Overview

Device Model

Hardware Assistance

IO VIRTUALIZATION

IO Virtualization

- Goal :
  - Share or create IO devices for virtual machines.
- Two types of IO subsystem architecture :
  - Port Mapped IO
    - Port-mapped IO uses a special class of CPU instructions specifically for performing IO.
  - Memory Mapped IO (MMIO)
    - Memory Mapped IO uses the same address bus to address both memory and IO devices, and the CPU instructions used to access the memory are also used for accessing devices.
- Traditional IO techniques :
  - Direct memory Access (DMA)
  - PCI / PCI Express

Port Mapped IO

- IO devices are mapped into a separate address space
  - IO devices have a separate address space from general memory, either accomplished by an extra “IO" pin on the CPU's physical interface, or an entire bus dedicated to IO.
  - Generally found on Intel microprocessors, specifically the **`IN`** and **`OUT`** instructions which can read and write one to four bytes (**`outb`**, **`outw`**, **`outl`**) to an IO device.
- Pros & Cons
  - Pros
    - Less logic is needed to decode a discrete address.
    - Benefits for CPUs with limited addressing capability.
  - Cons
    - More instructions are required to accomplish the same task.
    - IO addressing space size is not flexible.

Memory Mapped IO

- IO devices are mapped into the system memory map along with RAM and ROM.
  - To access a hardware device, simply read or write to those 'special' addresses using the normal memory access instructions.

- Pros & Cons
  - Pros
    - Instructions which can access memory can be used to operate an IO device.
    - Operate on the data with fewer instructions.
  - Cons
    - Physical memory addressing space must be shared with IO devices.
    - The entire address bus must be fully decoded for every device.

Direct Memory Access

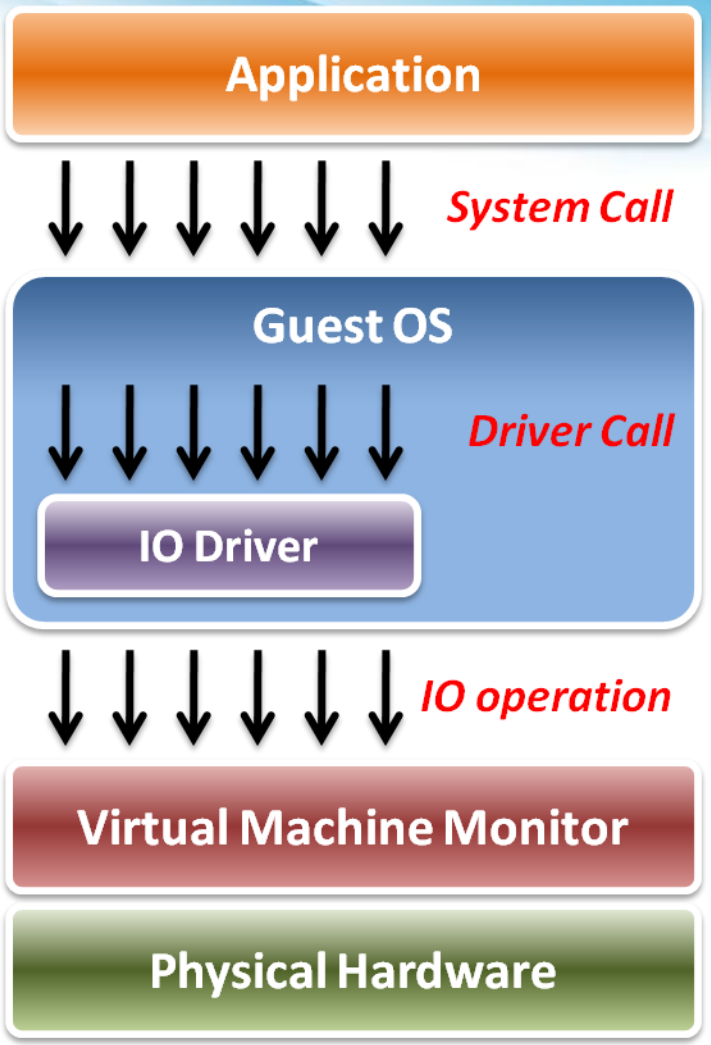
- What is DMA ?
  - Allow certain hardware subsystems within the computer to access system memory for reading and/or writing independently of the central processing unit.
- Two types of DMA :
  - Synchronous DMA
    - The DMA operation is caused by software.
    - For example, sound card driver may trigger DMA operation to play music.
  - Asynchronous DMA
    - The DMA operation is caused by devices (hardware).
    - For example, network card use DMA operation to load data into memory and interrupt CPU for further manipulation.

PCI & PCI Express

- What is PCI ?
  - PCI (Peripheral Component Interconnect) is a computer bus for attaching hardware devices.
  - Typical PCI cards used include :
    - Network cards, sound cards, modems
    - Extra ports such as USB or serial, TV tuner cards and disk controllers.
- What is PCI Express ?
  - PCIe is a computer expansion card standard designed to replace the older PCI, PCI-X, and AGP standards.
  - Its topology is based on point-to-point serial links, rather than a shared parallel bus architecture.

