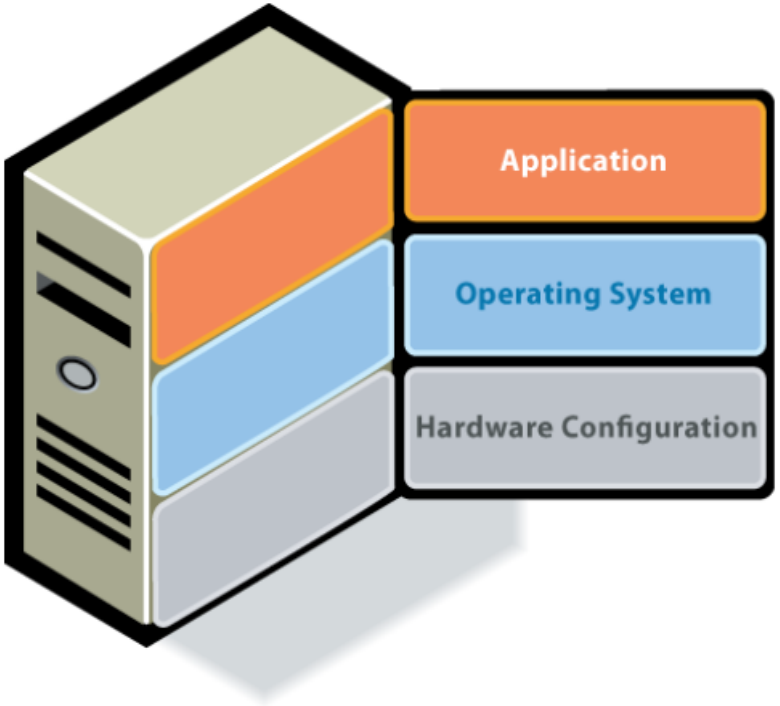
Virtualization Overview

IIT Jodhpur - Winter course – Dec 2012

Prem Karat (prem.karat@linux.vnet.ibm.com)
Linux Technology Center, India

The Challenge
Virtualization Technology Overview



Old Model:
Traditional x86 Architecture

- Single OS image per machine
- Software and hardware tightly coupled
- Multiple applications often conflict
- Underutilized resources

→ Old model is challenging!

State of Infrastructure Today – Physical

Server Sprawl

- 38 m physical servers by 2010 - 700% increase in 15 years
- $140 bn in excess server capacity - a 3-year supply

Power & Cooling

- 50c for every $1 spent on servers
- $29 bn in power and cooling industry wide

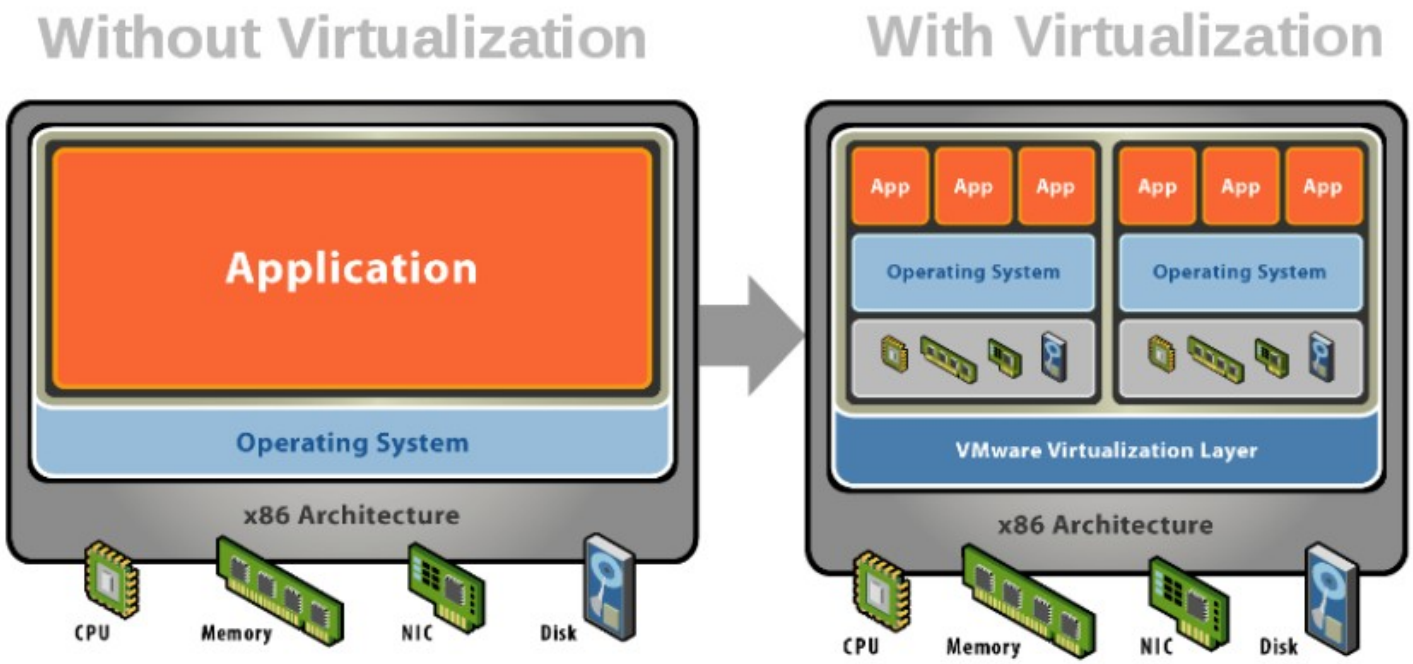
Space Crunch

- $1,000 /sqft
- $2,400 / server
- $40,000 / rack

Operating Cost

- $8 in maintenance for every $1 spent on new infrastructure
- 20-30 : 1 server-to-admin ratio

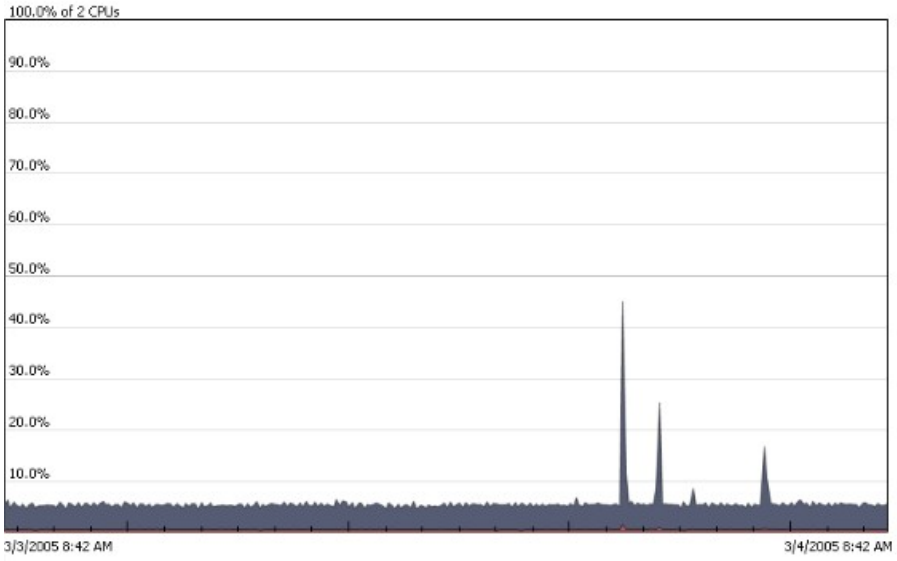
What is Virtualization?



