Rack Disaggregation Using PCIe Networking

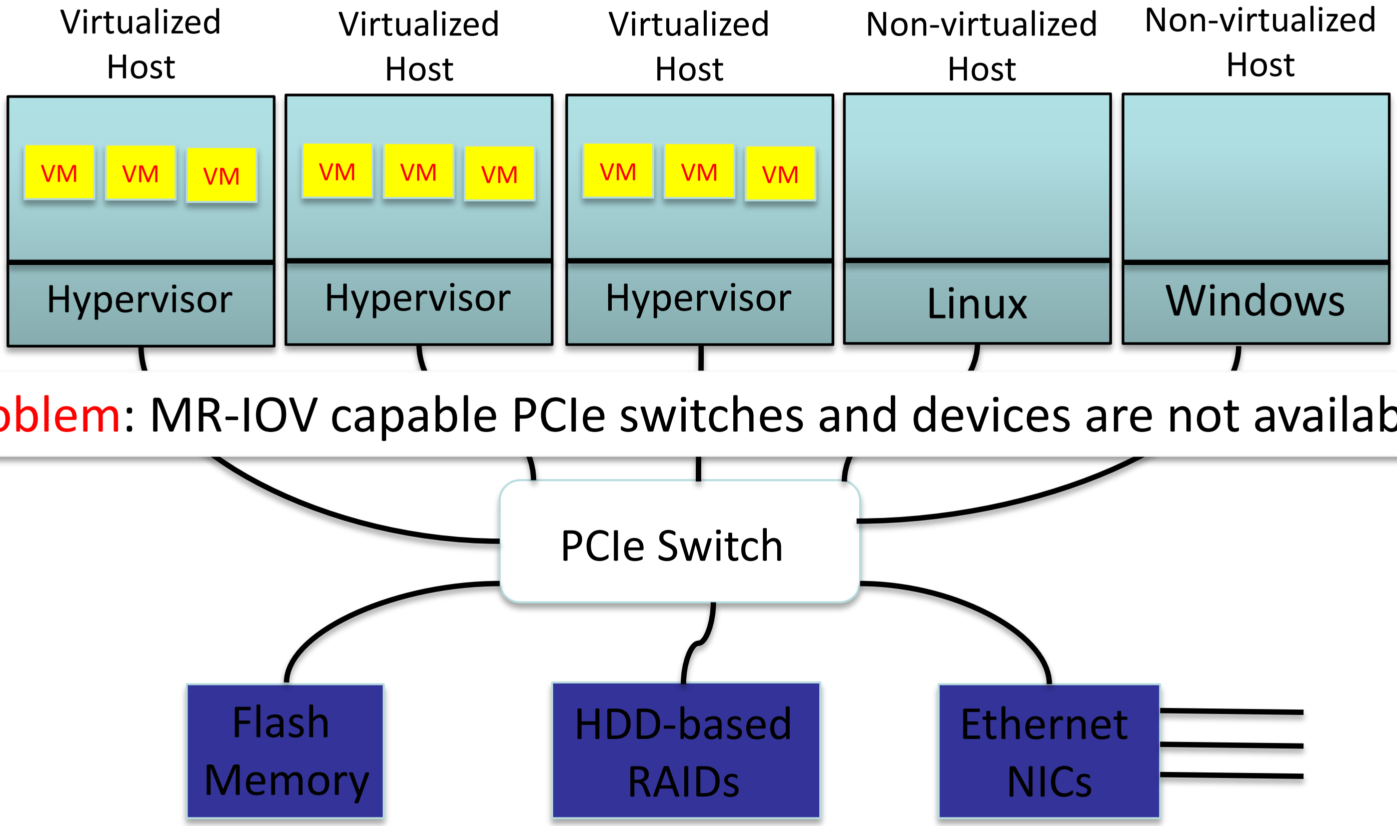
Cloud Computing Research Center for Mobile Applications (CCMA)
Industrial Technology Research Institute
雲端運算行動應用研究中心

Rack Disaggregation

- Rack as the basic building block for cloud-scale data centers
- Rack disaggregation: pooling of HW resources for global allocation and independent upgrade cycle for each resource type
- CPU/memory/NICs/Disks embedded in individual hosts
  - Disk pooling in a rack
  - NIC/Disk pooling in a rack
  - Memory/NIC/Disk pooling in a rack
- Enabling technology: High-speed rack-area networking that allows direct memory access
  - PCI Express is a promising candidate (Gen3 x 8 = 64Gbps)
  - New top-of-rack (TOR) switch consists of PCIe ports and Ethernet ports

I/O Device Disaggregation

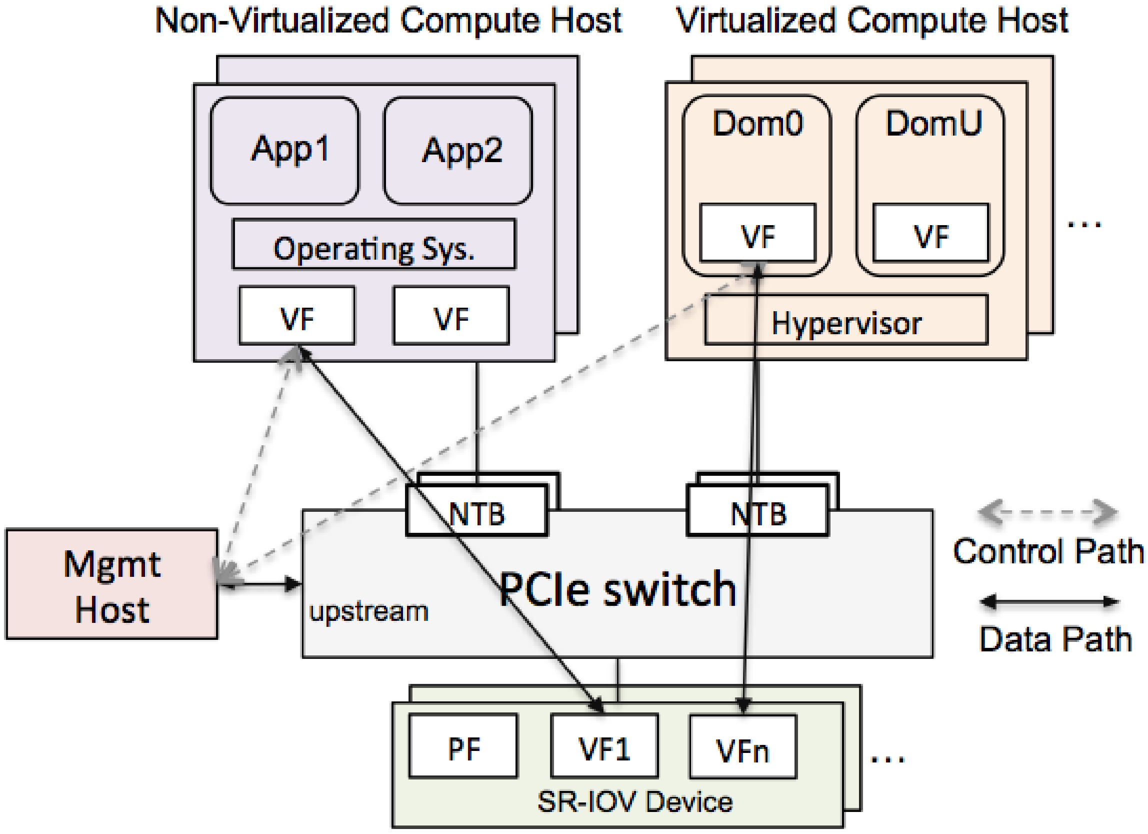
- Reduce cost: One I/O device per rack rather than one per host
- Maximize Utilization: Statistical multiplexing benefit
- Power efficient: intra-rack networking and device count
- Reliability: Pool of devices available for backup