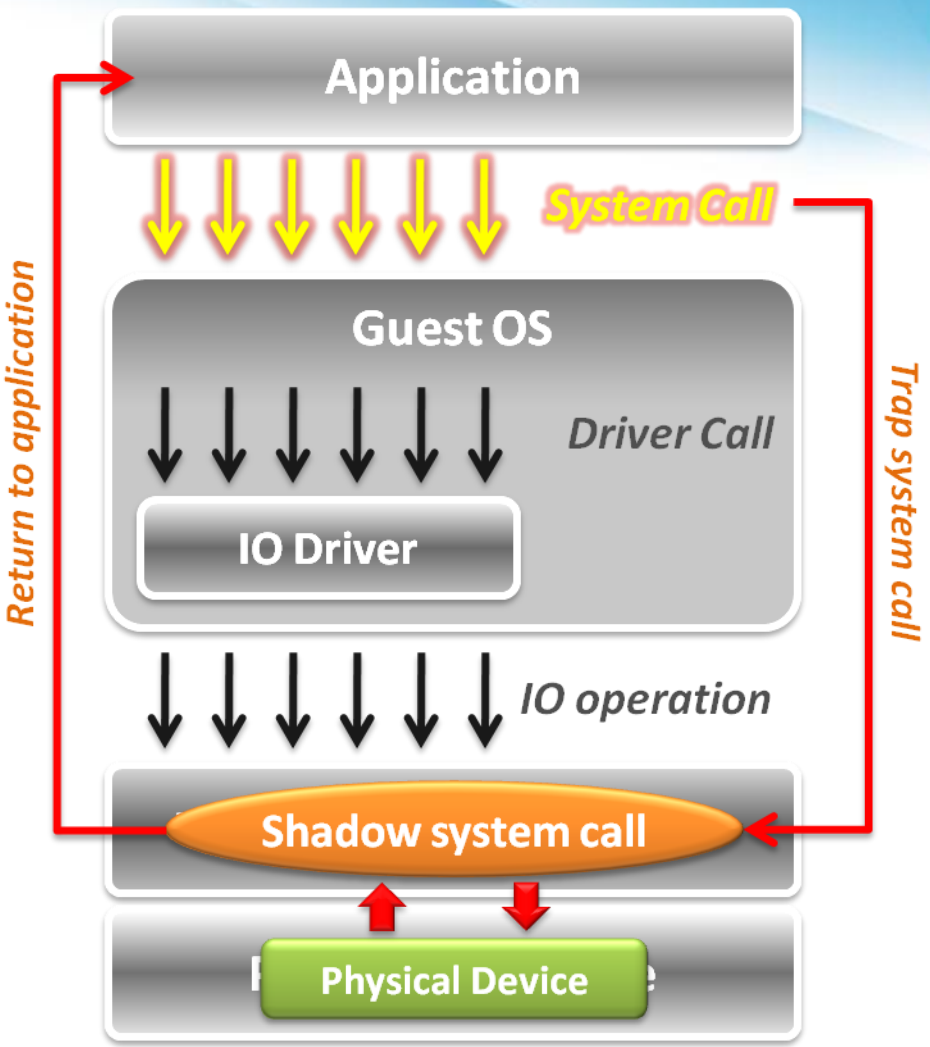
IO Virtualization

- Implementation Layers :
  - System call
    - The interface between applications and guest OS.
  - Driver call
    - The interface between guest OS and IO device drivers.
  - IO operation
    - The interface between IO device driver of guest OS and virtualized hardware (in VMM).



