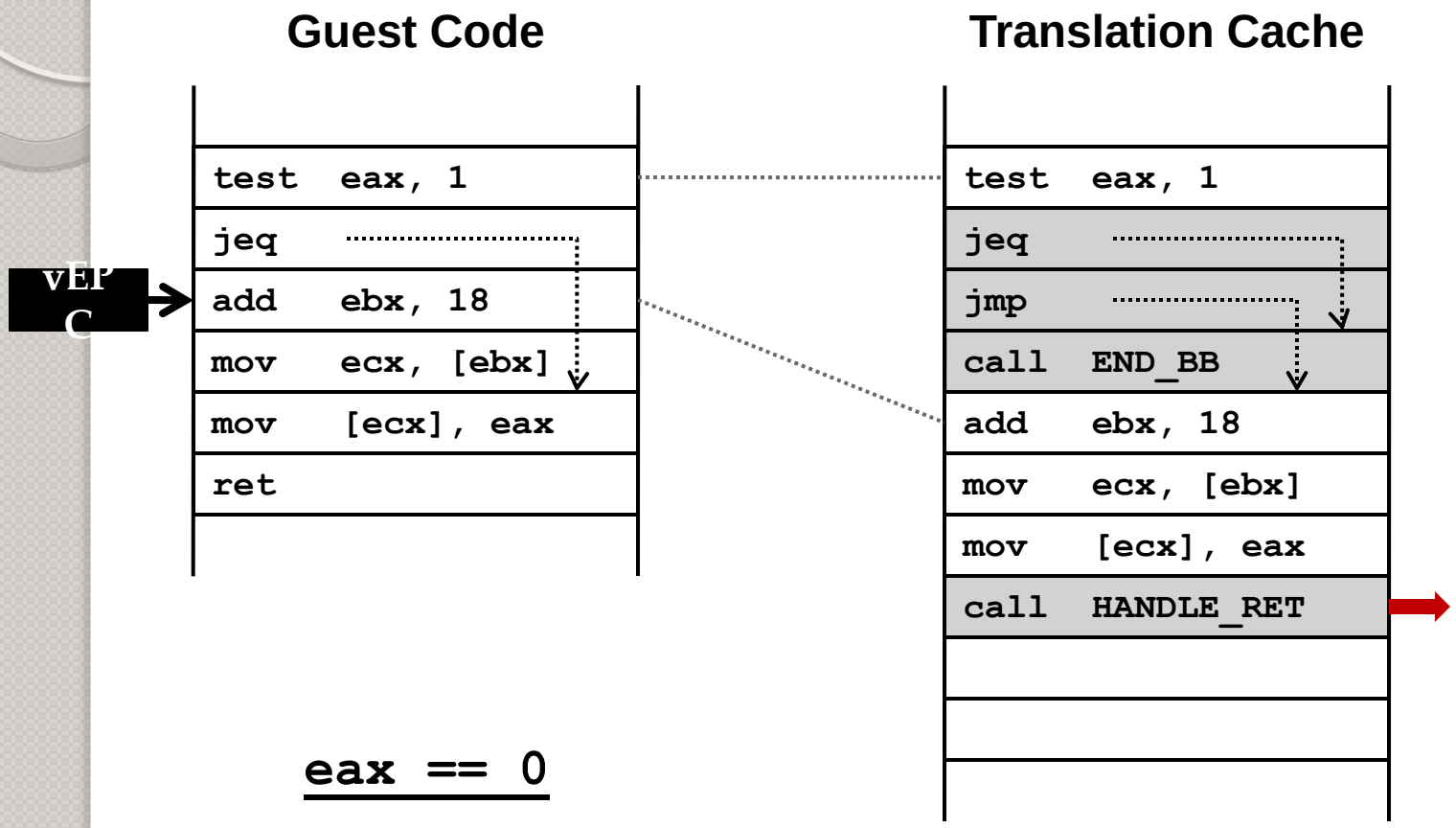
Controlling Control Flow



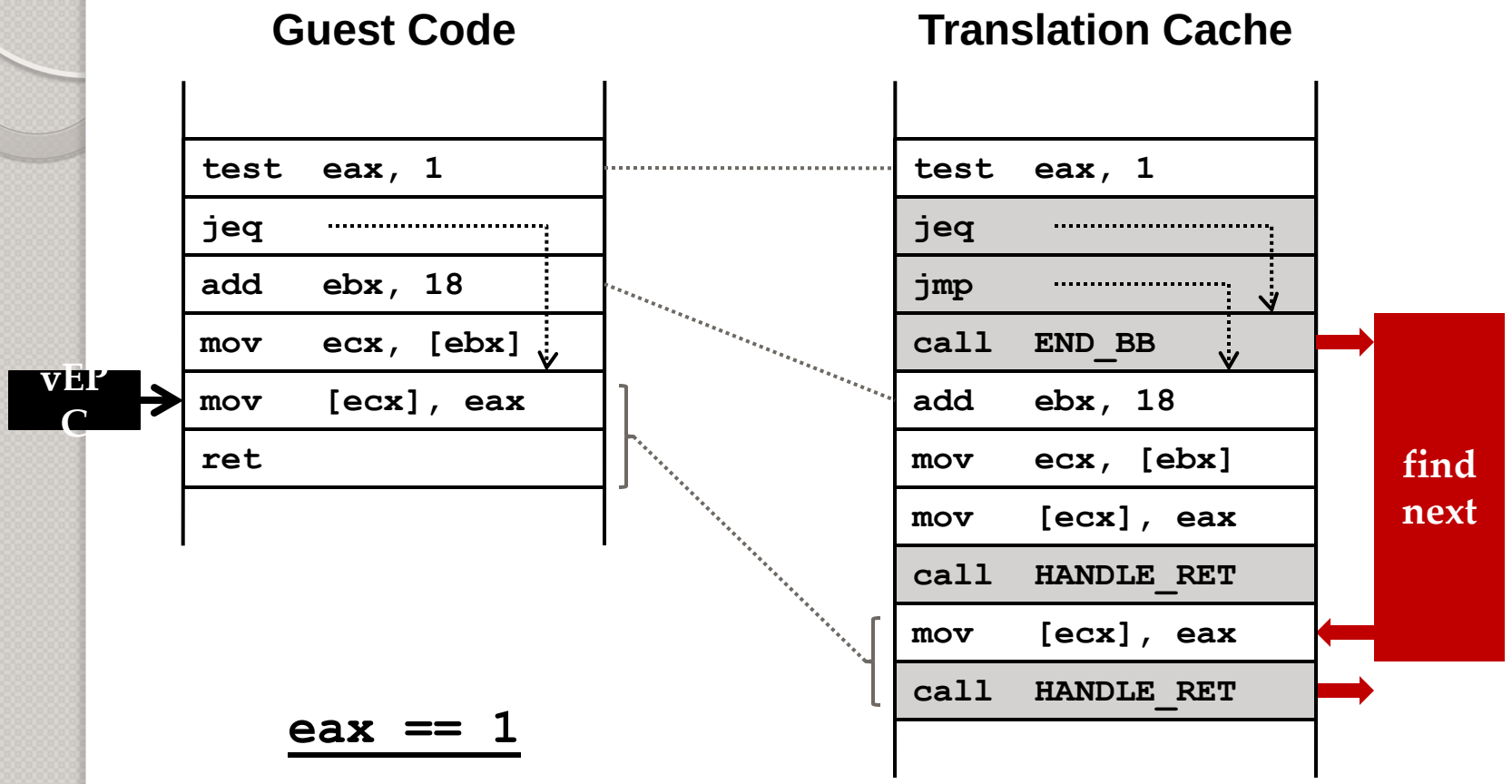
Controlling Control Flow



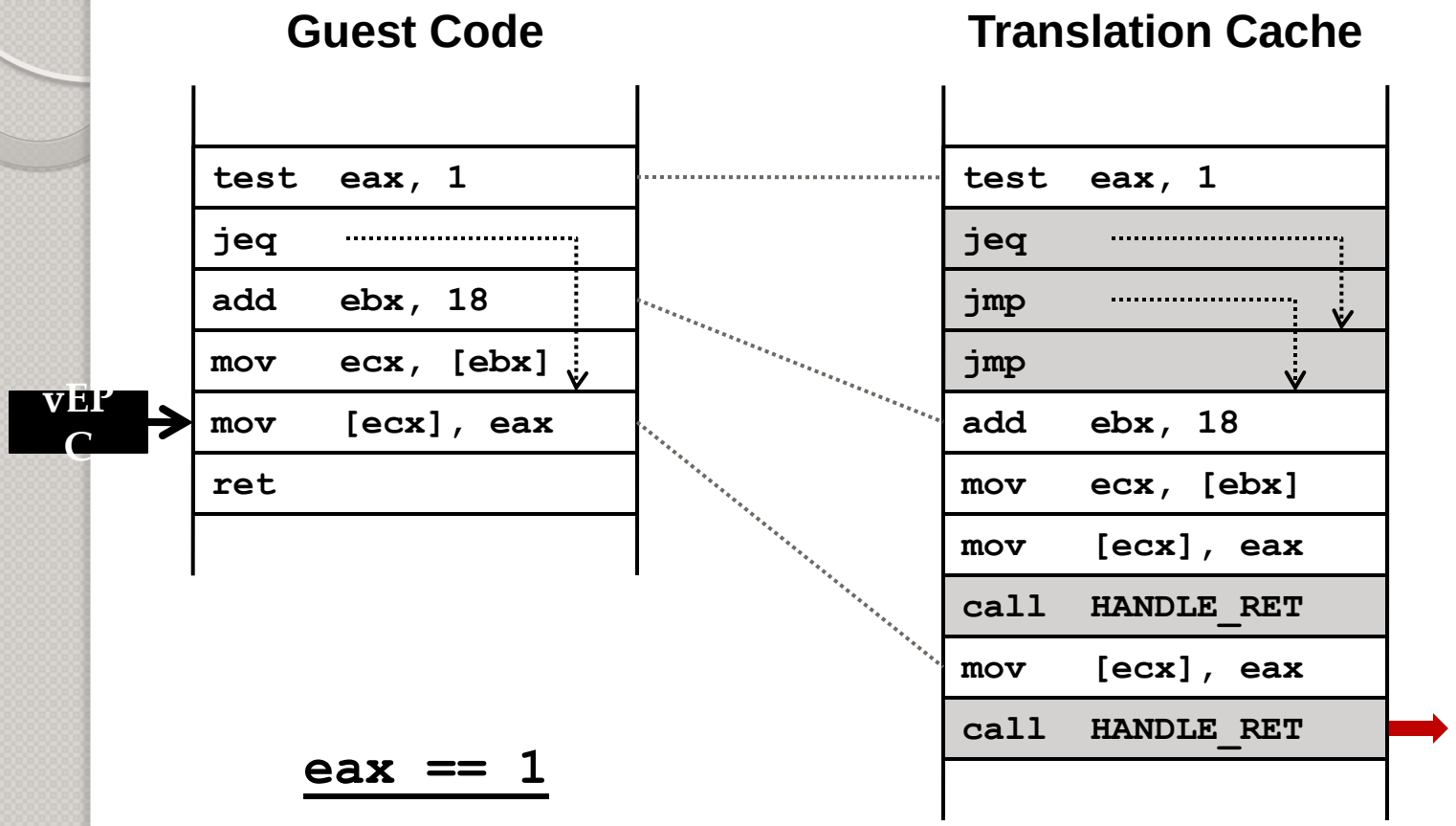
Controlling Control Flow