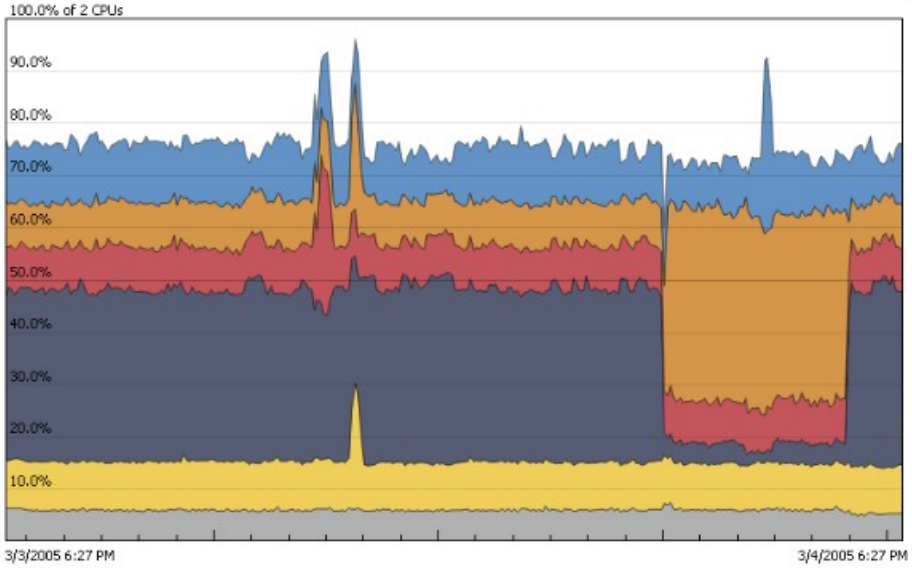
- What's in a hypervisor
  - I/O Stack
  - Device drivers
  - Platform support code
  - Resource management
  - Process scheduling
  - Memory manager
  - Security manager
  - Virtual Machine Monitor

Virtualization Increases Hardware Utilization

Before

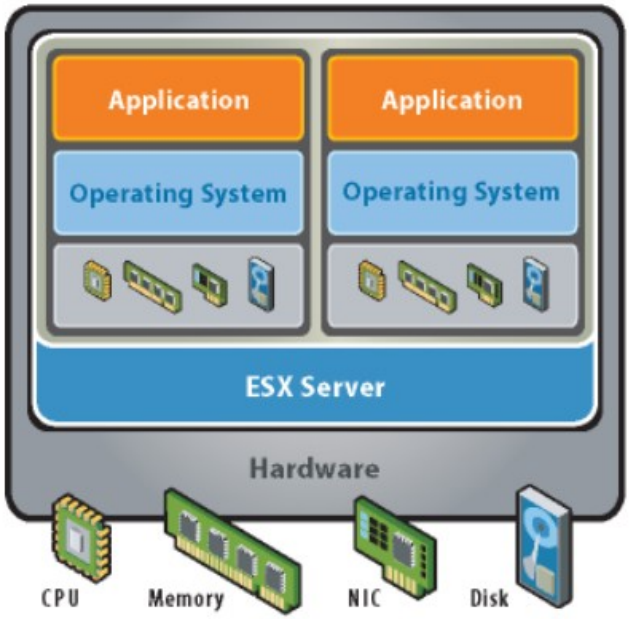


After



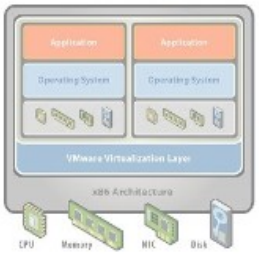
Virtualization enables consolidation of workloads from underutilized servers onto a single server to safely achieve higher utilization

Key Properties of Virtual Machines



- **Partitioning**
  - Run multiple operating systems on one physical machine
  - Divide system resources between virtual machines

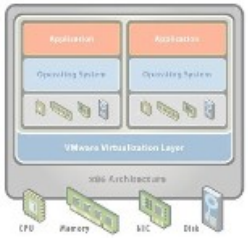
Key Properties of Virtual Machines



- **Partitioning**
  - Run multiple operating systems on one physical machine
  - Divide system resources between virtual machines

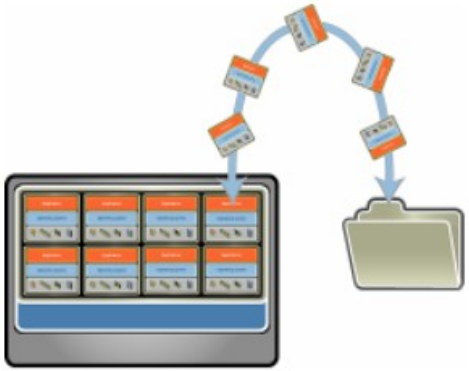
- **Isolation**
  - Fault and security isolation at the hardware level
  - Advanced resource controls preserve performance

Key Properties of Virtual Machines



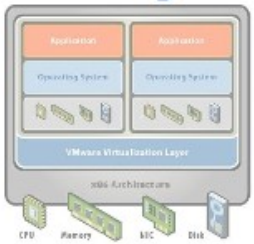
- **Partitioning**
  - Run multiple operating systems on one physical machine
  - Divide system resources between virtual machines

- **Isolation**
  - Fault and security isolation at the hardware level
  - Advanced resource controls preserve performance



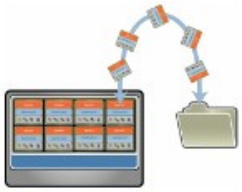
- **Encapsulation**
  - Entire state of the virtual machine can be saved to files
  - Move and copy virtual machines as easily as moving and copying files

Key Properties of Virtual Machines

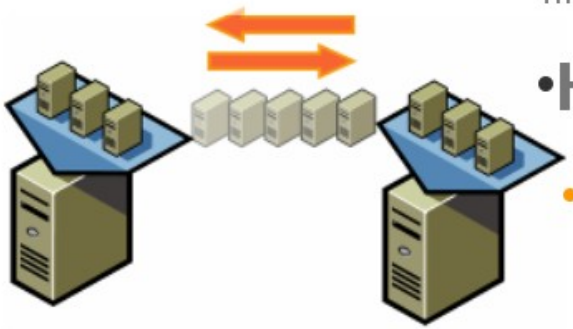


- **Partitioning**
  - Run multiple operating systems on one physical machine
  - Divide system resources between virtual machines

- **Isolation**
  - Fault and security isolation at the hardware level
  - Advanced resource controls preserve performance

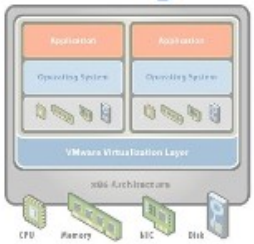


- **Encapsulation**
  - Entire state of the virtual machine can be saved to files
  - Move and copy virtual machines as easily as moving and copying files



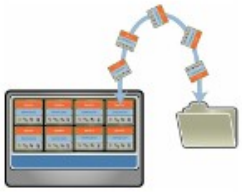
- **Hardware-Independence**
  - Provision or migrate any virtual machine to any similar or different physical server