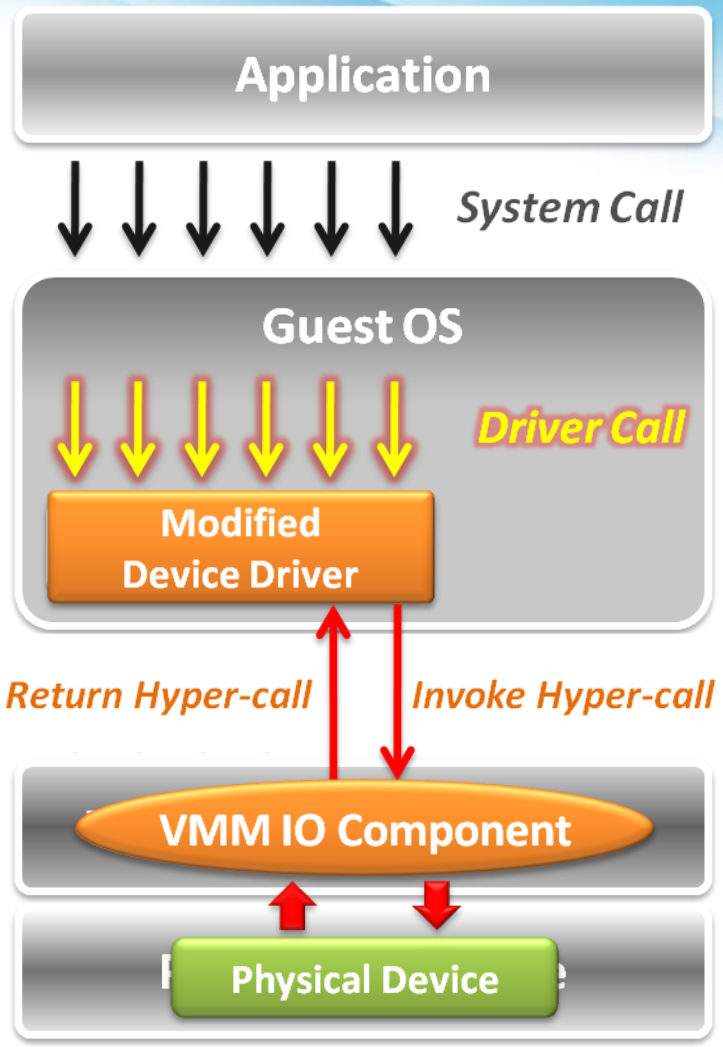
IO Virtualization

- In **system call** level :
  - When application invoke a system call, system will trap to VMM first.
  - VMM intercepts system calls, and maintains shadowed IO system call routines to simulate functionalities.
  - After simulation, VMM directly return to application in gust OS.



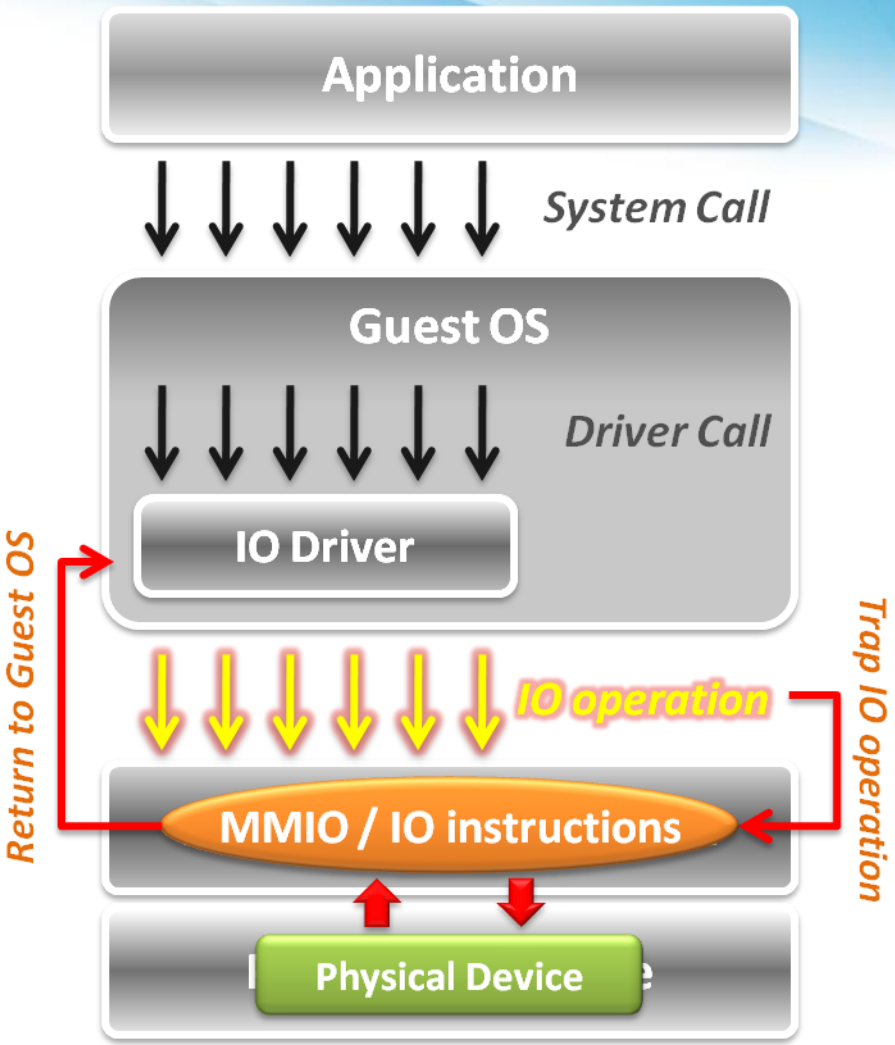
IO Virtualization

- In **device driver call** level :
  - Utilize para-virtualization technique, which means the IO device driver in guest OS should be modified.
  - The IO operation is invoked by means of hyper-call between the modified device driver and VMM IO component.



IO Virtualization

- In **IO operation** level, two approaches :
  - Memory mapped IO
    - Loads/stores to specific region of real memory are interpreted as command to devices.
    - The memory mapped IO region is protected.
  - Port mapped IO
    - Special input/output instructions with special addresses.
    - The IO instructions are privileged .
- Due to the privileged nature, these IO operations will be trapped to the VMM.