Controlling Control Flow



Controlling Control Flow



Issues with Binary Translation

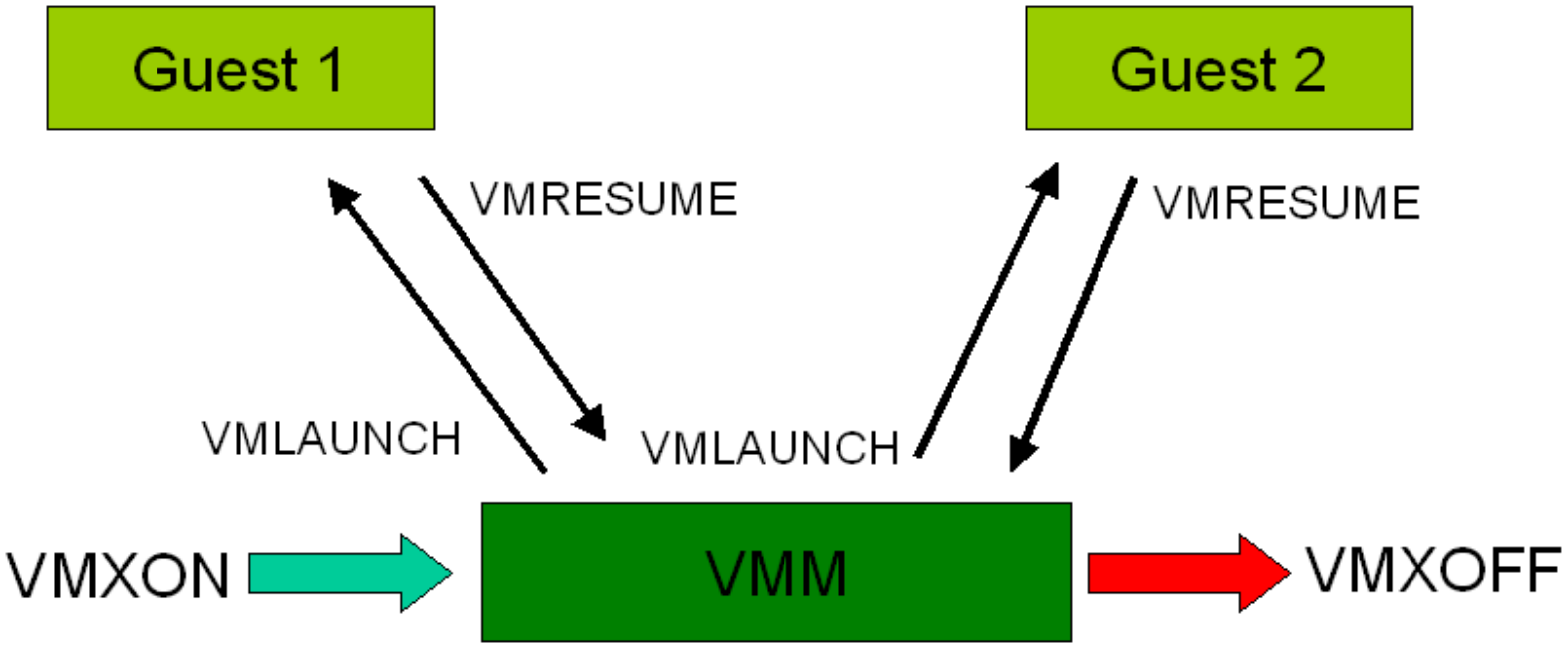
- Translation cache index data structure
- PC Synchronization on interrupts