Key Properties of Virtual Machines

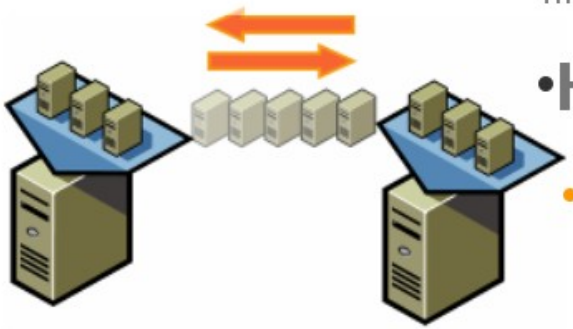


- **Partitioning**
  - Run multiple operating systems on one physical machine
  - Divide system resources between virtual machines

- **Isolation**
  - Fault and security isolation at the hardware level
  - Advanced resource controls preserve performance



- **Encapsulation**
  - Entire state of the virtual machine can be saved to files
  - Move and copy virtual machines as easily as moving and copying files



- **Hardware-Independence**
  - Provision or migrate any virtual machine to any similar or different physical server

Challenges in Virtualization

- **How to Virtualize CPU?**
- **How to Virutalize Memory?**
- **How to Virtualize IO?**

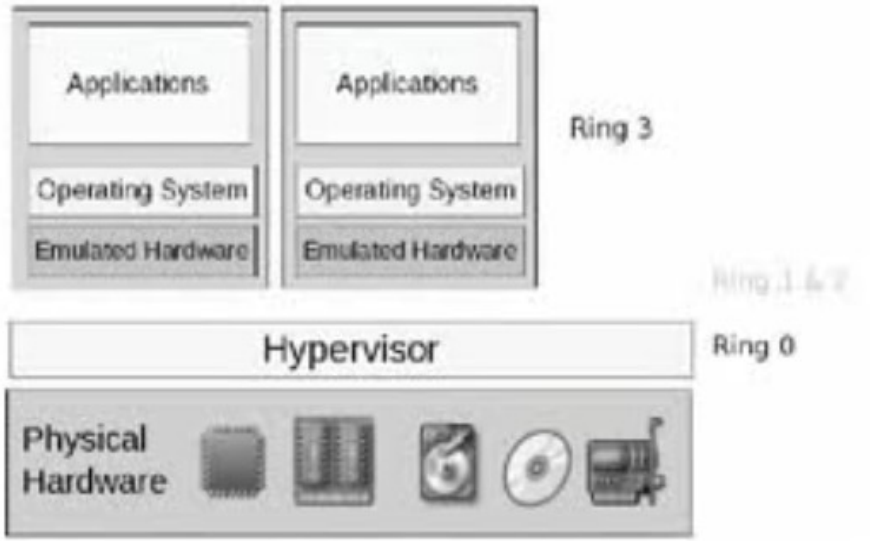
Virtualizing the X86 Architecture

- x86 architecture is difficult to virtualize
- CPU implements 4 privilege levels or “rings” - 0 thru 3
  - Privileged kernels calls run in ring 0
  - Applications / userspace run in ring 3

Virtualizing the X86 Architecture

- Hypervisor runs in ring 0
- Virtual machines run in ring 3

Operating system in Virtual machine makes calls privileged instructions
Will cause a machine fault



Virtualizing the X86 Architecture

- To virtualize x86 operating systems we need to handle privileged calls
  - "Ring compression" or "de-privileging"
- Four techniques for virtualizing x86 platform
  - Full emulation
  - Binary Translation
  - Paravirtualization
  - Hardware Assisted Virtualization

Virtualizing the X86 Architecture

- Full Emulation

  Emulate the entire machine, including CPU using software

  - Yields very more performance
  - Rarely used due to large overhead
  - Some niche use cases

    eg. Running x86 software on PPC platforms

Virtualizing the X86 Architecture

- Binary Translation

  On the fly translation of privileged kernel instructions
  - Unprivileged instructions run directly on CPU
  - Hypervisor reads ahead and re-writes privileged instructions
  - Redirects calls to the hypervisor
  - Also known as “Scan Before Executing”

Virtualizing the X86 Architecture

- Paravirtualization

Modify guest operating system to talk directly to the hypervisor

- Guest OS kernel is modified to remove privileged instructions
  - Replaced with direct calls to the hypervisor
- Advantages
  - Improved IO and resource scheduling -> Improved performance
- Disadvantages
  - Requires changes to the guest operating system -> new OS Kernel
    - Another kernel/OS to test and certify

Virtualizing the X86 Architecture

- Hardware Assisted Virtualization

  Extensions to x86 architecture to support virtualization

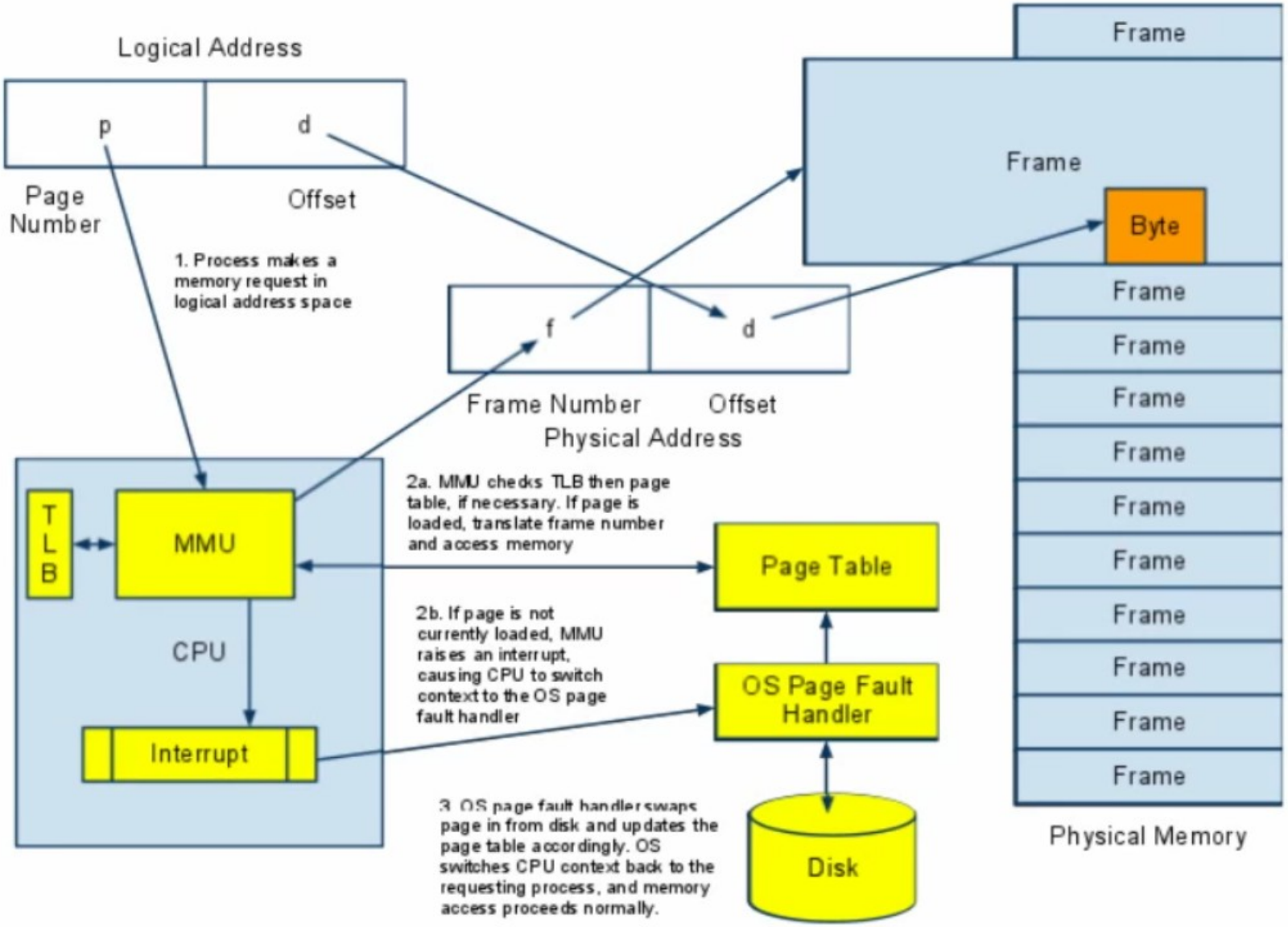
  - AMD AMD-V
  - intel VT-X
  - Available since 2006, now in all mainstream platforms