Overview

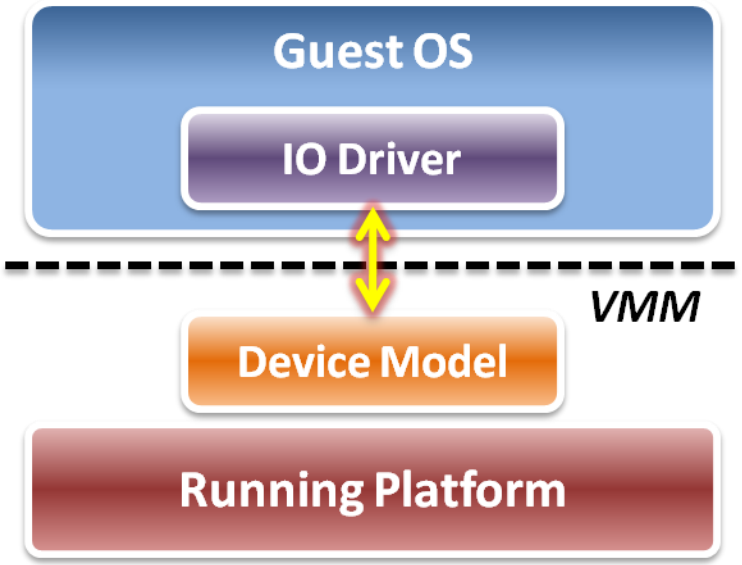
Device Model

Hardware Assistance

IO VIRTUALIZATION

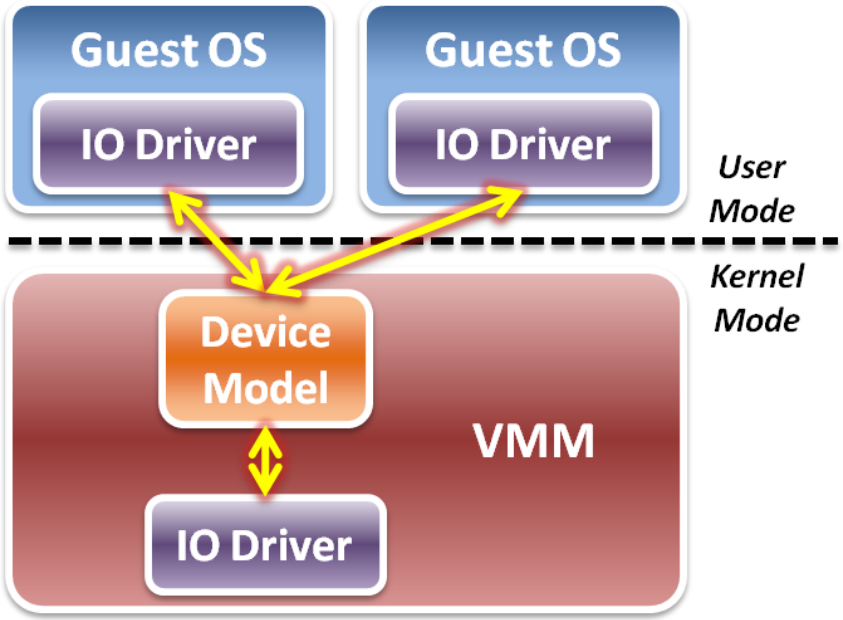
Device Model

- We focus on IO operation level implementation.
  - This is an approach of full virtualization.
- Logic relation between guest OS and VMM :
  - VMM intercepts IO operations from guest OS.
  - Pass these operations to device model on a running platform.
  - Device model need to emulate the IO operation interfaces.
    - Port mapped IO
    - Memory mapped IO
    - DMA
    - … etc.

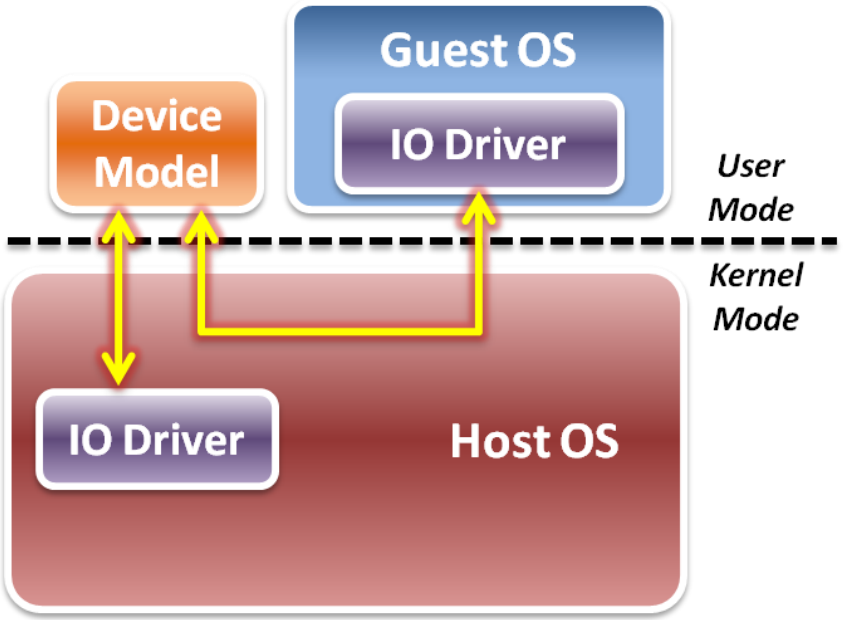


Device Model

- Two different implementations of device model ：
  1. Device model is implemented as a part of VMM.
  2. Device model is running in user space as a stand alone service.