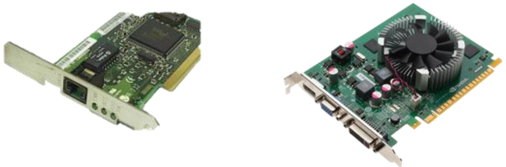


Problem: MR-IOV capable PCIe switches and devices are not available

ITRI Software-based MR-SRIOV

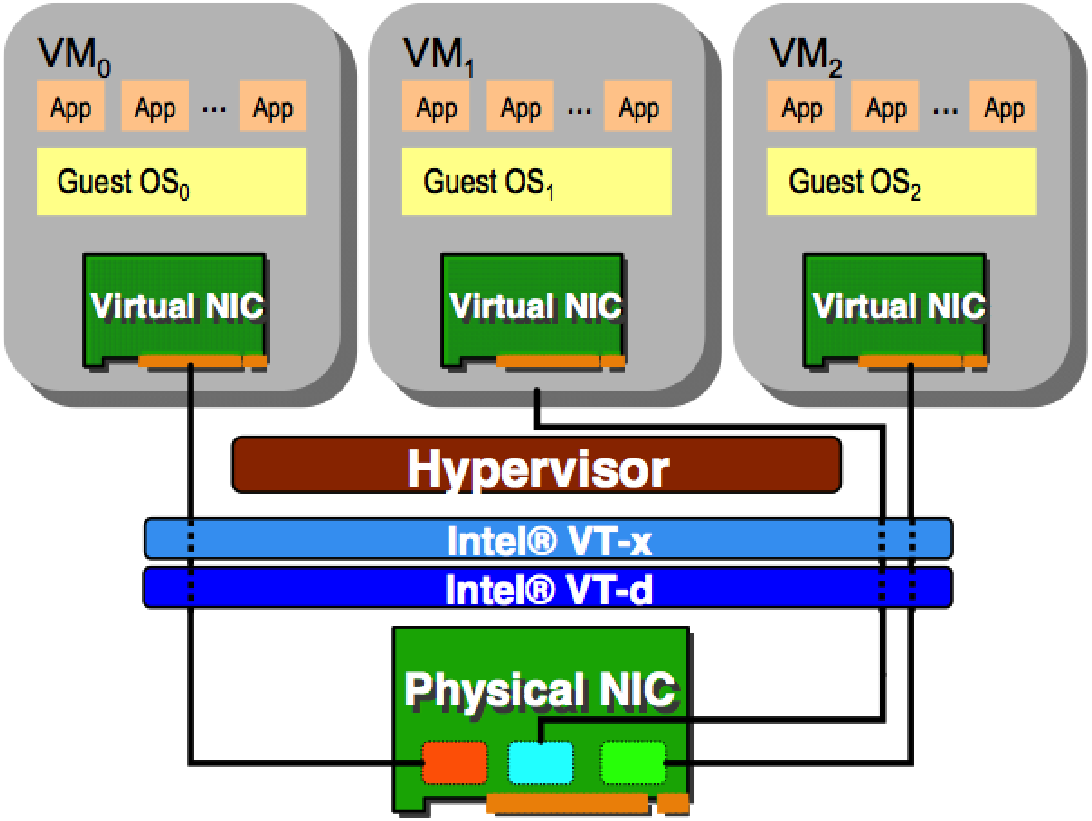
- Standard PCIe switches and device:
  - No MR-IOV capability
  - Native driver
  - Zero data copying
- Native Performance:
  - Control path goes to a central authority
  - Direct data path to and from devices
- Secure Sharing:
  - Access control using existing technologies





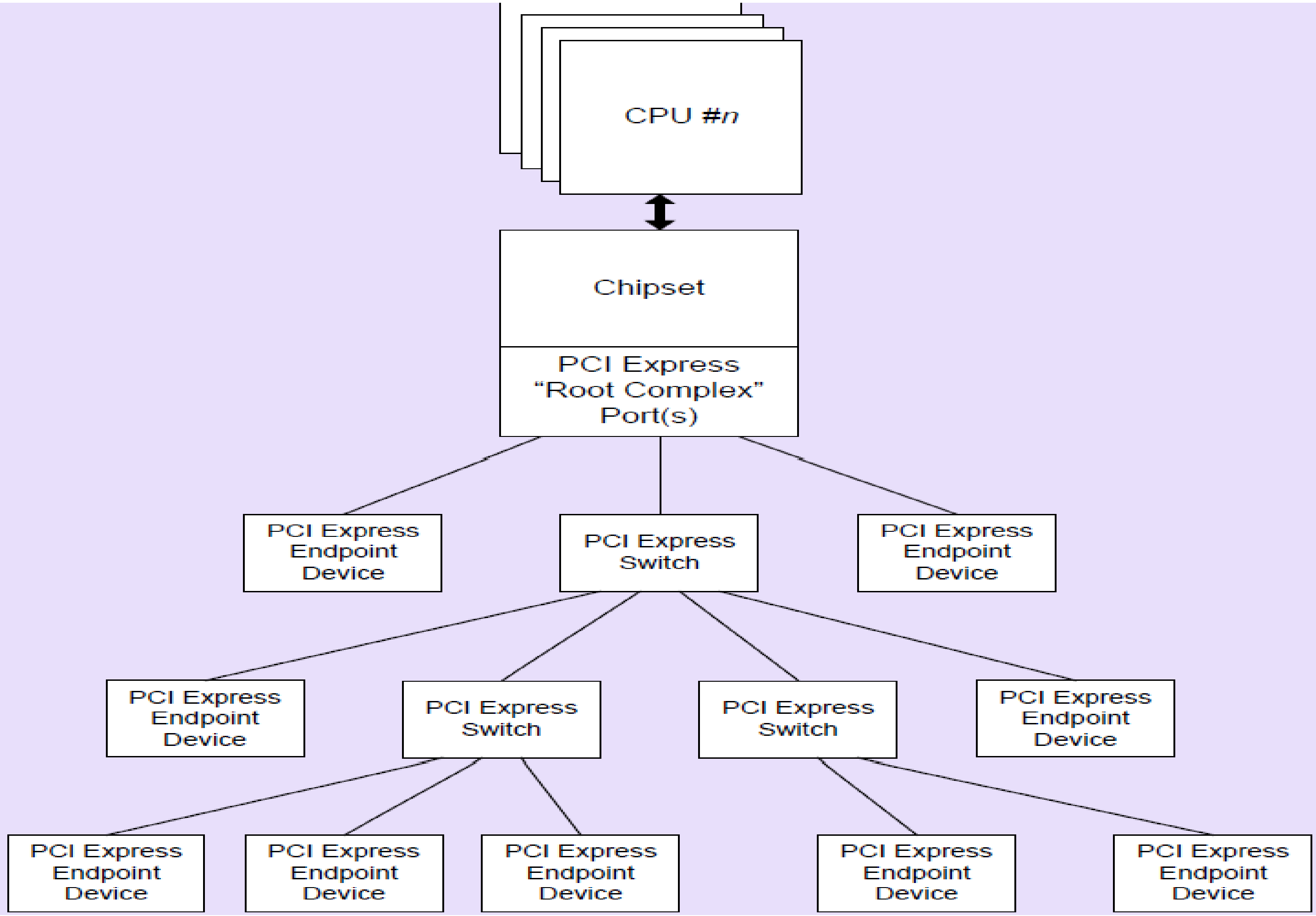
I/O Virtualization

- Direct communication:
  - Device directly assigned to VMs
  - Bypass the hypervisor
- Physical Function:
  - configure and manage the SR-IOV functionality
- Virtual Function:
  - lightweight PCIe function with resources necessary for data movement
- VT-x and VT-d
  - CPU/Chipset support for VMs and devices