- Offloads “Ring compression” to CPU
  - Effectively provides new privilege level
  - Used by hypervisor to help trap and handle privileged instructions

Hardware Assisted Virtualization

- First generation – CPU Virtualization
- Second generation – Memory Management
  - Offloads memory page table management to CPU / chipset
  - Provides significant performance improvement
  - Intel : Extended Page Tables (EPT)
  - AMD : Rapid Virtualization Indexing (RVI) - previously called NPT

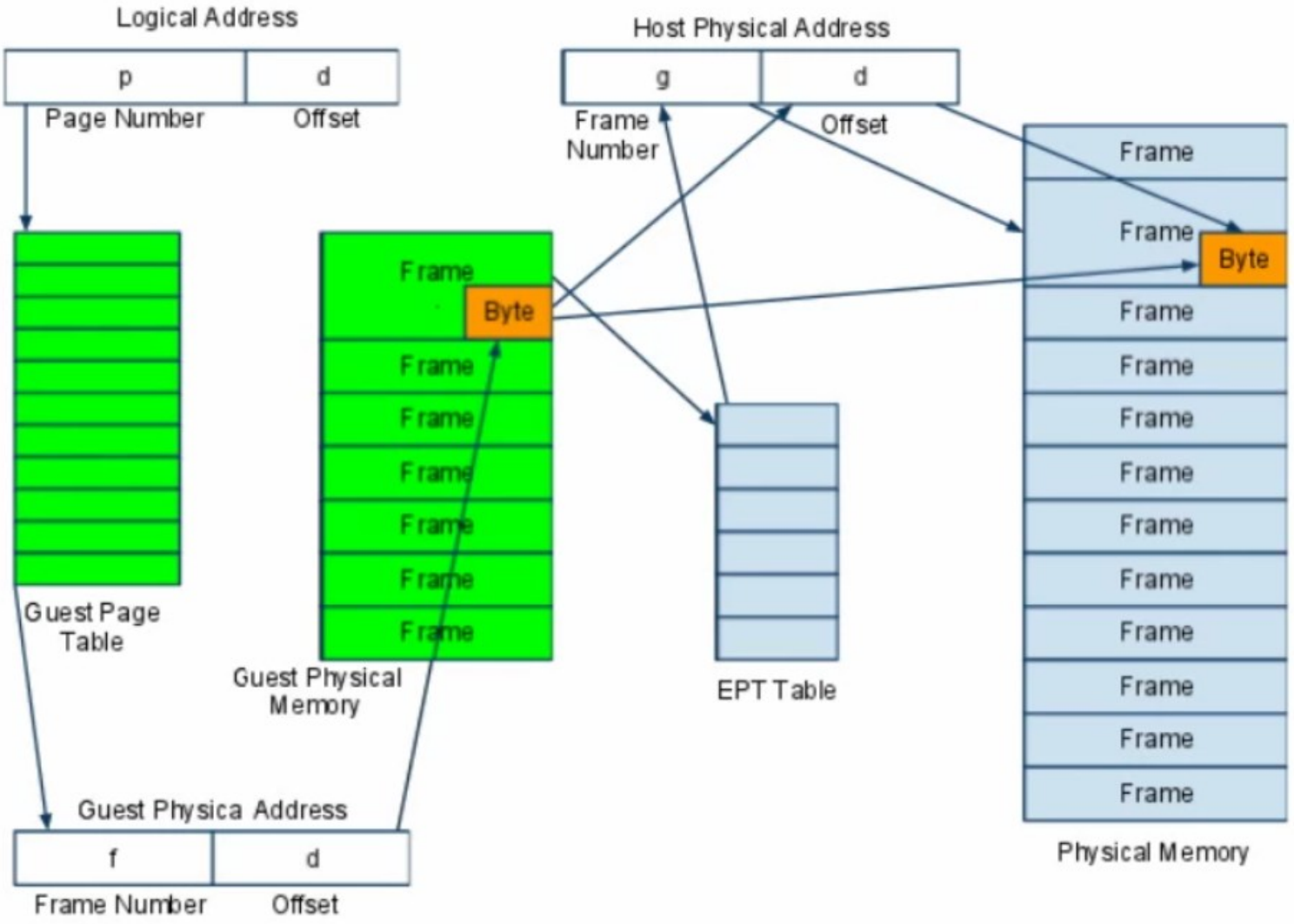
Page Fault Handling



Extended Page Tables (EPT)

- Intel virtualization extension
  - AMD equivalent is Rapid Virtualization Indexing (RVI)
    - Formerly Nested Page Tables (NPT)
- Two levels of paging
  - First level translates pages to guest frame numbers
  - Second level (EPT table) translates guest frame numbers to physical frame numbers
- Enables each concurrent virtual machine to manage its own memory efficiently, without having to invoke the hypervisor to perform page mapping