- Performance Overheads.

Virtualization Support In CPUs

- Software like VMWare are available then why do we need hardware support?
- The advantage is that CPUs with virtualization have some new instructions to control virtualization.
- Moreover the design of VMMs/ Hypervisors would be simpler.
- Improved performance.

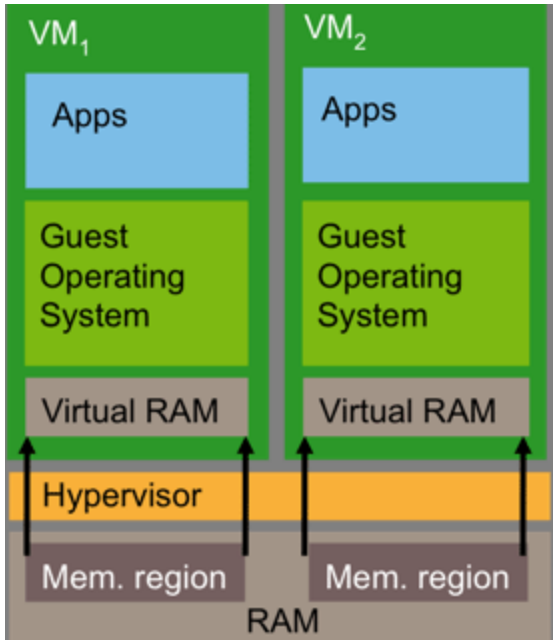
How It Works..?