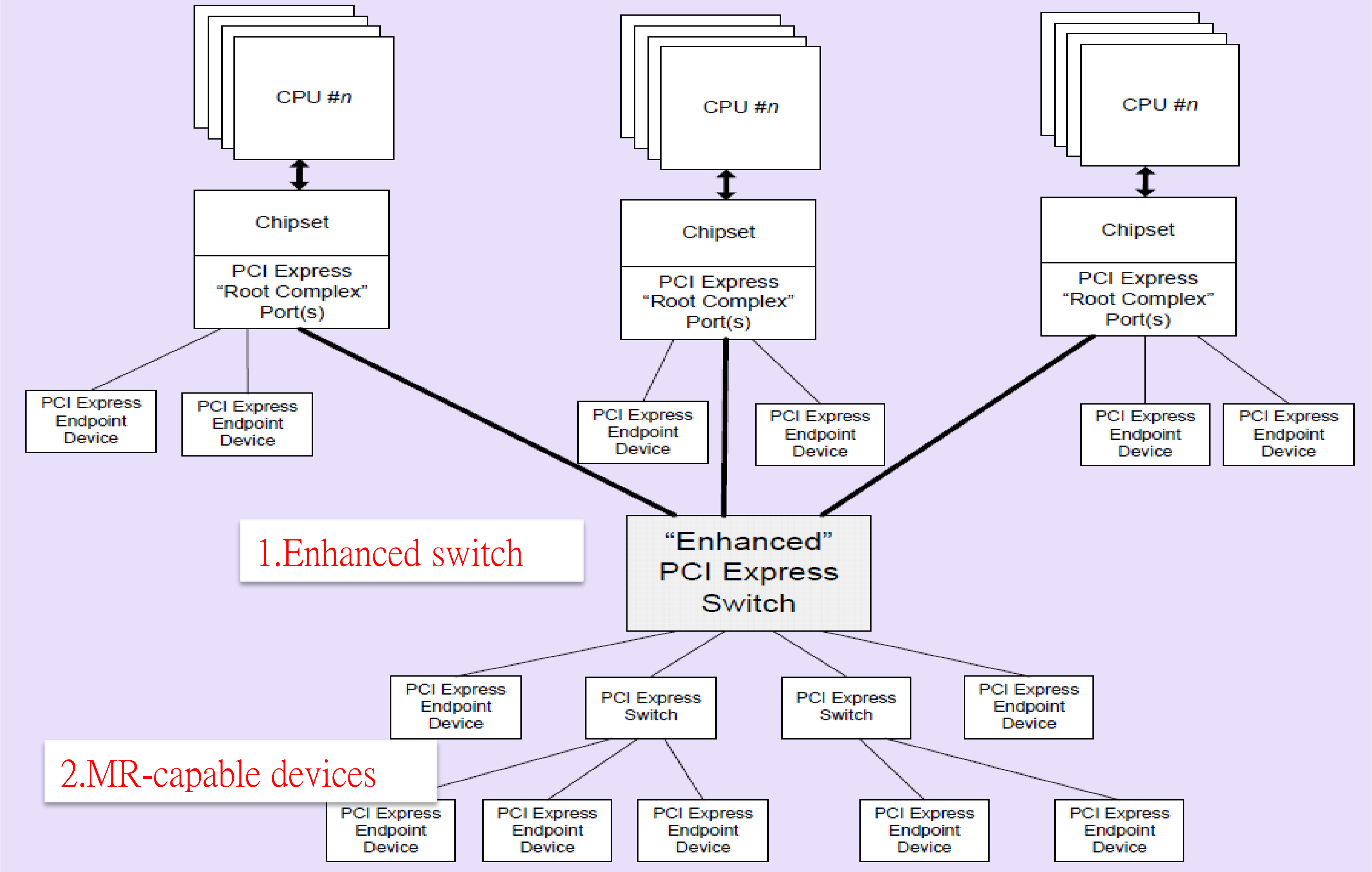


However, works only in a single host (hence Single Root)

Single Root (SR) Architecture


CPU #*n*
Chipset
PCI Express
"Root Complex"
Port(s)
PCI Express Endpoint Device
PCI Express Switch
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Switch
PCI Express Switch
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Endpoint Device

Multi Root (MR) Architecture

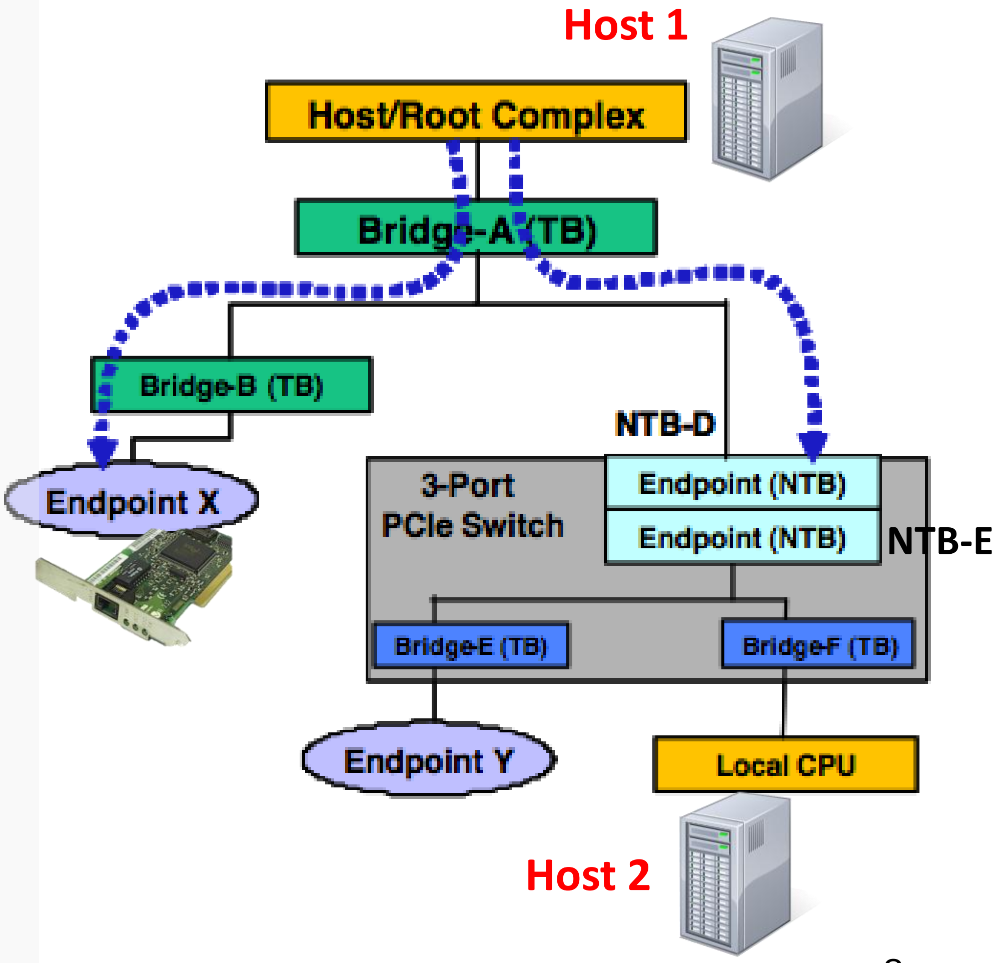

CPU #n
CPU #n
CPU #n
Chipset
PCI Express
"Root Complex"
Port(s)
Chipset
PCI Express
"Root Complex"
Port(s)
Chipset
PCI Express
"Root Complex"
Port(s)
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Endpoint Device
"Enhanced"
PCI Express
Switch
PCI Express Endpoint Device
PCI Express Switch
PCI Express Switch
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Endpoint Device
PCI Express Endpoint Device

1.Enhanced switch

2.MR-capable devices

Non-Transparent Bridge (NTB)

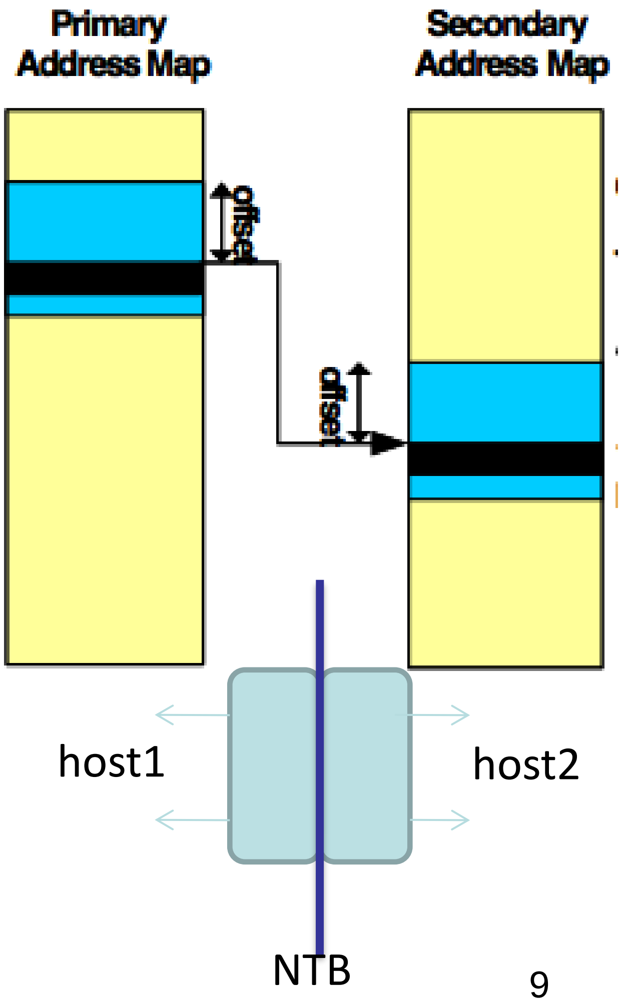
- PCI Enumeration
  - Host1’s enumeration finds X and NTB-D
  - Host 2’s enumeration finds Y and NTB-E
- Translations between domains
  - NTB-D exposes a portion of physical memory address range of Host2’s domain
  - Memory address:
    - From primary side and secondary side
  - PCI device ID:
    - Querying the ID lookup table (LUT)



