

I/O Virtualization: A NEP Perspective

Version 1.0, February 12, 2009

Copyright © 2008 SCOPE Alliance. All rights reserved.

The material contained herein is not a license, either expressed or implied, to any IPR owned or controlled by any of the authors or developers of this material or the SCOPE Alliance. The material contained herein is provided on an “AS IS” basis and to the maximum extent permitted by applicable law; this material is provided AS IS AND WITH ALL FAULTS, and the authors and developers of this material and SCOPE Alliance and its members hereby disclaim all warranties and conditions, either expressed, implied or statutory, including, but not limited to, any (if any) implied warranties that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

Also, there is no warranty or condition of title, quiet enjoyment, quiet possession, correspondence to description or non-infringement with regard to this material. In no event will any author or developer of this material or SCOPE Alliance be liable to any other party for the cost of procuring substitute goods or services, lost profits, loss of use, loss of data, or any incidental, consequential, direct, indirect, or special damages whether under contract, tort, warranty, or otherwise, arising in any way out of this or any other agreement relating to this material, whether or not such party had advance notice of the possibility of such damages.

Questions pertaining to this document, or the terms or conditions of its provision, should be addressed to:

SCOPE Alliance,
c/o IEEE-ISTO
445 Hoes Lane
Piscataway, NJ 08854
Attn: Board Chairman

or

for questions or feedback, use the web-based forms found under the Contacts tab on www.scope-alliance.org

1. PURPOSE

The SCOPE Alliance was established early in 2006 to foster the growth of the supply chain ecosystem for a base platform comprising commercial off-the-shelf (COTS) carrier-grade hardware, operating systems, and middleware. Virtualization technology, especially CPU/memory virtualization using virtual machine monitors (VMMs), is attracting a lot of interest, and the SCOPE Alliance has already published two documents discussing the use of virtualization technology by network equipment providers (NEPs). Multiple virtual machines (VMs) sharing CPU/memory resources can operate independently with a secure partition between each VM and the others.

As the virtualization of CPU/memory becomes a more mature technology, poor performance and controllability of I/Os becomes a bottleneck. This problem is more serious for network equipment (NE) than for servers, because NE is I/O intensive. Network equipment generally contains many kinds of interface cards that use various protocols and operate at various speeds. In addition, when a system is I/O intensive, the aggregation/consolidation type of virtualization as well as the sharing type of virtualization becomes important.

Considering I/O virtualization from a network equipment viewpoint, two types of I/O virtualization can be distinguished. One is mainly for server-based NE and uses VMMs. Direct control of I/Os from each VM is a main topic. The other is an aggregation/consolidation type of virtualization. This paper first introduces five technologies and then describes management and software issues, such as intervening real and virtual resources and service continuity with I/O reconfiguration; it then discusses the use of these technologies in a carrier-grade system and identifies a number of issues not currently addressed by industry standards.

This paper also describes the landscape of I/O virtualization and is intended to make network equipment providers and ecosystem vendors more aware of this technology.

2. AUDIENCE

This document is intended for the following audiences:

- Developers of carrier-grade service-availability middleware services
- Third-party component developers, suppliers, and consumers
- Developers, suppliers, and consumers of integrated product platforms, including Hardware and I/Os
- Developers of service availability interface specifications

3. REFERENCES

1. SCOPE ALLIANCE documents on virtualization are available at

<http://www.scope-alliance.org/pr/SCOPE-Virtualization-Requirements-Version-1.0.pdf>,
<http://www.scope-alliance.org/pr/SCOPE-Virtualization-StateofTheArt-Version-1.0.pdf>,
<http://www.scope-alliance.org/pr/SCOPE-Virtualization-UseCases-Version-1.0.pdf>

2. AMD I/O Virtualization Technology (IOMMU) Specification. Available at http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/34434.pdf
3. Intel Technology Journal Vol. 10, Issue 3, Intel Virtual Technology for directed I/O, 2006 Aug. 10th. Available at <http://download.intel.com/technology/itj/2006/v10i3/v10-i3-art02.pdf>
4. I/O Virtualization in PCISIG. Available at <http://www.pcisig.com/specifications/iov/>
5. PICMG specification of AMC. Available at <http://www.picmg.org/v2internal/AdvancedMC.htm>
6. PICMG specification of u-TCA. Available at <http://www.picmg.org/v2internal/microTCA.htm>
7. An example of virtual router which supports resource slicing in <http://www.planet-lab.org/files/presentation-2007-05-01-planetlab.ppt>
8. Virtualization for embedded system. Refer to discussion in <http://www.power.org/home>
9. Backplane Ethernet project in <http://grouper.ieee.org/groups/802/3/ar/index.html>
10. Ethernet-Level Congestion Management project in <http://www.ieee802.org/1/pages/802.1au.html>
11. Fibre Channel over Ethernet (FCoE); proposed mapping of Fibre Channel over selected full duplex IEEE 802.3 networks <http://fcoe.com/>
12. I/O consolidation work. See, for example, <http://h18013.www1.hp.com/products/blades/components/ethernet/vcem/index.html>
13. I/O consolidation work. See, for example, <http://publib.boulder.ibm.com/infocenter/systems/scope/hw/index.jsp?topic=/iphb1/iphb1kickoff.htm>
14. PCI-Express switch over Ethernet. Proposed in. <http://www.expether.org/>
15. Path Computation Element <http://www.ietf.org/html.charters/pce-charter.html>
16. RapidIO Trade Association www.rapidio.org/
17. InfiniBand Trade Association www.infinibandta.org/
18. Open Flow Switch <http://openflowswitch.org/>
19. Ethernet-Level Link Aggregation in <http://grouper.ieee.org/groups/802/3/ad/index.html>
20. Availability calculation work in academics. See for example, N. Kami et. al. Multilayer In-service Reconfiguration for Network Computing Systems [Sixth IEEE International Symposium on Network Computing and Applications \(NCA 2007\)](#) pp. 324-331