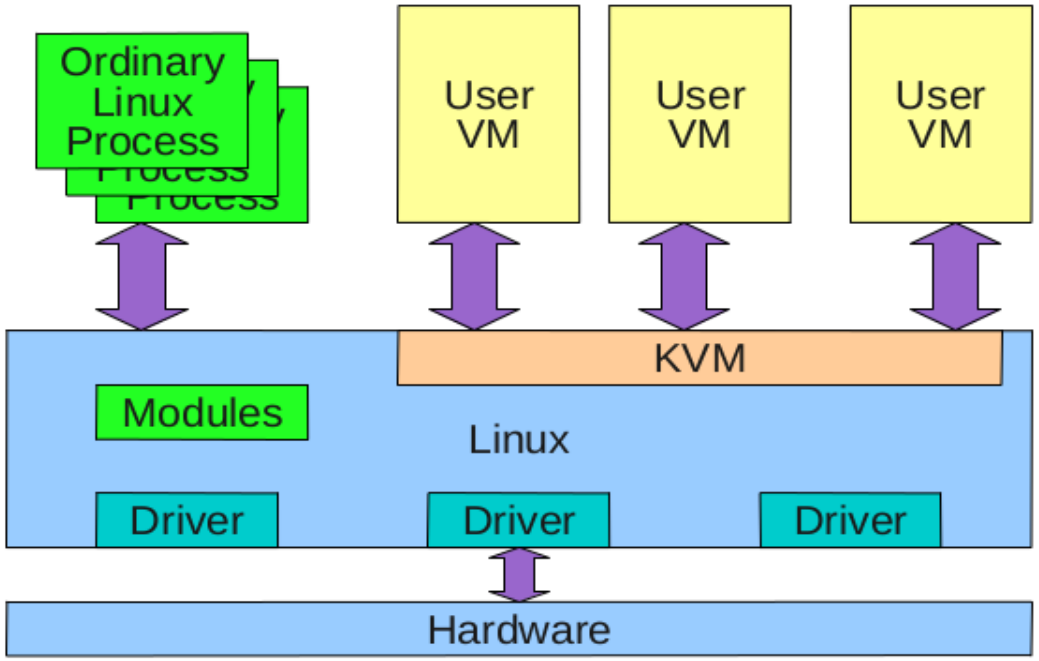
EPT Translation



Hardware Assisted Virtualization

- First generation – CPU Virtualization
- Second generation – Memory Management
  - Offloads memory page table management to CPU / chipset
  - Provides significant performance improvement
  - Intel : Extended Page Tables (EPT)
  - AMD : Rapid Virtualization Indexing (RVI) - previously called NPT
- Third generation – I/O Offload
  - Secure PCI Pass-through (Intel VT-D, AMD IOMMU)
  - Single Root I/O Virtualization – SR/IOV
    - Allows physical PCI devices to be split into multiple virtual devices
    - Allows single PCI device to be passed through to multiple virtual machines

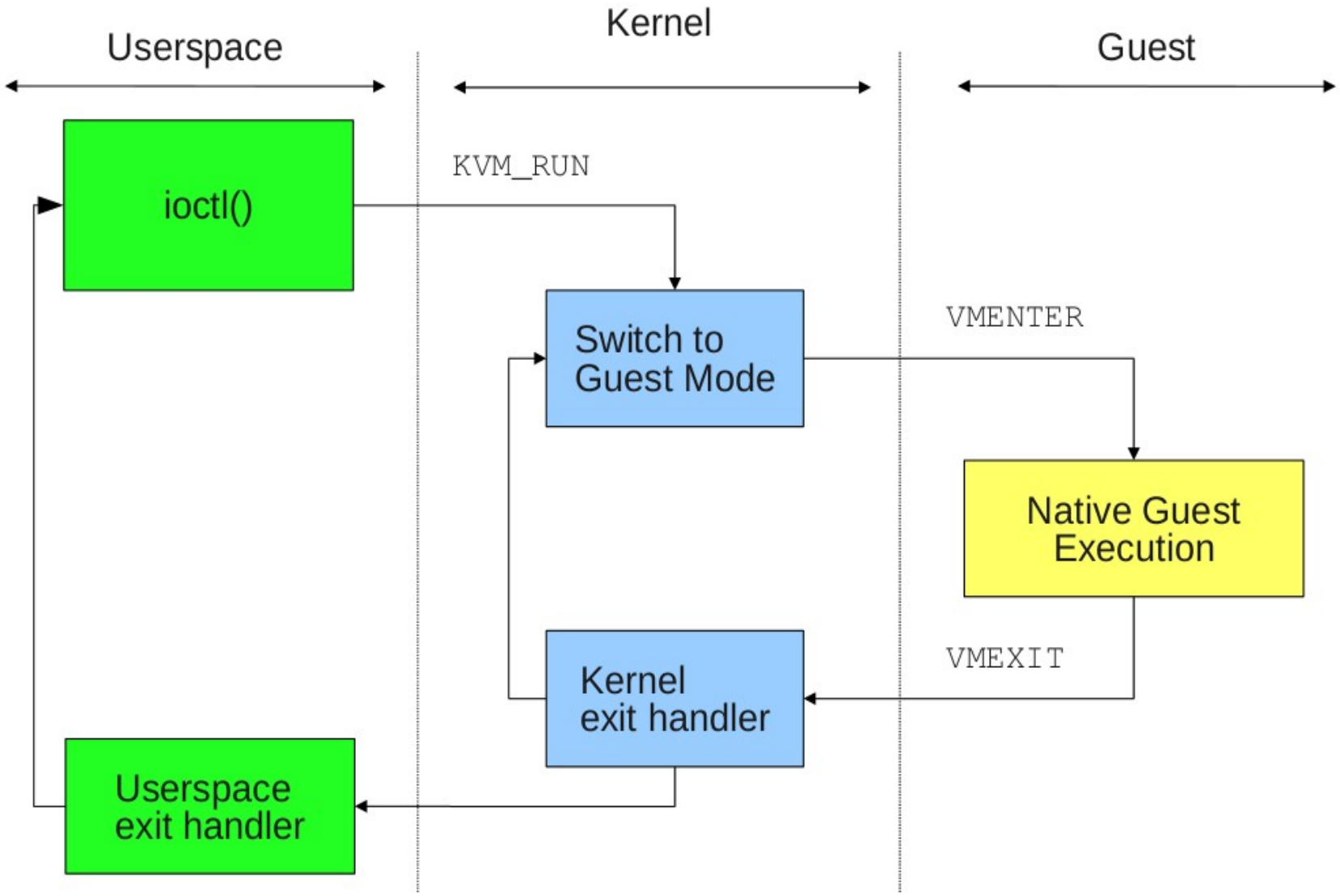
Quick Overview - KVM Architecture



Quick Overview – KVM Architecture

- Guests run as a process in userspace on the host
- Guests inherits features from the kernel (NUMA, huge pages, support for new hardware)
- Disk and Network IO through host (most of the time)
  - IO settings in host can make a big difference in guest IO performance
  - Need to understand host buffer caching
    - Proper settings to achieve true direct IO from the guest
    - Deadline scheduler (on host) typically gives best performance
- Network typically goes through a software bridge
- Device assignment can help with network performance