- Processors with virtualization technology have extra instruction set called virtual machine extensions or VMX.
- There are two modes to run under virtualization: root operation and non-root operation. Usually only the virtualization controlling software, called Virtual Machine Monitor (VMM), runs under root operation, while operating systems running on top of the virtual machines run under non-root operation. Software running on top of virtual machines is also called ”guest software“.
- To enter virtualization mode, the software should execute the VMXON instruction and then call the VMM software. Then VMM software can enter each virtual machine using the VMLAUNCH instruction, and exit it by using the VMRESUME. If VMM wants to shutdown and exit virtualization mode, it executes the VMXOFF instruction.


Guest 1
Guest 2
VMRESUME
VMRESUME
VMLAUNCH
VMLAUNCH
VMXON
VMM
VMXOFF

Memory Virtualization

- Guest OS sees flat "physical" address space.
- Page tables within guest OS:
  - Translate from virtual to physical addresses.
- Second-level mapping:
  - Physical addresses to machine addresses.
- VMM can swap a VM's pages to disk.

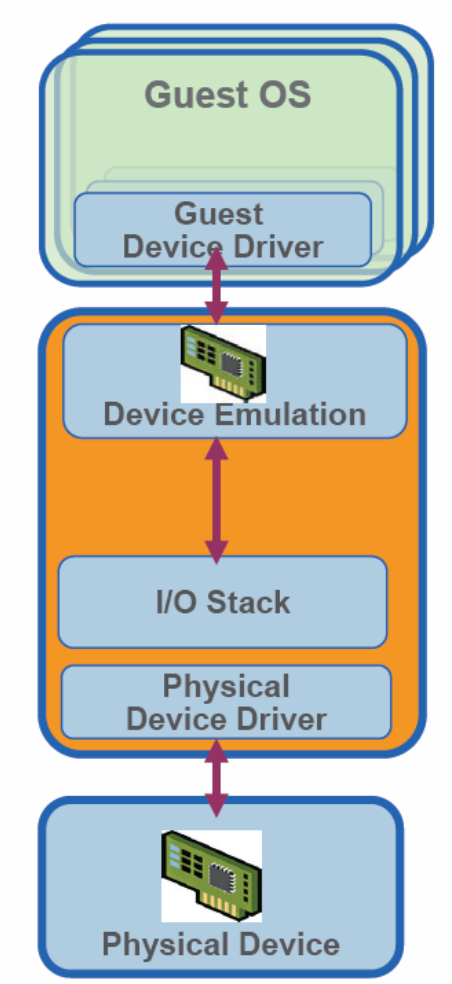


Memory Virtualization

Memory Virtualization

- Traditional way is to have the VMM maintain a shadow of the VM's page table
- The shadow page table controls which pages of machine memory are assigned to a given VM
- When OS updates it's page table, VMM updates the shadow