NTB Address Mapping

- NTB maps from:
  - <the primary side to the secondary side>
- Mapping from addrA at primary side to addrB at the secondary side

- Example:
  - AddrA = 0x8000 at BAR4 from Host1
  - AddrB = 0x10000 at Host2's DRAM
- One-way Translation:
  - Read/write at addrA in Host 1== read/write addrB in Host 2
  - Read/write at addrB in Host 2 has nothing to do with addrA in Host 1



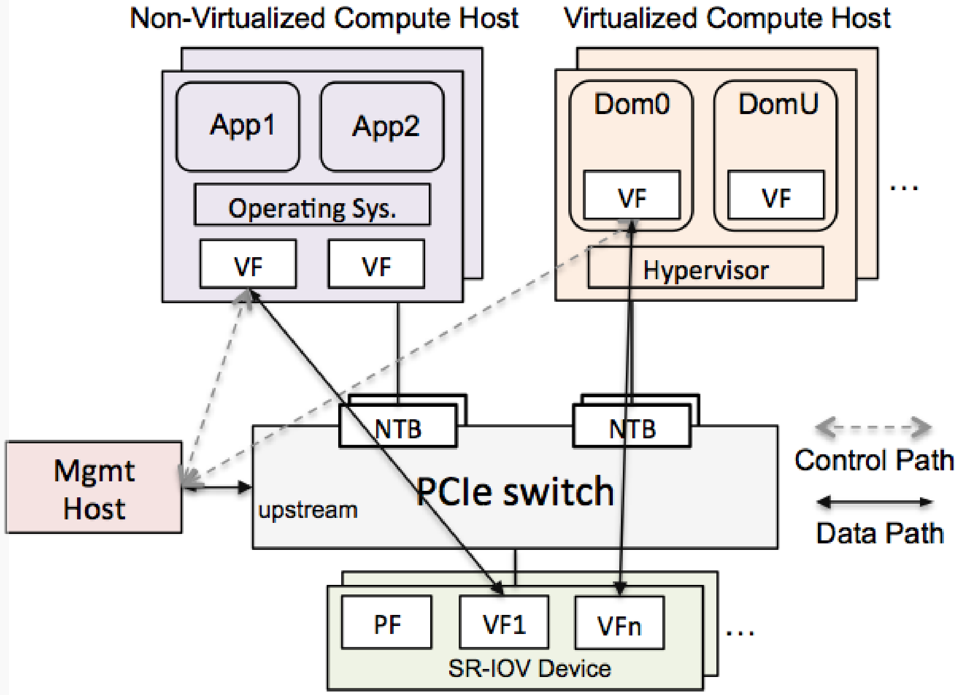
Ladon: a hundred-heads dragon
who guards the Garden of Hesperides

Ladon: A software-defined PCIe network architecture that supports secure MR-IOV using SR-IOV devices

工業技術研究院
Industrial Technology
Research Institute

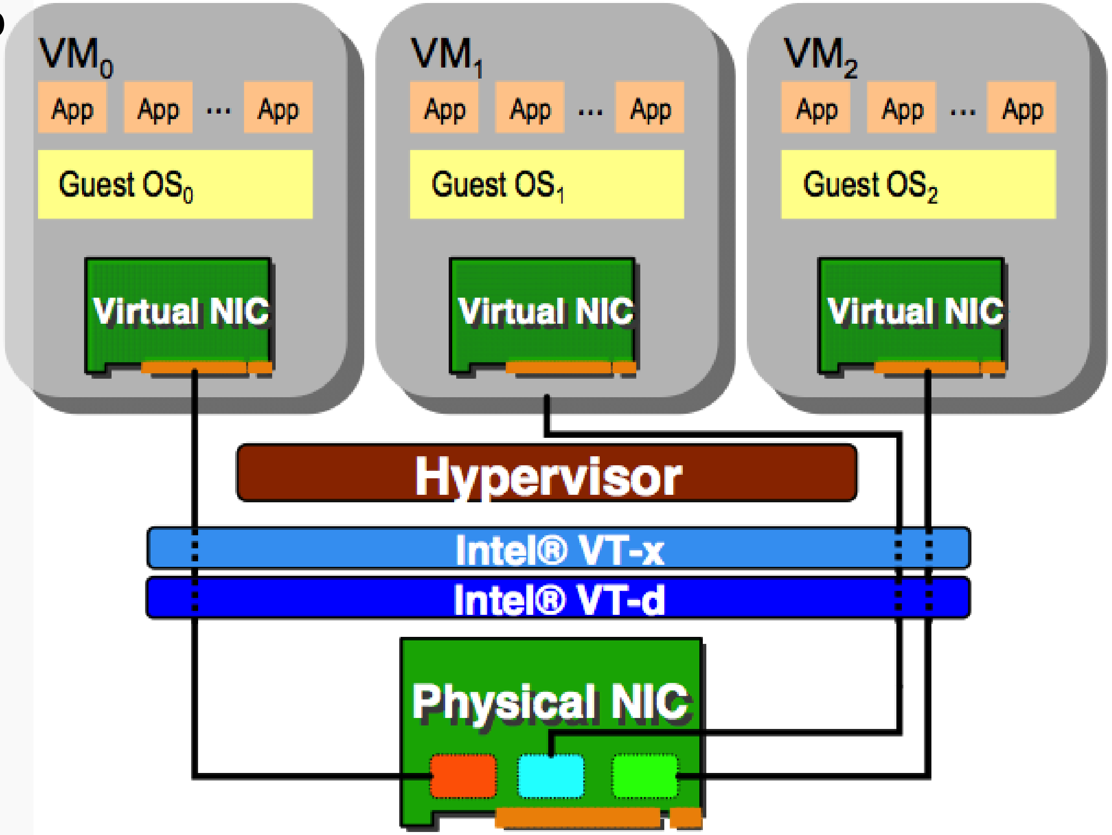
System Components

- Management Host (MH)
  - Manage the shared I/O devices
- Compute Host (CH)
  - Non-virtualized host OS can directly access VF
  - Virtualized host with VMs can directly access VF
- Non-Transparent Bridge (NTB)
  - Each CH connects to the fabric via an NTB
- PCIe Switch & SR-IOV device
  - PF and multiple VFs