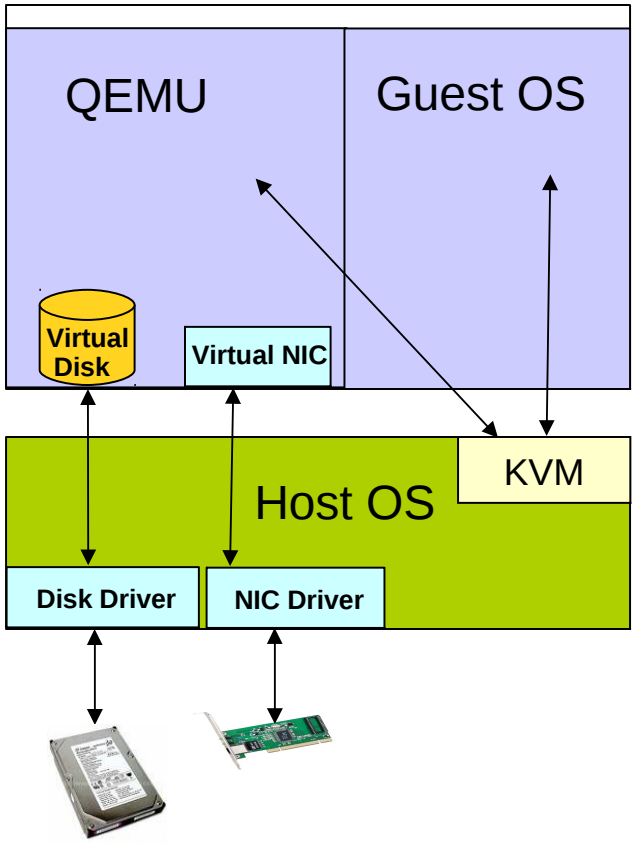
KVM Execution Model



KVM Execution model

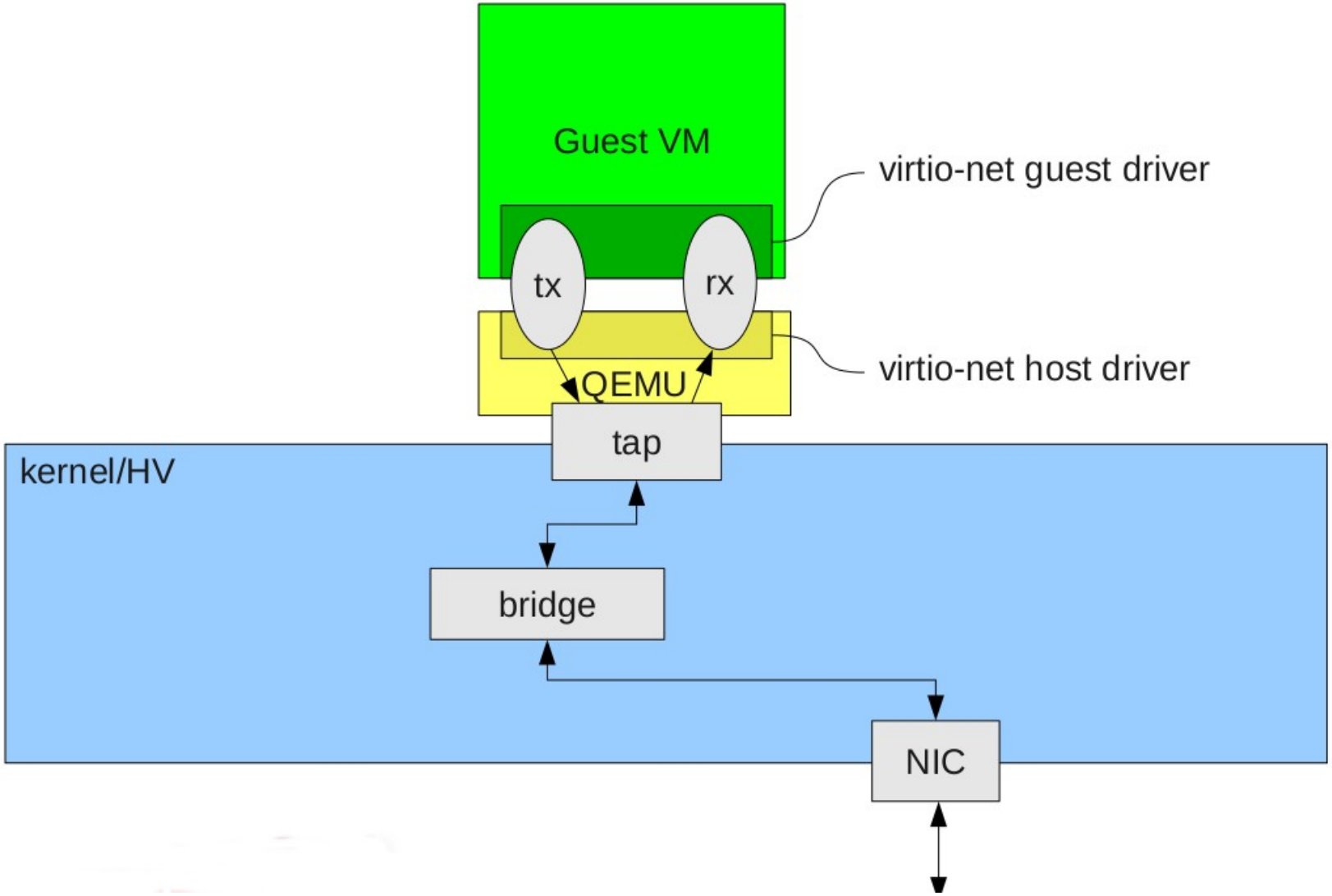
- Three modes for thread execution instead of the traditional two:
  - User mode
  - Kernel mode
  - Guest mode
- A virtual CPU is implemented using a Linux thread
  - The Linux scheduler is responsible for scheduling a virtual CPU, as it is a normal thread
- Understanding these help when tuning