Type 1 Virtualization



Type 2 Virtualization

Device Model

- IO virtualization paradigm
  - Initialization – device discovery
    - VMM will make guest OS discover the virtualized IO devices.
    - Then guest OS will load the corresponding device driver.
  - Operation – access interception
    - When guest OS applies IO operations, VMM will intercept those accesses.
    - After virtual device operations, VMM returns the control to guest OS.
  - Virtualization – device virtualization
    - Device model should emulate the real electronic logic to satisfy all device interface definition and its effects.
    - VMM may share physical devices to all virtual machines.

Device Discovery

- Virtualize physical bus devices
  - Non-enumerable physical device
    - These devices have their own hard-coded numbers.
    - VMM should setup some status information on the virtual device ports.
    - For example, PS/2 keyboard and mouse.
  - Enumerable physical device
    - These devices defined a complete device discover method.
    - VMM have to emulate not only the device itself, but the bus behavior.
    - For example, PCI or PCI express devices.
- Virtualize non-exist devices
  - VMM must define and emulate all functions of these devices
    - VMM may define them as either non-enumerable or enumerable devices.
    - Guest OS needs to load some new drivers of these virtual devices.

Access Interception

- After virtual device discovered by guest OS, VMM has to intercept and control all the IO operations from guest OS.
- Port mapped IO operation
  - Direct device assignment
    - VMM should turn ***ON*** the physical IO bitmap.
    - All the IO instructions (**`IN`**/**`OUT`**) from guest OS will be directly performed onto hardware without VMM intervention.
  - Indirect device assignment
    - VMM should turn ***OFF*** the physical IO bitmap.
    - All the IO instructions from guest OS will be intercepted by VMM and forward to physical hardware.

Access Interception

- Memory mapped IO operation
  - Direct device assignment
    - VMM should use the shadow page table to map IO device addressing space of guest OS to the space of host.
    - Then all the IO operations from guest OS will not be intercepted.
  - Indirect device assignment
    - VMM should make the all entries of the IO device addressing space in the shadow page table to be invalid.
    - When guest OS access those addressing space, it will introduce the page fault which trap CPU to VMM for device emulation.
- DMA mechanism
  - Address remapping
    - Because the device driver in the guest OS have nothing to know with the host physical address, VMM need to automatic remap the DMA target when intercepting guest OS.

Device Virtualization

- IO device types :
  - Dedicated device
    - Ex : displayer, mouse, keyboard …etc.
  - Partitioned device
    - Ex : disk, tape …etc
  - Shared device
    - Ex : network card, graphic card …etc.
  - Nonexistent physical device
    - Ex : virtual device …etc.

Device Virtualization

- Dedicated device
  - Do not necessarily have to be virtualized.
  - In theory, requests of such device could bypass the VMM.
  - However, they are handled by the VMM first since OS is running in user mode.
- Partitioned device
  - Be partitioned into several smaller virtual devices as dedicated to VM.
  - VMM translates address spaces to those of the physical devices.

Device Virtualization

- Shared device
  - Should be shared among VMs.
  - Each VM has its own virtual device state.
  - VMM translates requests from a VM to physical device .
- Nonexistent physical device
  - Virtual device “attached” to a VM for which there is no corresponding physical device.
  - VMM intercepts requests from a VM, buffers it and interrupts other VMs.

Performance Issues

- When considering performance, two major problems :
  - How to make guest OS directly access IO addresses ?
    - Other than software approaches discussed above, we can make use of the hardware assistance (Intel EPT technique in memory virtualization) to map IO addresses from host to guest directly without software overhead.
  - How to make DMA directly access memory space in guest OS ?
    - For the synchronous DMA operation, guest OS will be able to assign the correct host physical memory address by EPT technique.
    - For the asynchronous DMA operation, hardware must access memory from host OS which will introduce the VMM intervention.