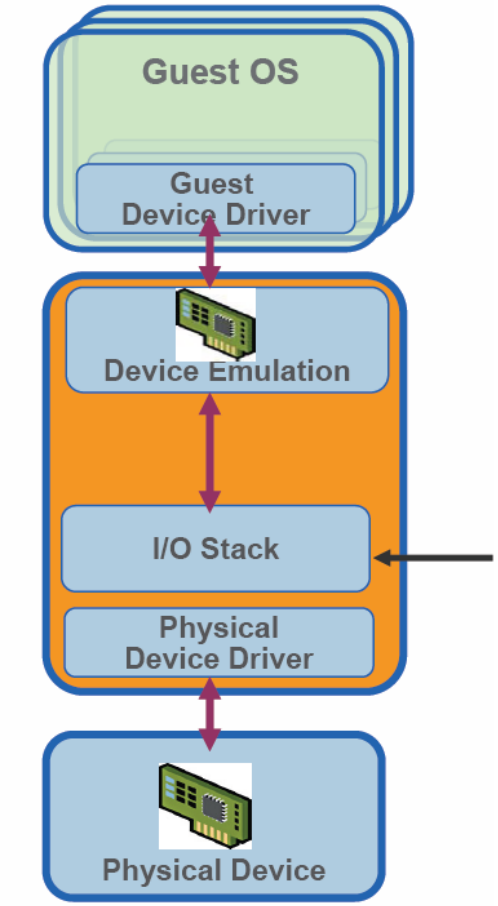
I/O Virtualization



I/O Virtualization architecture consists of

- Guest driver
- Virtual device
- Communication mechanism between virtual device and virtualization stack
- Virtualization I/O stack
- Physical device driver
- Real device

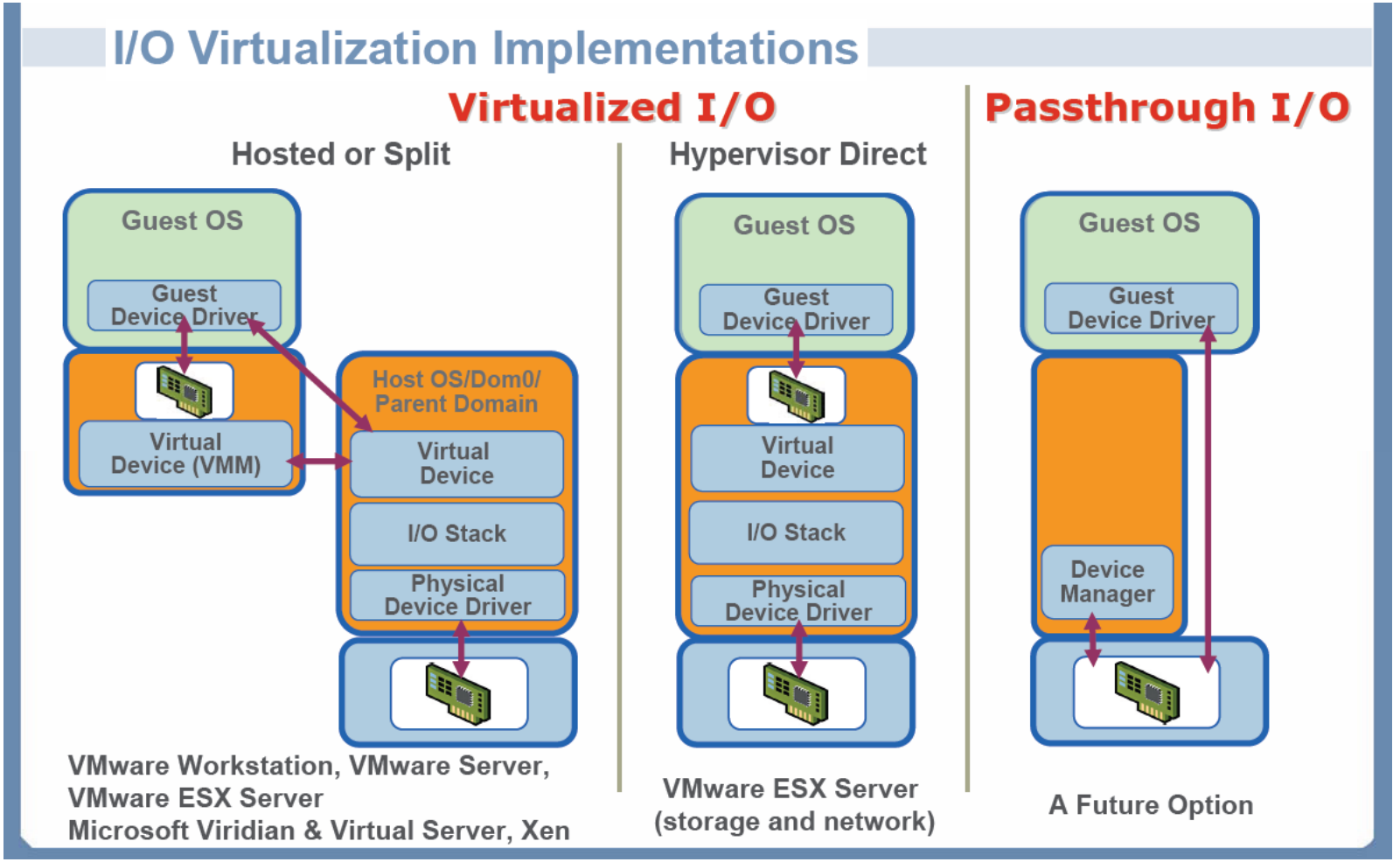
I/O Virtualization(Contd..)