Virtualization Phase - KVM Acceleration

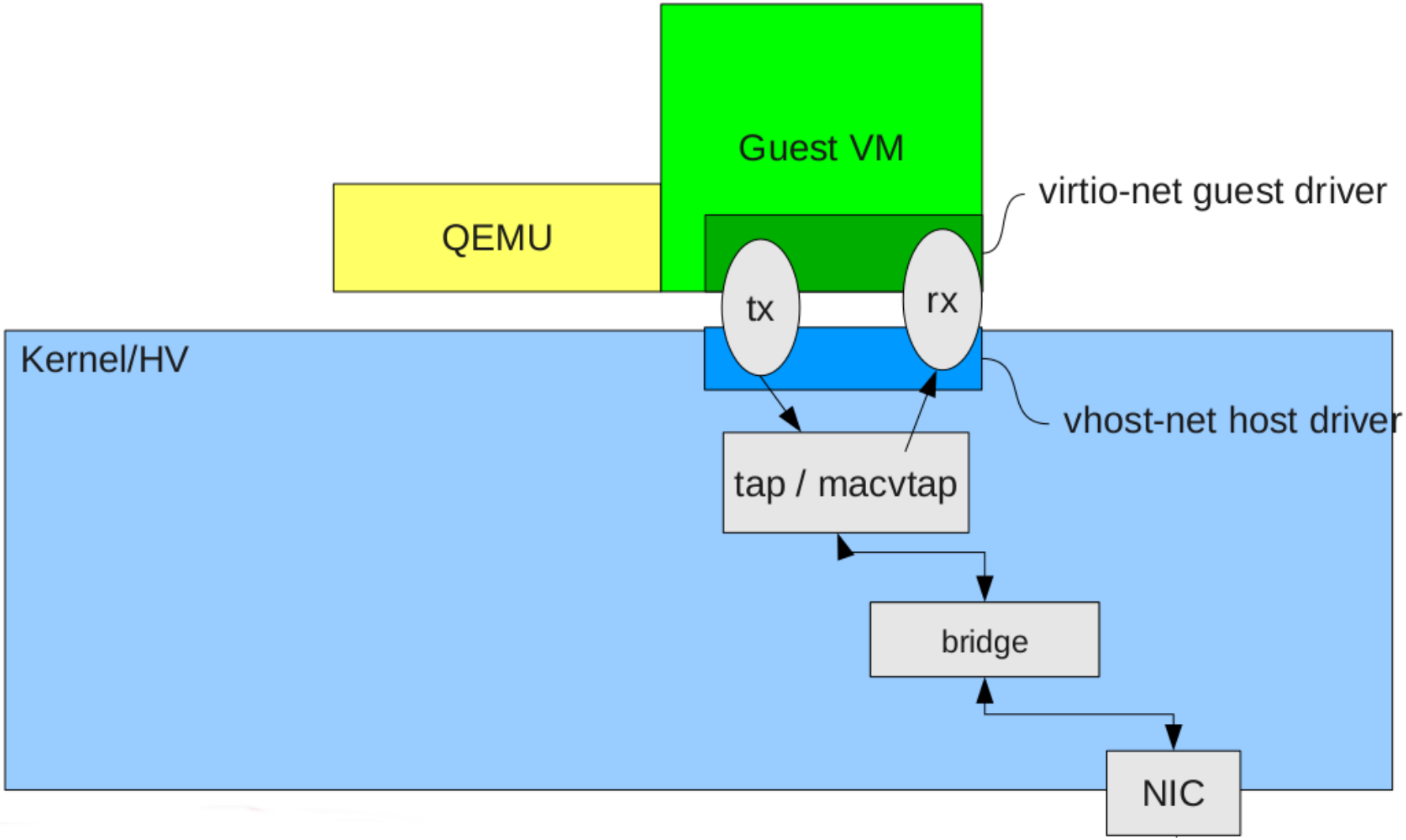


- Qemu emulates devices and runs in User Mode
- Guest still part of the QEMU process
- Guest image run in Guest Mode facilitated by KVM
- KVM exploits Intel VT / AMD-V CPU support
- Performance
  - Guest CPU speed is near native
  - IO is slow
- Guest->Host->User mode and vice-versa
- context switch penalty for each i/o operation

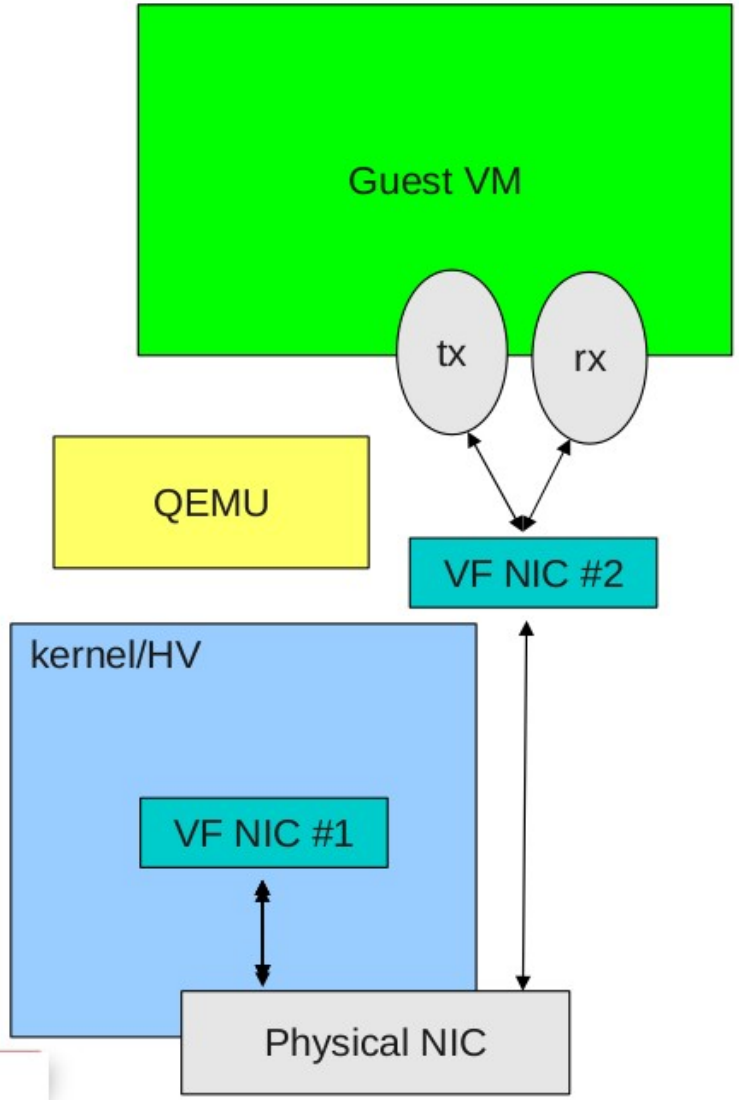
virtio network architecture



In-Kernel vhost-net architecture (RHEL6)



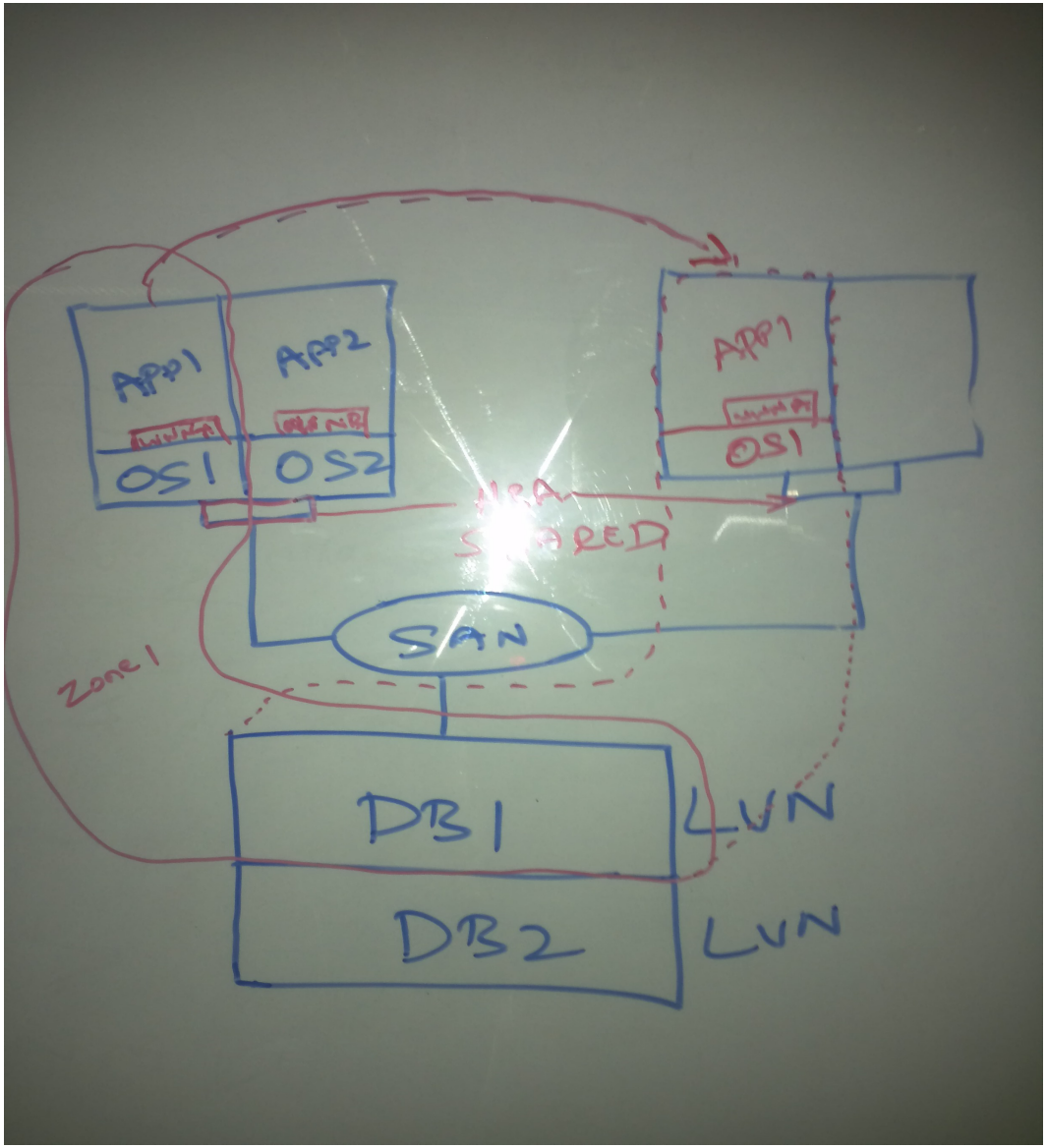
PCI device assignment network (vt-d/SR-IOV)