Overview

Device Model

Hardware Assistance

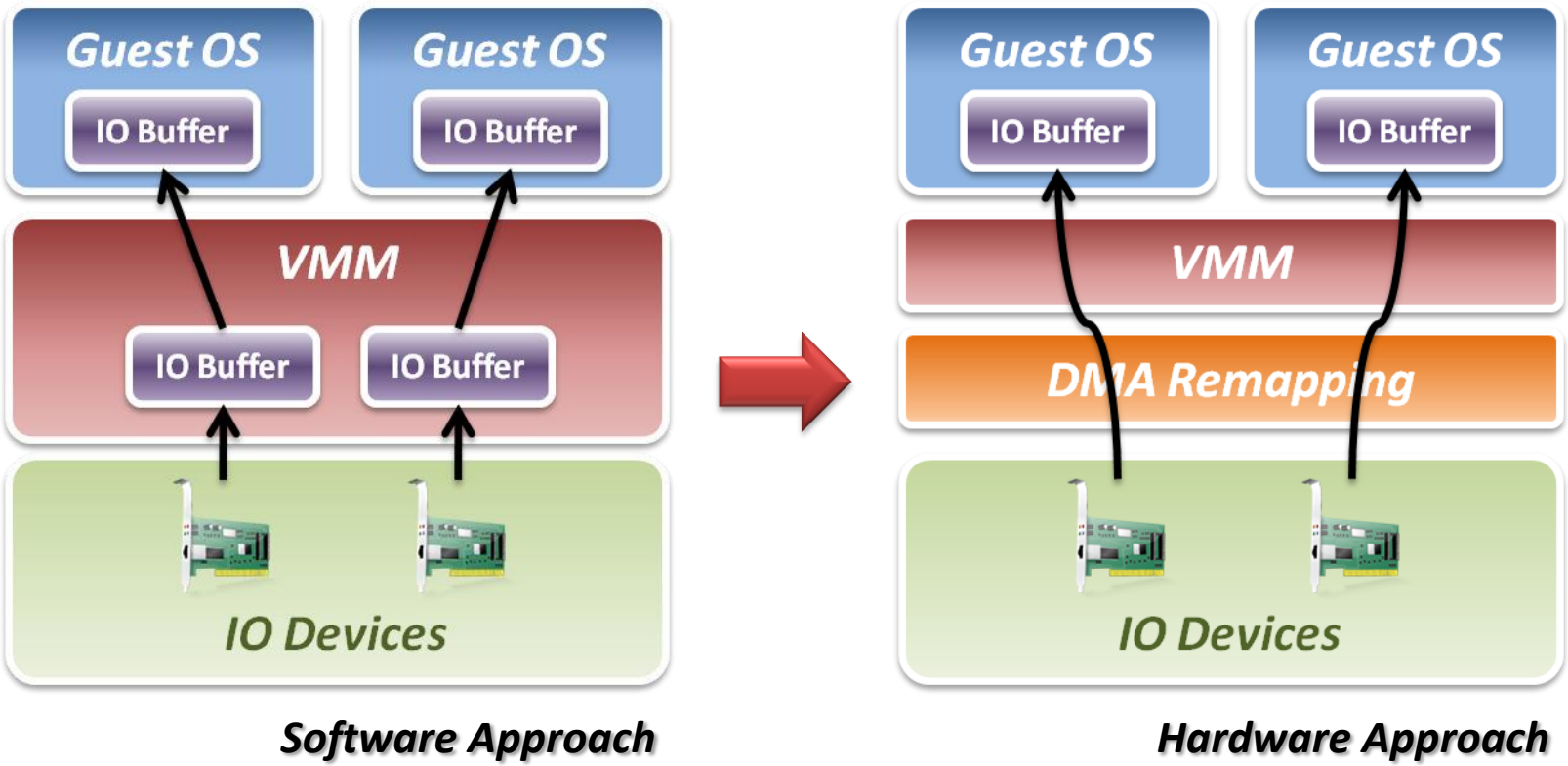
IO VIRTUALIZATION

Hardware Solution

- Difficulty :
  - Software cannot make data access directly from devices.
- Tow hardware solutions :
  - Implement DMA remapping in hardware
    - Remap DMA operations automatically by hardware.
    - For example, ***Intel VT-d*** .
  - Specify IO virtualization standards of PCI Express devices
    - Implement virtualizable device with PCI Express interface.
    - For example, ***SR-IOV*** or ***MR-IOV***.

Intel VT-d

- Add DMA remapping hardware component.