Virtualization I/O stack

- Translates guest I/O addresses to host addresses
- Handles inter Vm communication
- Multiplexes I/O requests from/to the physical device
- Provides enterprise-class I/O features to the Guest

I/O Virtualization(Contd..)



Scheduling

- **BVT Scheduler:** Virtual Machines are scheduled as per their accumulated running time with a weighting factor that allows some VMs to be given preference over others. When a VM needs to run quickly , time is borrowed from its future use by subtracting a warp factor from its accumulated time.
- **SEDF Scheduler:** The deadlines are set according to a per domain parameter called period. At the end of each time slice, the next deadline is set to current time plus period. The period can be used to determine the share of the CPU each domain gets.

Open Virtualization Format

- OVF enables efficient, flexible, and secure distribution of enterprise software, facilitating the mobility of virtual machines and giving customers vendor and platform independence.
- Customers can deploy an OVF formatted virtual machine on the virtualization platform of their choice.
- The proposed format accepted by Distributed Management Task Force(DMTF) uses existing packaging tools to combine one or more VM together with a standards-based XML wrapper that provides the virtualization platform -- from VMware, Microsoft, Citrix, or others -- a portable package, which includes installation and configuration parameters for the VMs.
- The OVF could also help IT managers understand how virtual machines have been changed throughout their lifecycle. For instance, if a VM template is cloned and that clone has changed from the master template, IT managers need to know what has changed to be able to troubleshoot performance problems on the VM.

Case Study

- Qualcomm is an global leader in providing high-value wireless data solutions. The company pioneered Code Division Multiple Access (CDMA) technology, and their Network Management Center processes more than 7M transactions/day.
- Qualcomm started a Server Consolidation Project in the first half of 2003. Today, 60% of Qualcomm's x86 environment is virtualized (1900 total servers/1150 are virtualized). The number of physical servers has grown from 950 to 1900 over the past 2.5 years, and because of the much simplified provisioning with virtualization, they have been able to maintain the same number of server admin's today. They provision 68 new VM's/month. This would be impossible in the physical world without dramatic staffing increases. Which means that the number of physical servers a single sys admin can manage has more than doubled. This translates into substantial operational savings for the company.
- In aggregate, they've saved $4.5M over 3 yrs with VMware. This calculation doesn't include the additional cost for storage in the virtual world (all VM's are SAN connected now), but it also doesn't include the cost savings from power, cooling etc.

Why not to virtualize?