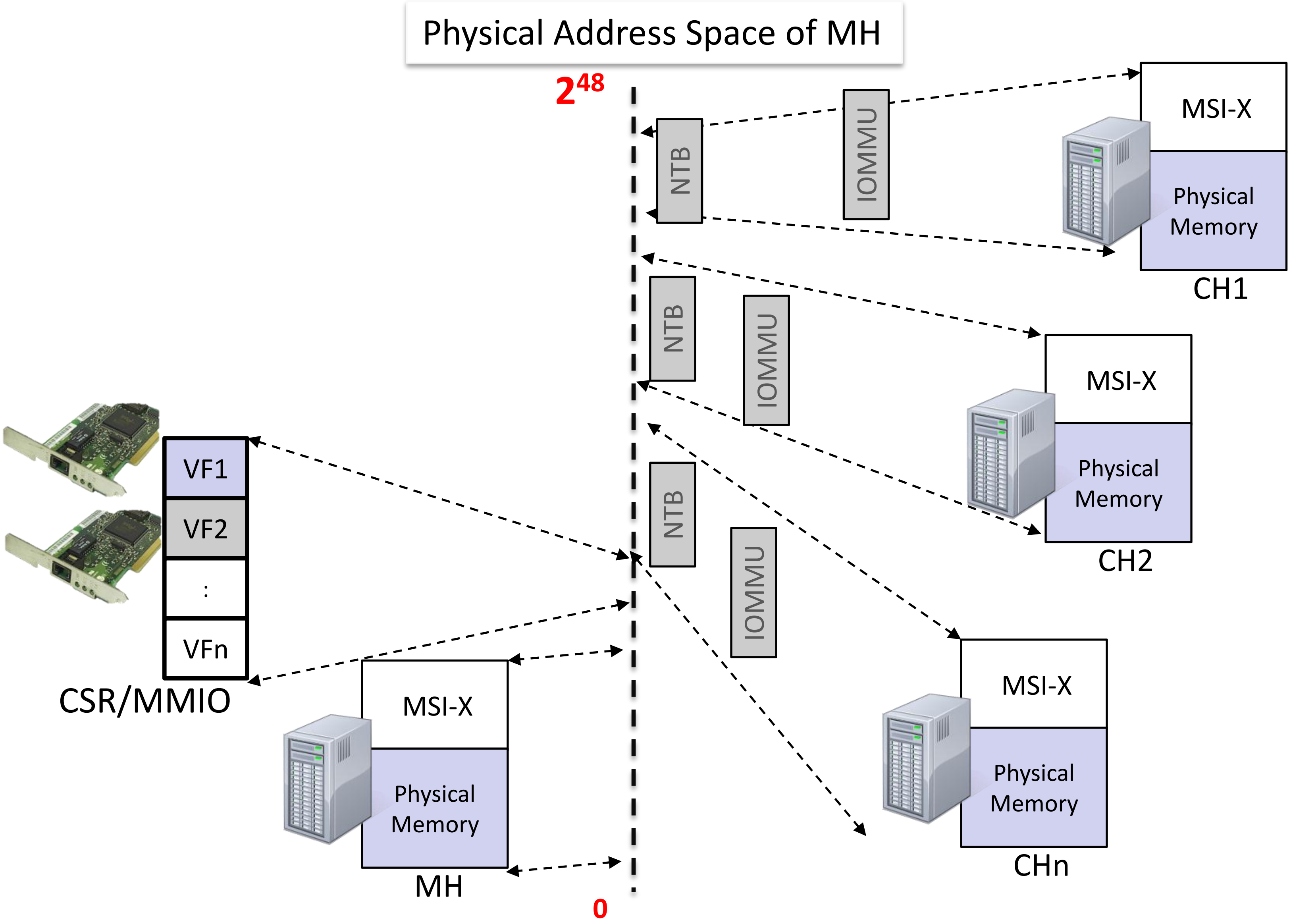
I/O Virtualization

- Direct communication:
  - Device directly assigned to VMs
  - Bypass the hypervisor overhead
- Physical Function:
  - configure and manage the SR-IOV functionality
- Virtual Function:
  - lightweight PCIe function with resources necessary for data movement
- VT-x and VT-d
  - CPU/Chipset support for VMs and devices



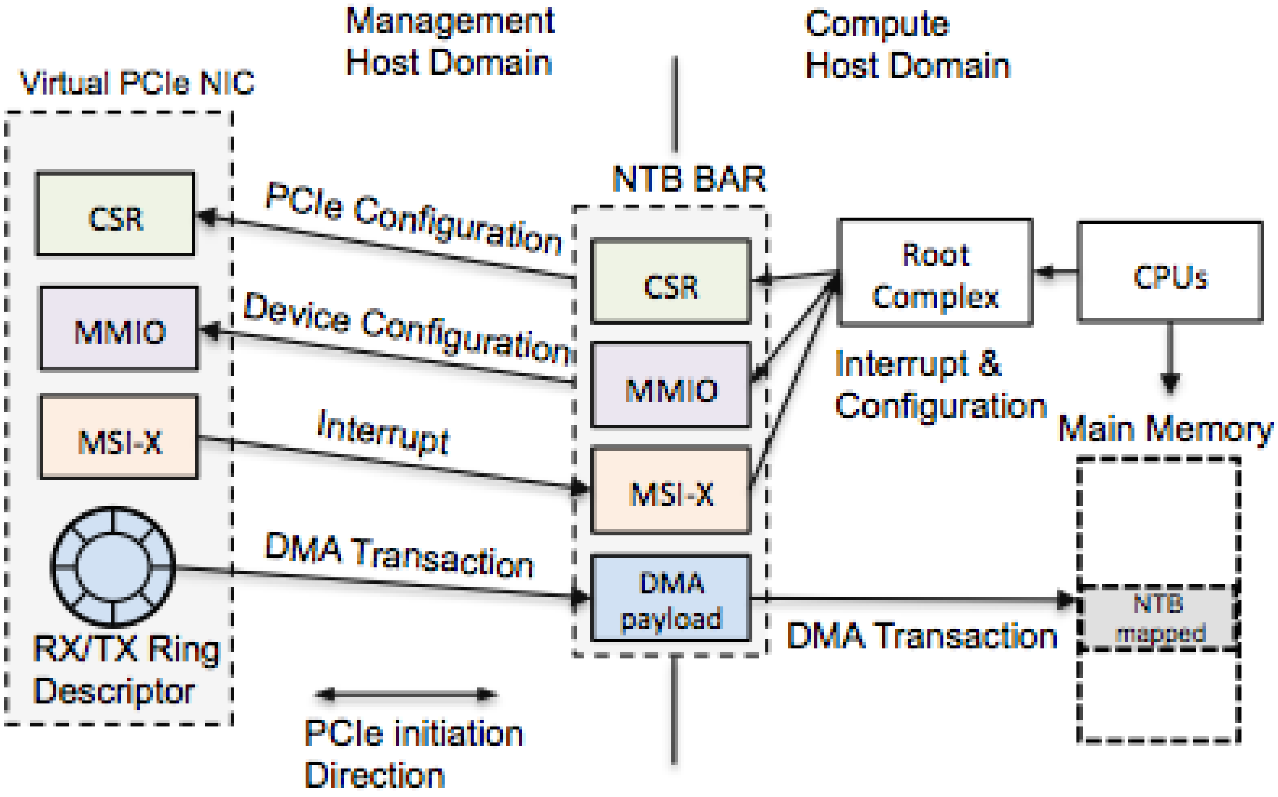
However, applicable in a single host (SR -> Single Root)