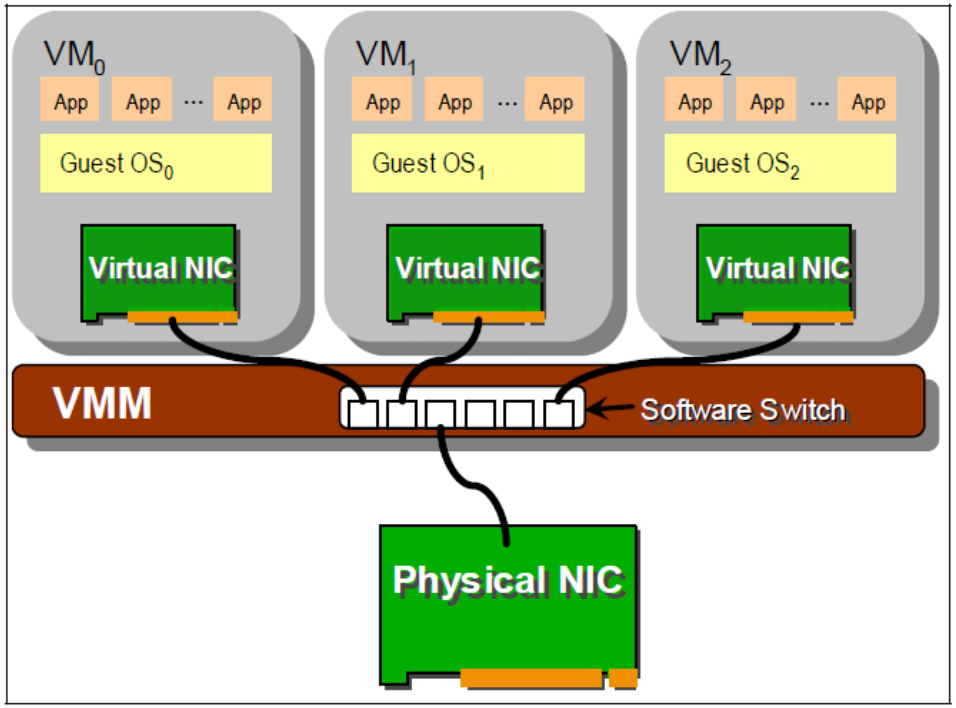
Software Approach

Hardware Approach

IO Virtualization Brief Review

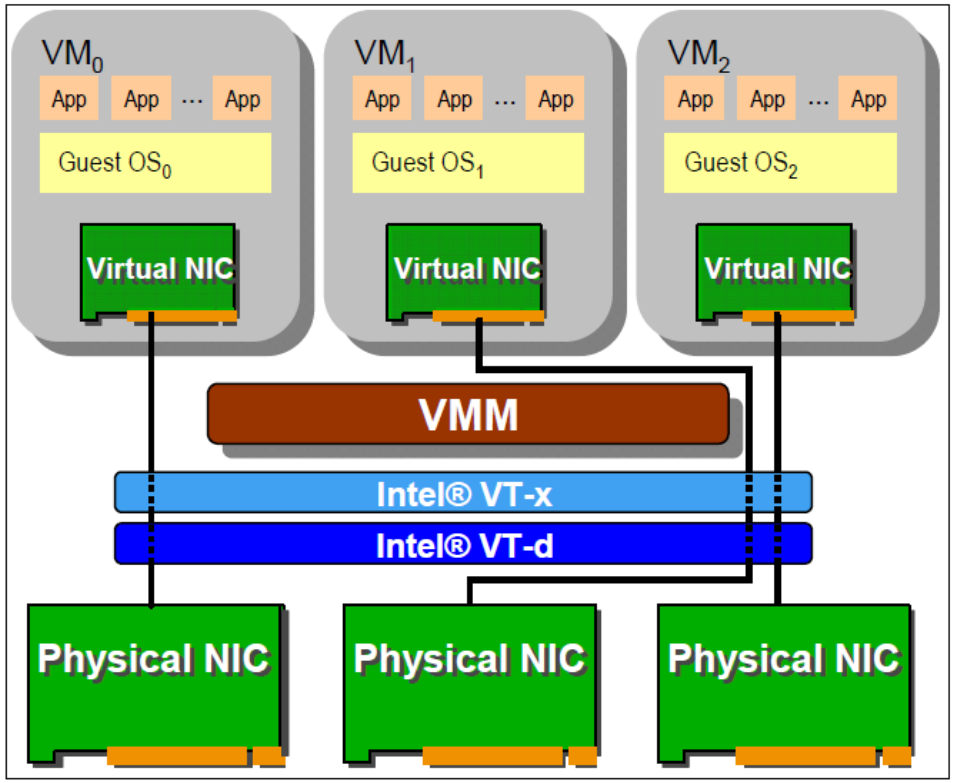
- Let's give a brief IO virtualization review.

- Software based sharing
  - Implement virtualization by VMM software stack.
  - Advantage
    - Full virtualization without special hardware support.
  - Disadvantage
    - Significant CPU overhead may be required by the VMM.