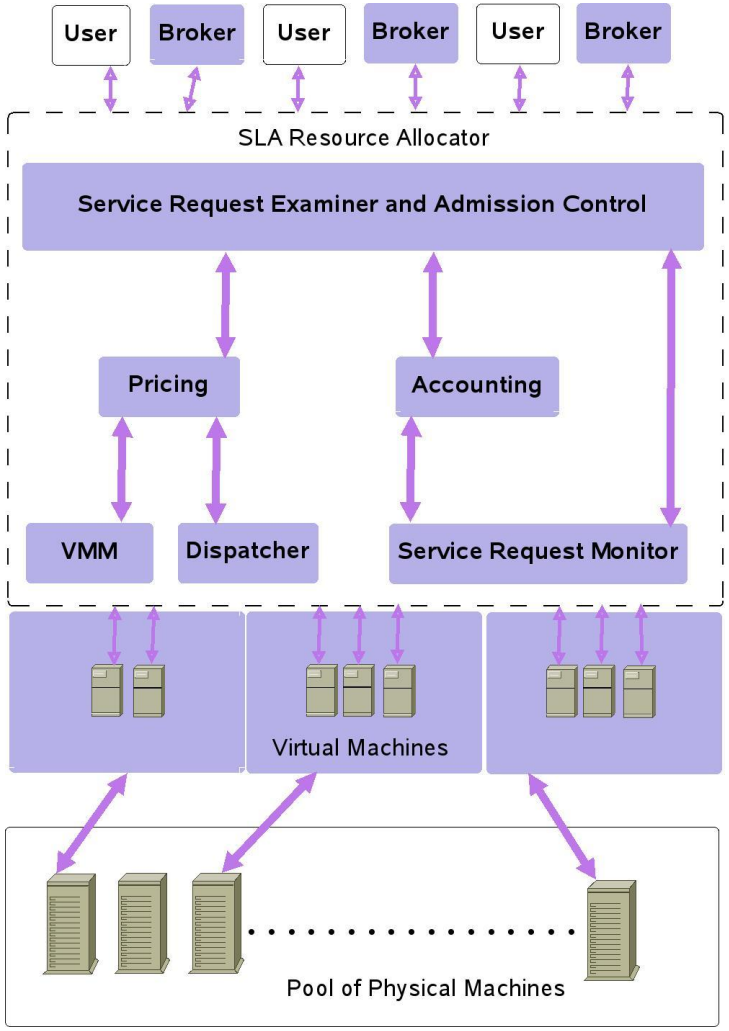
- **Overhead:** Performance is often being compromised due to flexibility
- **Single point of failure:** Even though the virtual machine is decoupled from the hardware, it is still dependent on the hardware working.

Virtualization in cloud

"A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers."

Open Source IaaS Cloud Platforms
OpenNebula, Haizea, XenCloud, and Eucalyptus etc.

Open Source VMMs
Xen, KVM, Linux VServer, and UML etc.



Architecture of cloud computing

OpenNebula And Haizea

- OpenNebula is Virtual Infrastructure (VI) software toolkit, which is used to control a VMs lifecycle. It manages the VM image and storage, the network fabric (such as DHCP) services to tie in VMs with the environment, and hypervisors which create and control the VM. It can deploy groups of virtual machines to be treated as a single unit.
- Haizea can be used to extend OpenNebula's scheduling capabilities, allowing it to support advance reservation of resources and queueing of best effort requests. OpenNebula and Haizea complement each other, since OpenNebula provides all the enactment muscle (OpenNebula can manage Xen, KVM, and VMWare VMs on a cluster) and Haizea provides the scheduling brains.

References

1. “Virtualization and Cloud Computing” , Norman Wilde and Thomas Huber, University Of West Florida.
2. “Introduction to virtualization: Get started with ESXi”, Vmware .
3. “Virtual Machine Monitors” , Dr. Marc E. Fiuczynski, Princeton University.
4. “Virtualization In Linux”, Atul Bansal, Manish Pal, Pulkit Gambhir.
5. “Virtual Machines : CPU Virtualization”, Scott Devine, VMWare Inc.
6. “Virtual Machines : Device Virtualization”, Scott Devine, VMWare Inc.

References (Cont…)

7. "I/O Architectures for Virtualization", Mallik Mahalingam, VMWare Inc.
8. "Virtual Machines : Memory Virtualization", Scott Devine, VMWare Inc.
9. "Resource Management for Virtualized Systems ", Carl Waldspurger, VMWare Inc.
10. "Cloud Computing: Concepts And Applications", Prof Sanjay Chaudhary, DA-IICT
11. "Intel Virtualization Technology (VT) Explained", available at http://www.hardwaresecrets.com/printpage/263
12. "Open Virtualization Format for Virtual Machines", available at http://www.vmware.com/appliances/getting-started/learn/ovf.html