Global Physical Address Space



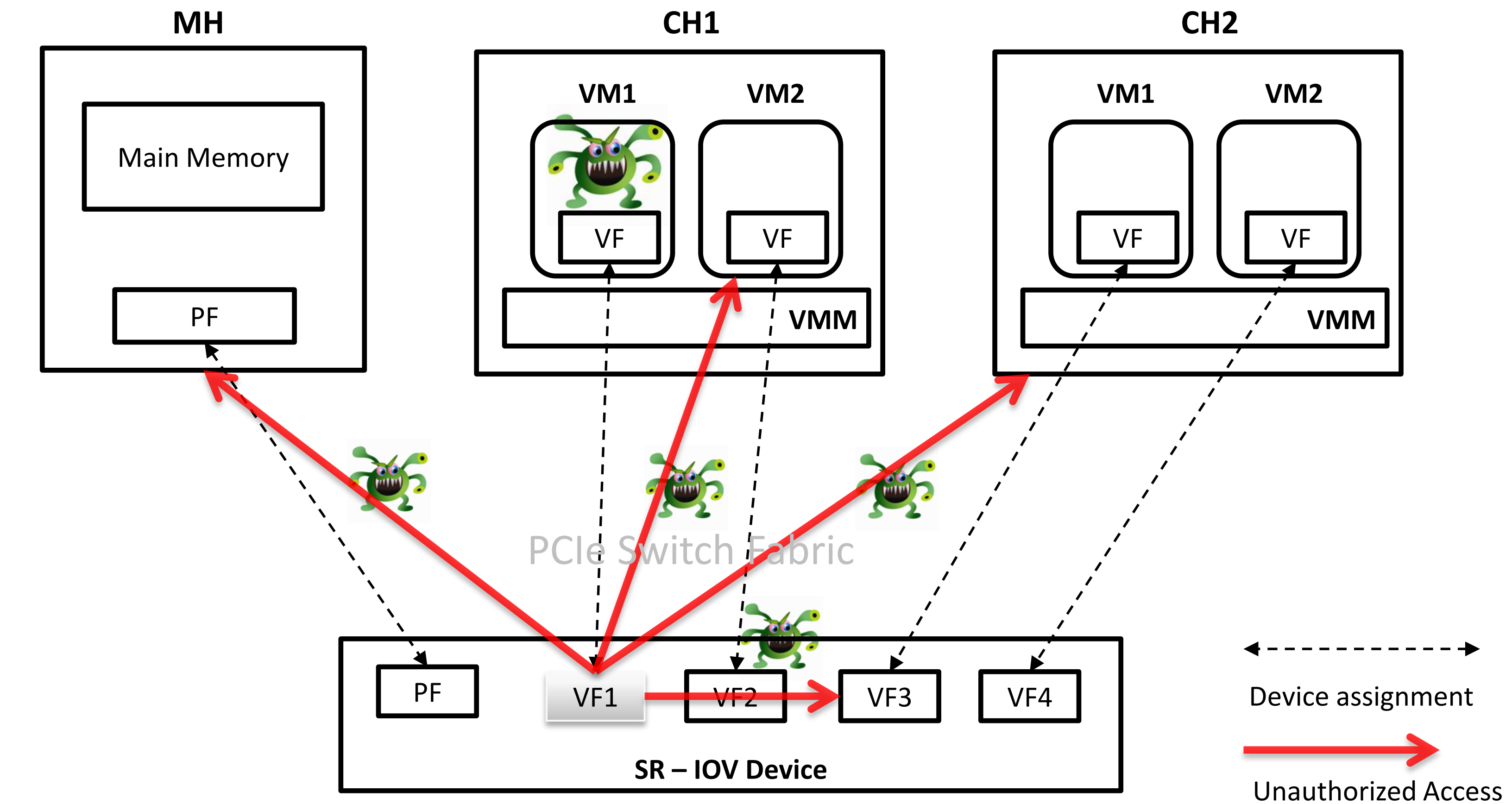
Virtual NIC Configuration

- Each virtual NIC in a CH is backed by a VF of an SRIOV NIC
- Identify the virtual NIC’s CSR, MMIO, MSI-X and DMA payload area
- Install mappings in the BARs of the asscociated NTB port



Secure Threats: 4 Cases

Malicious VM, CH and MH



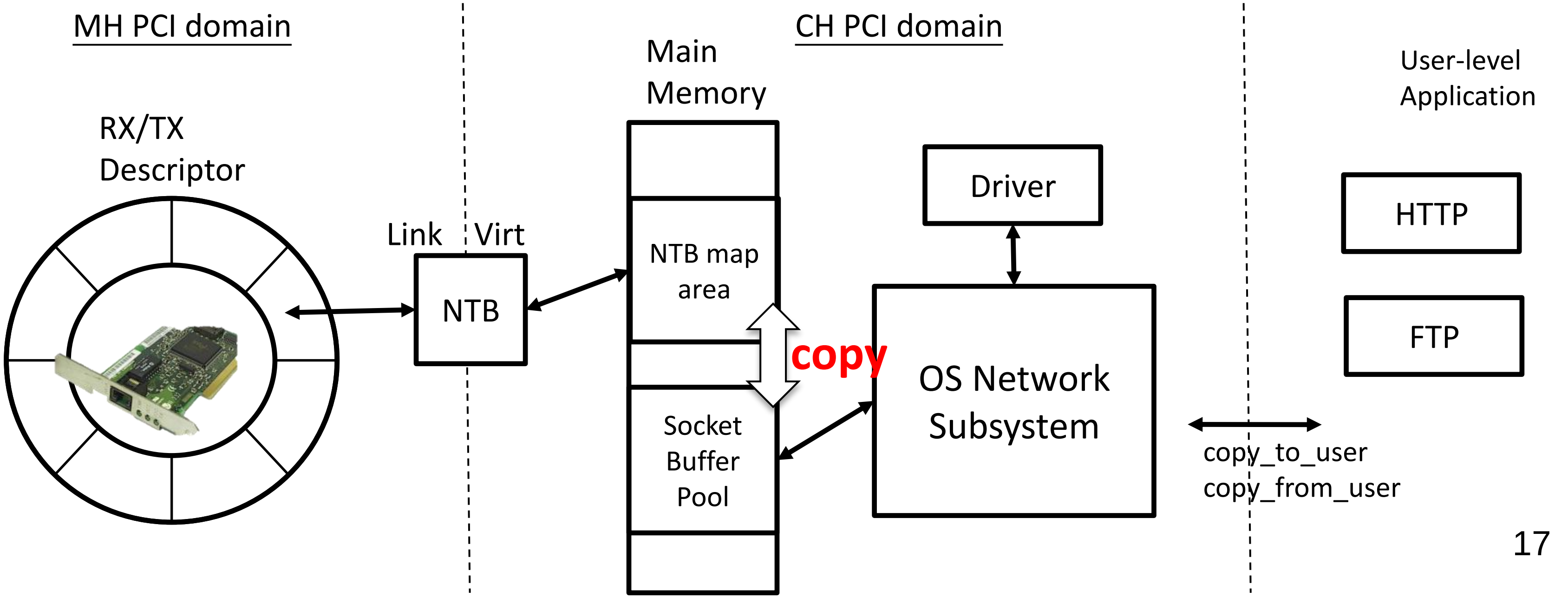
VF1 is assigned to VM1 in CH1, but it can screw multiple memory areas.

Security Guarantees: Summary

- Intra-host
  - A VF assigned to a VM can only access to the physical memory of its associated VM
  - Protected by CH’s IOMMU
- Inter-host
  - A VF can only access the physical memory of its associated CH
  - A host cannot touch other CHs’ or MH’s memory
  - Protected by CHs’ LUT & IOMMU
- Inter-VF / inter-device
  - A VF can not write to other VFs’ registers
  - Protected by MH’s IOMMU
- Even a compromised MH cannot touch other CHs’ memory