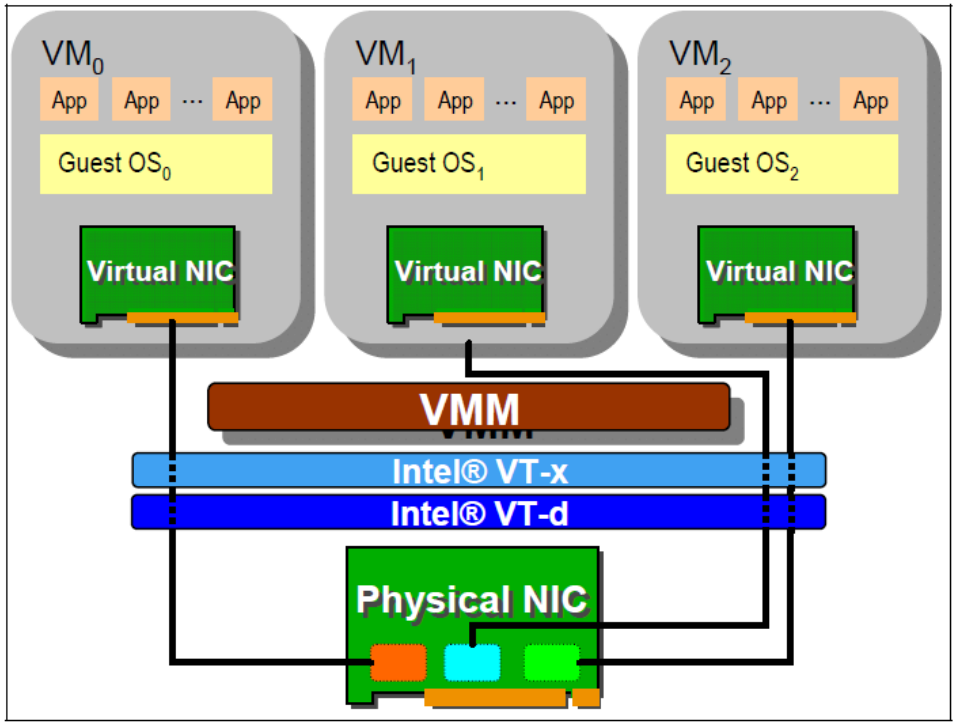
IO Virtualization Brief Review

- Hardware direct assignment
  - Implement virtualization with Intel VT-x and VT-d supports.
  - Advantage
    - Data access bypass VMM.
    - Improve IO performance.
  - Disadvantage
    - Dedicate physical device assignment limit the system scalability.



IO Virtualization Brief Review

- New industrial standard
  - Instead of implementing virtualization in CPU or memory only, industry com up with new IO virtualization standard in PCI Express devices.
  - Advantages
    - Fully collaboration with physical hardware devices.
    - Improve system scalability.
    - Improve system agility.
  - Disadvantages
    - IO devices must implement with new specification.



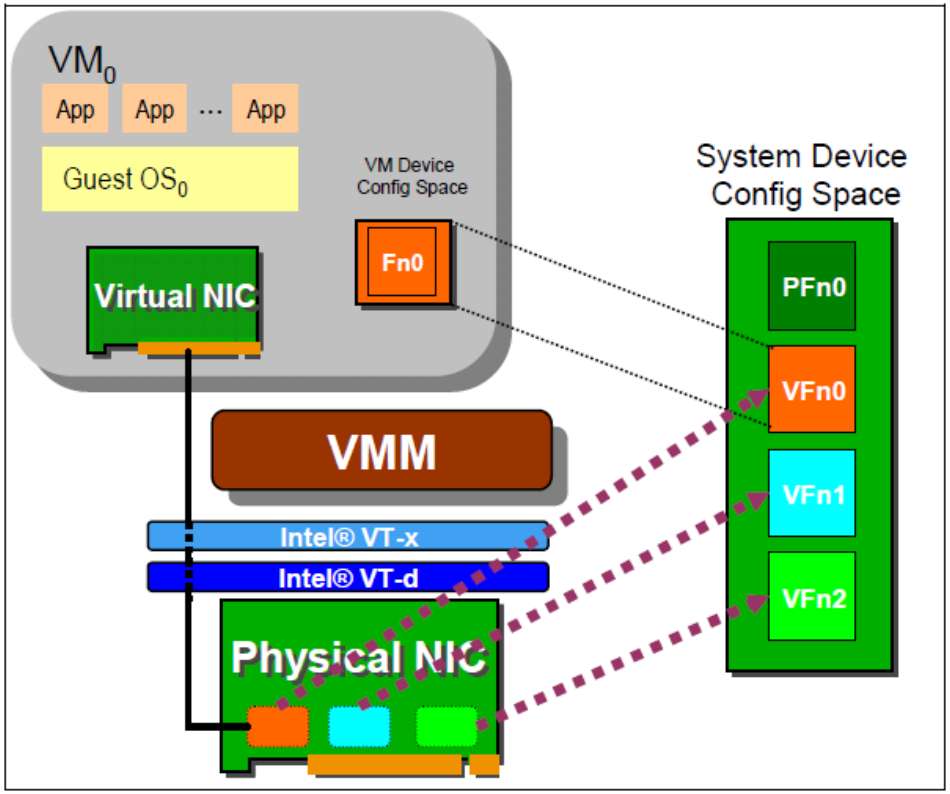
Single Root – IO Virtualization

- What is SR-IOV ?
  - The PCI-SIG Single Root I/O Virtualization and Sharing (SR-IOV) specification defines a standardized mechanism to create natively shared devices.

- Basic components :
  - Physical Functions (PFs):
    - These are full PCIe functions that include the SR-IOV Extended Capability.
    - The capability is used to configure and manage the SR-IOV functionality.
  - Virtual Functions (VFs):
    - These are “lightweight” PCIe functions that contain the resources necessary for data movement but have a carefully minimized set of configuration resources.

Single Root – IO Virtualization

- SR-IOV works with VMM :
  - VMM
    - An SR-IOV-capable device can be configured to appear in the PCI configuration space as multiple functions.
  - VM
    - The VMM assigns one or more VFs to a VM by mapping the actual configuration space the VFs to the configuration space presented to the virtual machine by the VMM.



IO Virtualization Summary

- IO subsystem architecture
  - Port Mapped IO vs. Memory Mapped IO
  - Direct Memory Access (DMA)
  - PCI / PCI Express
- IO virtualization
  - Three implementation layers
  - IO virtualization paradigm with device model
- Hardware assistance
  - DMA remapping hardware
  - Single Root – IO Virtualization specification