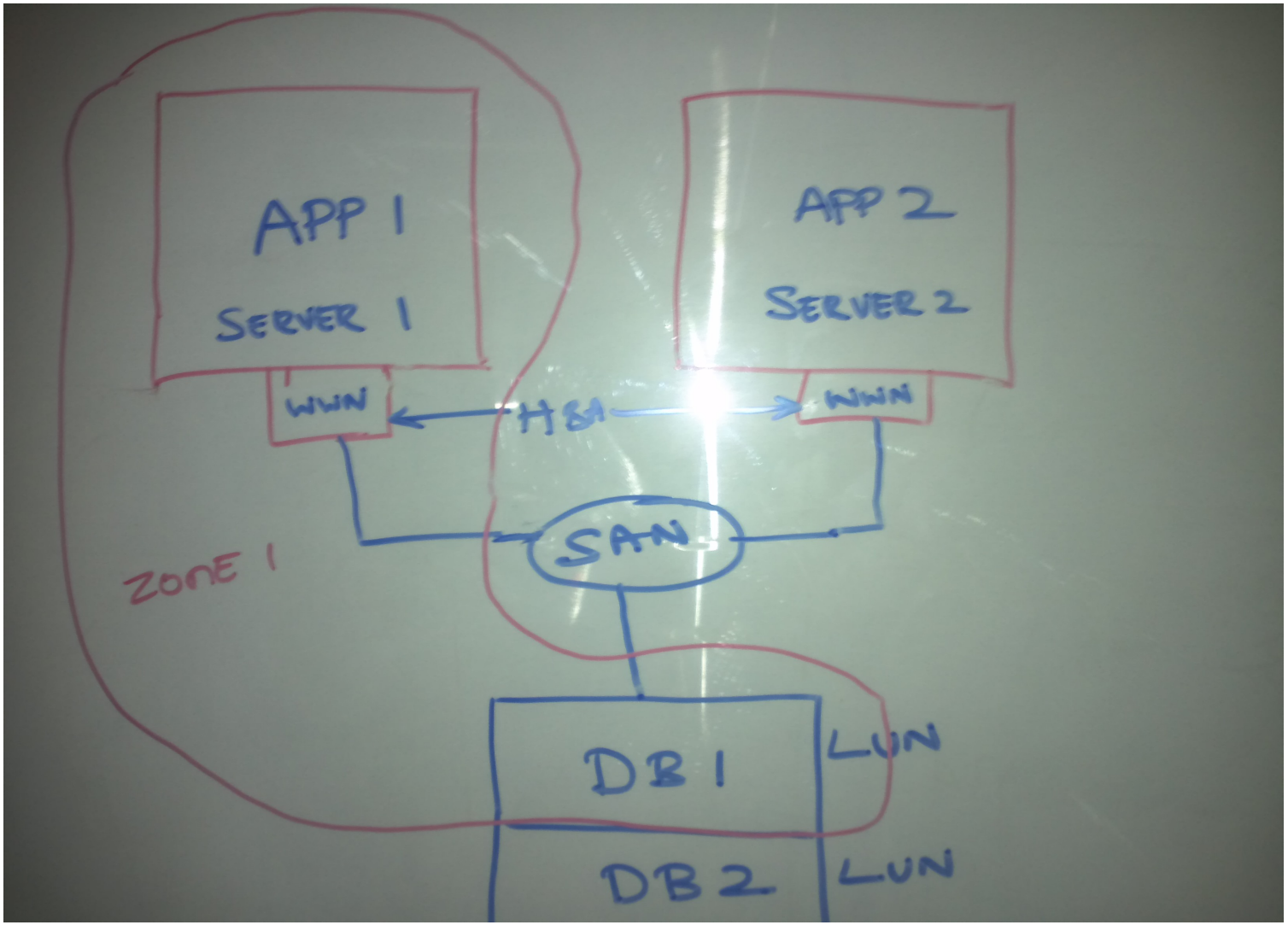
KVM & Storage

N Port-ID Virtualization (NPIV) in Action

NPIV technology provides virtual wwn to each guest and helps in sharing the HBA among guests

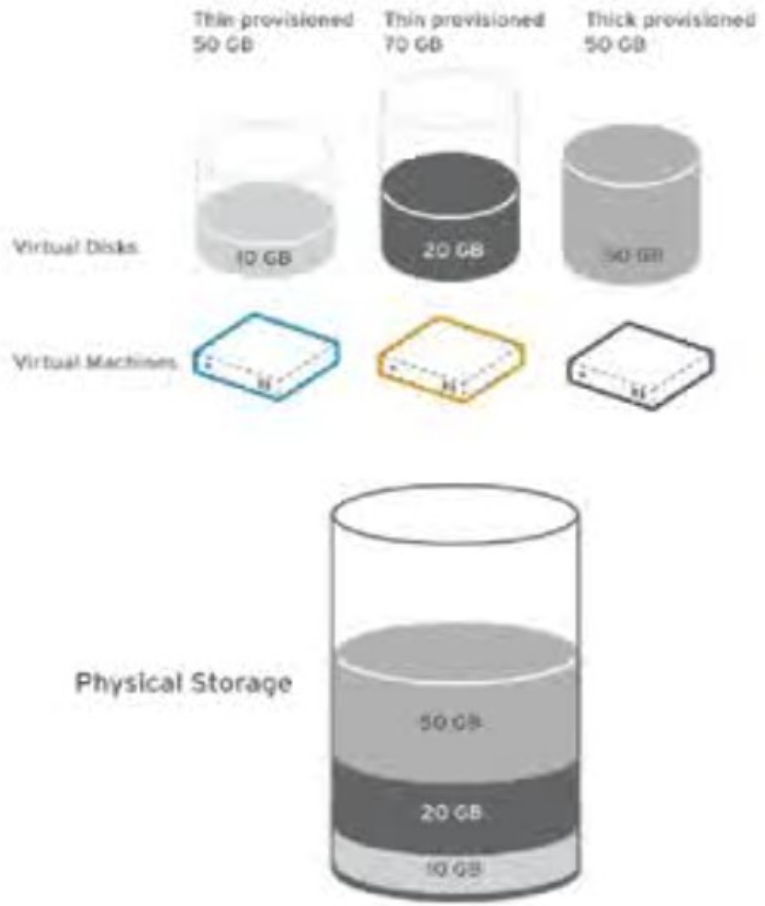


Isolated server and storage are bound together by assigning WWN to the HBA



Thin Provisioning

- Allocate storage only when needed
- Oversubscribe storage
- Transparent to virtual machine
- Improve Storage Utilization
- Reduced Storage Costs
- Works with NFS, iSCSI and Fiber Channel



Thank You! Questions?

References:

Redhat White paper on KVM Architecture

- Redhat JBOSS & Redhat Summit presentations