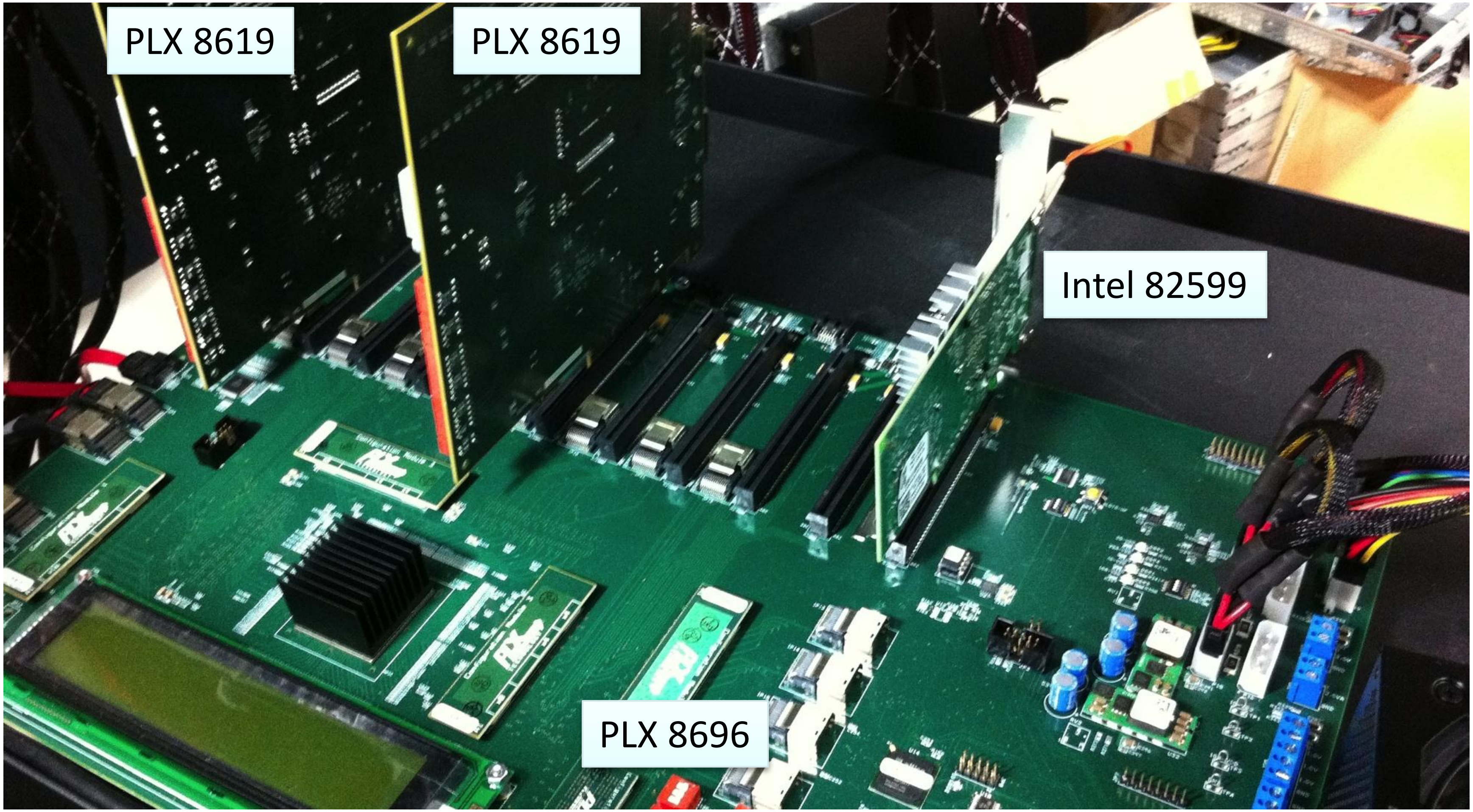
Optimizations

- Zero driver modification
  - DMA operands’ addresses are in MH’s address space rather than in CH’s or VM’s physical address space
  - Solution: transparent trapping of DMA access API calls
- Zero-copy
  - BAR window size is limited by BIOS (between 3G-4G)
  - Solution: Mapping the entire CH’s physical memory space to avoid copying.



Inter-Host Communications

- RDMA: Remote DMA from one host’s memory to another host’s memory
- Cross-Machine Memory Copying (CMMC): from the address space of one process on one host to the address space of another process on another host
- Ethernet over PCIe (EOP):
  - Destination node is inside the rack: sent out via PCIe
  - Destination node is outside the rack: sent out via shared Ethernet NIC

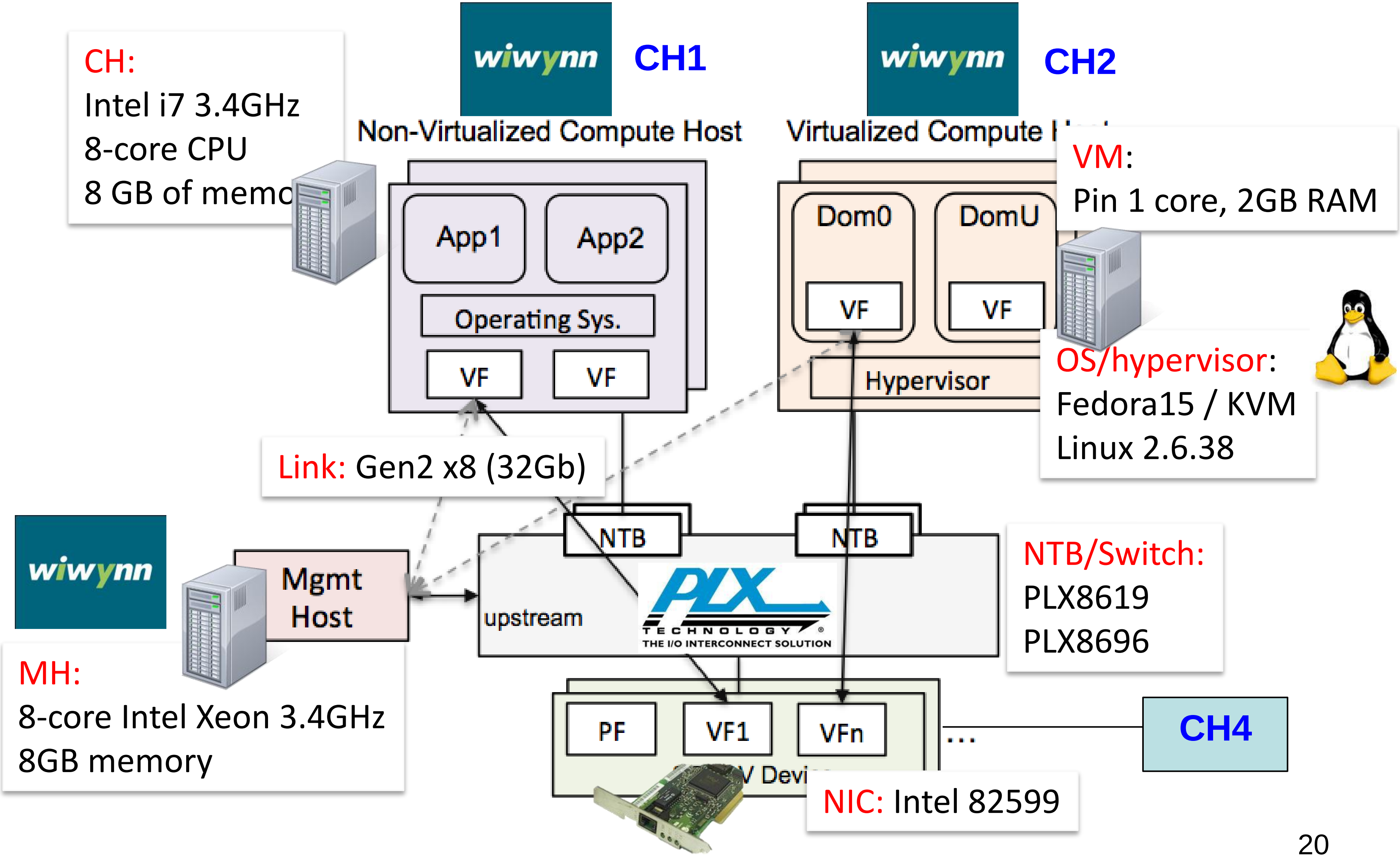


Let’s see some performance numbers!