21. PCI-SIG's compliance workshop http://www.pcisig.com/events/compliance_workshop/

22. Distributed Management Task Force <http://www.dmtf.org/home>

4. TERMS AND DEFINITIONS

ATM	Asynchronous Transfer Mode
ATS	Address Translation Service
API	Application Programming Interface
BAR	Base Address Register
CIM	Common Information Model
CPU	Central Processor Unit
DMA	Direct Memory Access
DMTF	Distributed Management Task Force
DVA	DMA Virtual Address
FC	Fibre Channel
FIB	Forwarding Information Base
FLR	Functional Level Reset
FRU	Field Replaceable Unit
GE	Gigabit Ethernet
HBA	Host Bus Adapter
HPA	Host Physical Address
INT	INTerrupt
I/O	Input Output
IOMMU	Input Output Memory Management Unit
MRA	Multi-Root Aware
MR-IOV	Multi-Root I/O Virtualization
MSI	Message-Signaled Interrupts
NE	Network Equipment
NEP	Network Equipment Provider
NFS	Network File System
NIC	Network Interface Card
NWP	Network Processor
OS	Operating System
PCE	Path Computation Element
PCI	Peripheral Component Interface
PCIM	PCI Manager
PE	Processing Element
PF	Physical Function
QoS	Quality of Service
RASD	Resource Allocation Setting Data
SBC	Single-Board Computer
SR-IOV	Single-Root I/O Virtualization
SVPC	System Virtualization Partitioning and Clustering
TOE	TCP Offload Engine
TLP	Transaction Layer Packet
UML	Unified Modeling Language
VF	Virtual Function

VH	Virtual Hierarchy
VMM	Virtual Machine Monitor
10GE	10-Gigabit Ethernet

5. INTRODUCTION

5.1 Virtualization Technology

Virtualization is one of today's most important technologies in servers and in processing-intensive network equipment because CPUs typically have more power than a single user can use [ref 1]. With virtualization technology, one can consolidate physical resources to reduce power consumption, maintenance and management costs. In addition, by loosening the binding of services to the physical resources providing those services, virtualization increases reconfiguration flexibility. The potential of virtualization technology to increase reliability, availability, and serviceability is thus attracting attention of service providers as well as of system vendors.

From the system point of view, virtualization is a technology that abstracts physical resources to generate logical resources. Two types of virtualization can be distinguished based on whether the single resource is a physical or a logical resource (Fig. 1). The former is the "share" type of virtualization. The virtualization mechanism makes multiple virtual resources and provides them to the upper layer. A virtual machine monitor (VMM) (or hypervisor) is the typical example of this "share" type virtualization. A VMM controls the CPU scheduler to cut the CPU time into slices and provides them to the virtual machines (VMs) as virtual CPUs. That is, a single physical CPU resource is virtualized into multiple logical CPUs and shared by multiple VMs. Secure partitioning is a key to "share" type virtualization at the hardware level.