TESTBED EVALUATION

Video Available: http://www.youtube.com/watch?v=B_-GesOjkG0

Prototype Implementation



CH:
Intel i7 3.4GHz
8-core CPU
8 GB of memo

VM:
Pin 1 core, 2GB RAM

OS/hypervisor:
Fedora15 / KVM
Linux 2.6.38

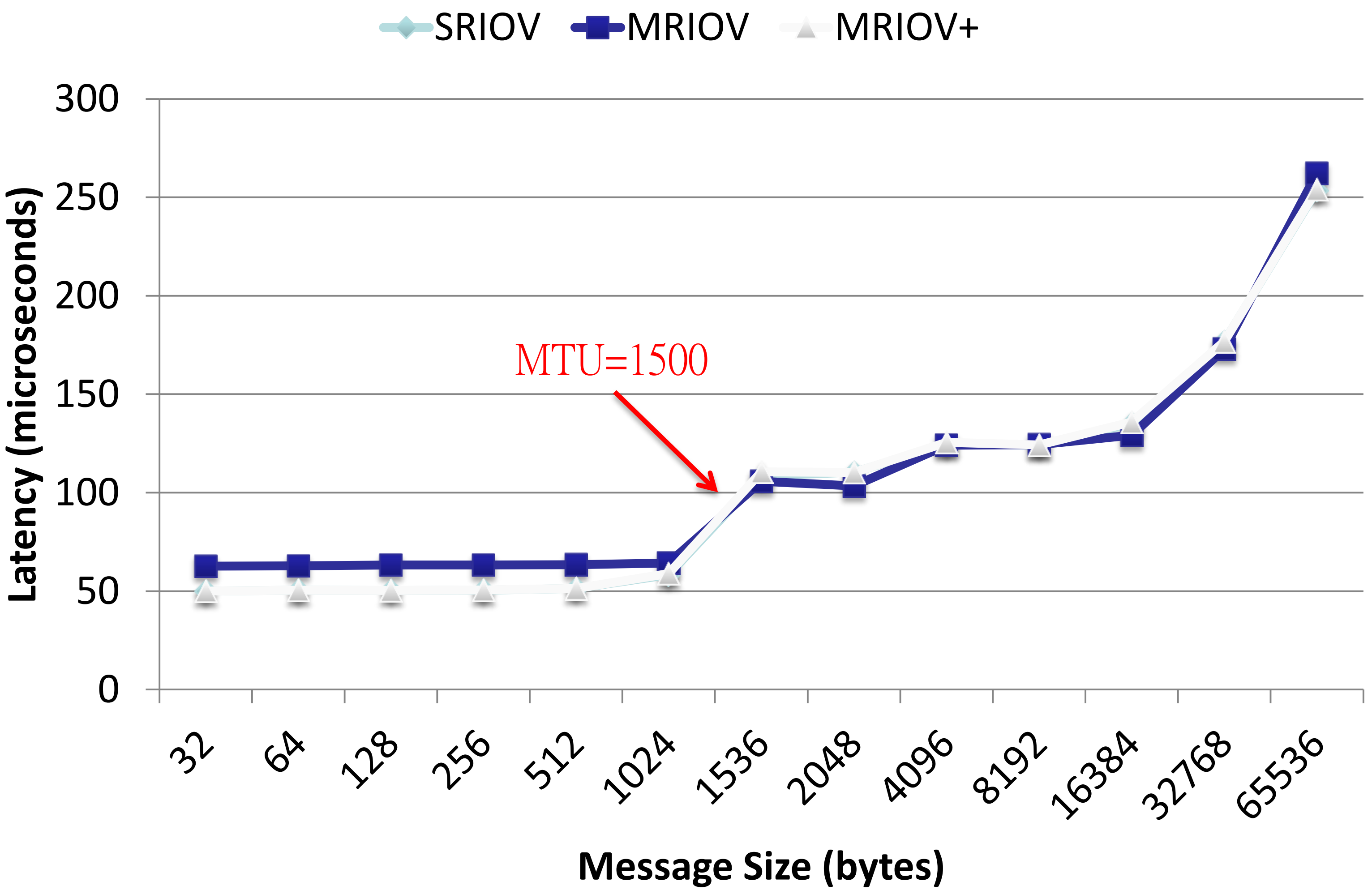
Link: Gen2 x8 (32Gb)

NTB/Switch:
PLX8619
PLX8696

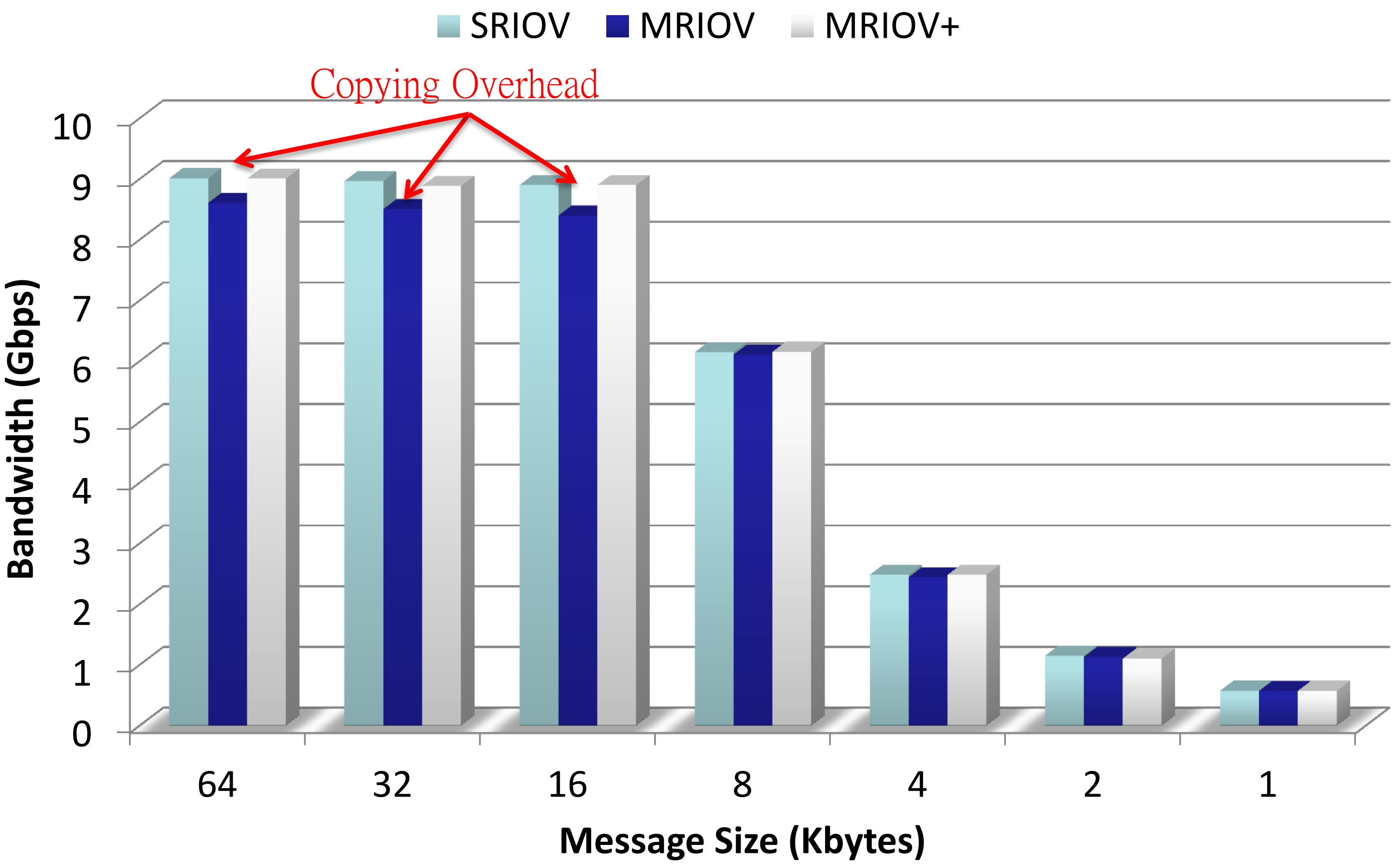
MH:
8-core Intel Xeon 3.4GHz
8GB memory

NIC: Intel 82599

Latency


SRIOV
MRIOV
MRIOV+
300
250
200
150
100
50
0
Latency (microseconds)
MTU=1500
32
64
128
256
512
1024
1536
2048
4096
8192
16384
32768
65536
Message Size (bytes)

TCP RX/TX Throughput



Scalability / Compatibility

- How many CHs/VMs can Ladon support?
  - Each CH: 2 BARs for CSR and MMIO, 2 BARs for DMA and MSI-X
  - As many VMs as a CH could run
  - As many CHs as are allowed by the PCIe switch
- How many SRIOV devices can Ladon shares?
  - As many as are allowed by the PCIe switch
- Vendor-neutrality
  - Ladon only uses the basic features of NTB: address translation, device ID translation and DMA
  - Intel's on-board NTB and DMA support: Xeon C5500/C3500
  - PCIe device: physical device driver needs modification but SRIOV driver does not

Summary

- A software-defined PCIe-based rack-area network architecture
  - Allows multiple servers to share and directly access SRIOV PCIe devices
- A secure I/O device sharing scheme
  - Prevents unauthorized accesses to shared I/O devices by leveraging EPT, IOMMU, BAR translation tables, and LUT
- A fully operational Ladon prototype
  - Successfully demonstrates the feasibility and efficiency of the software-based MRIOV
  - Zero throughput/latency penalty when compared with SRIOV
  - Full HA support for MH
- Integration of PCIe with Intel's silicon photonics technology?

Strawman Proposal for Disaggreated I/O

- Every host comes with an NTB, a DMA engine and a PCIe expansion adapter
- A transparent PCIe switch
- A set of SRIOV PCIe devices
- An MH or controller that
  - Enumerates the available SRIOV devices
  - Allocates and reclaims the virtual functions on SRIOV devices
  - Includes HA support
- On each CH
  - An original SRIOV device driver for each SRIOV device
  - A specially-built PCIe driver that communicates with MH

Thank You!

For More Information:

Y.F. Juan / 阮耀飛
Deputy Director
Strategy & Business Development
Cloud and Mobile Computing Center
Industrial Technology Research Institute
t: +886 (0) 3.591.6173
m: +886 (0) 975.876.919
e: yf.juan@itri.org.tw