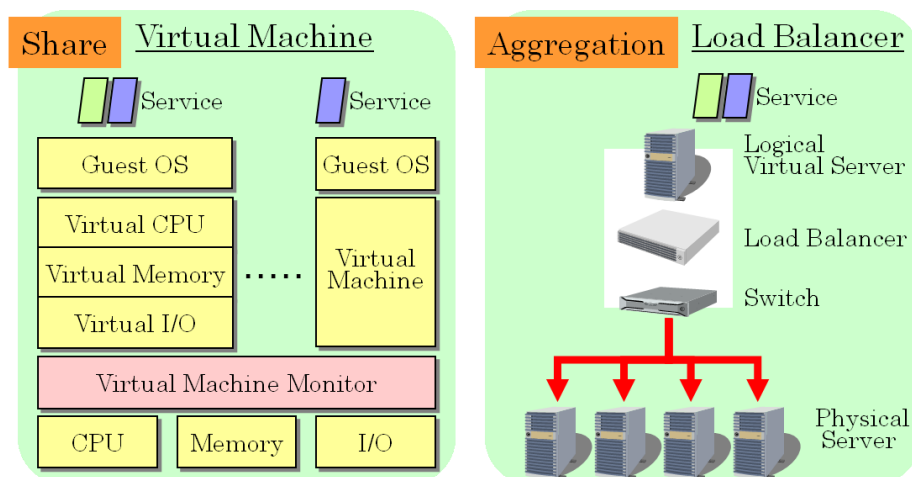


Fig. 1 Two Types of Virtualization: Share and Aggregation

The other type of virtualization is the "aggregation". The virtualization mechanism makes a single virtual resource out of multiple physical resources in the resource pool. Load balancing is the typical example of "aggregation" virtualization. A load balancer presents a single server to the accesses from a network and then distributes the accesses and their counter processing to multiple servers in a server pool. Seamless

reconfiguration is a key to the “aggregation” type of virtualization, which is usually provided by a software layer.

5.2 I/O Virtualization from the Network Equipment Provider’s Perspective

First, the definition of I/O from the viewpoint of the network equipment provider (NEP) will be considered.

The term “I/O” comes from the traditional computer architecture. Its basic element is a processing element (PE) consisting of a CPU and memory connected through a memory controller hub, which is often called a northbridge; the other elements in this architecture are input and output (I/O) devices, which perform input and output (I/O) operations for the PE, as shown in Fig. 2. These I/O devices are typically graphics cards, storage, network cards, keyboard, mouse, and so on. All of these devices respond more slowly than the memory in PE. Although I/Os are still mapped in a CPU-accessible address space, the reading and writing of data is done as direct memory access (DMA), and not by the CPU itself, but rather by an I/O controller in a chipset, so that the CPU does not have to wait for the slow response of the I/O itself. This definition of I/O is closely related both to the computing–intensive network equipment that handles the control plane and to network appliances whose operations are mainly on the software plane.

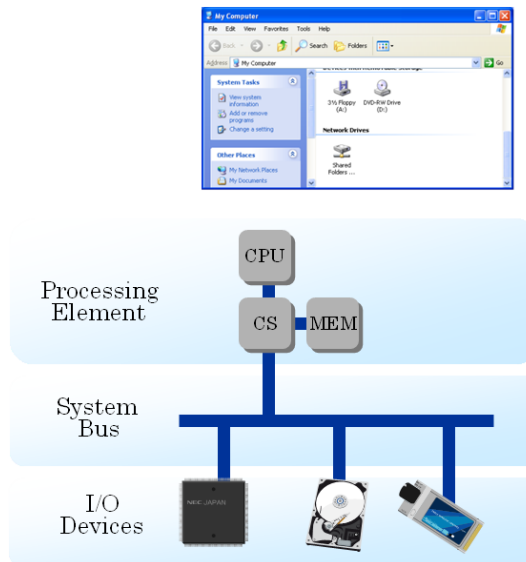


Fig. 2 I/Os in Traditional Computer Architecture

There are two kinds of I/O virtualization in this computing–intensive architecture. One is the I/O access/control method used to support CPU and memory virtualization, that is, to assure that a virtual machine monitor (VMM) has good I/O performance and dedicated control of I/Os. The pertinent functions are provided by an input/output memory management unit (IOMMU). AMD’s IOMMU [ref. 2] and Intel’s VTd [ref. 3] are good examples of this type of I/O virtualization. The other kind of I/O virtualization is the virtualization of I/O for the CPU/software, which is done by presenting to the CPU/software a partly abstracted logical entity of I/O and by sharing this logical entity among instances of the CPU/software. PCI-SIG’s I/O virtualization is a typical example. It enables a single I/O resource to be shared by multiple VMs either in a single host (SR-IOV) or in multiple hosts (MR-IOV) [ref. 4].

One key issue in this area is how to present virtual I/Os to CPU/software without severe degradation of the performance. The main topics are DMA remapping (that is, address transfer), interrupt remapping, and virtual configuration.

Another kind of I/O virtualization for computer-based architecture is virtualization at the system-bus level, in other words, at the interconnection, to make an I/O consolidated system. The typical system is a blade server with a CPU blade that has a minimum of I/O devices directly connected to it; however, the CPU card has also an interconnection that acts as a big pipe to the I/O concentrator located in another card (AMC carrier [ref. 5]) or in another chassis (Micro-TCA [ref. 6]). This I/O-consolidation architecture enables a flexible I/O assignment to the CPU blade. It is very valuable for server consolidation, because it reduces capital expenditure for I/Os, increases utilization, and supports system reconfiguration to cover many services and their dynamic changes.

All the systems mentioned above are computer-based systems, which are mainly used to provide control-plane services. From the perspective of NEPs supplying data-plane and user-plane equipment, interface cards and line cards are considered to be I/O devices. Therefore, from a NEP perspective [ref. 7], a virtual router that supports resource slicing to provide a multi-tenant virtual-operator solution should also be included in the I/O virtualization. These I/O virtualization technologies are mapped as shown in Fig. 3.

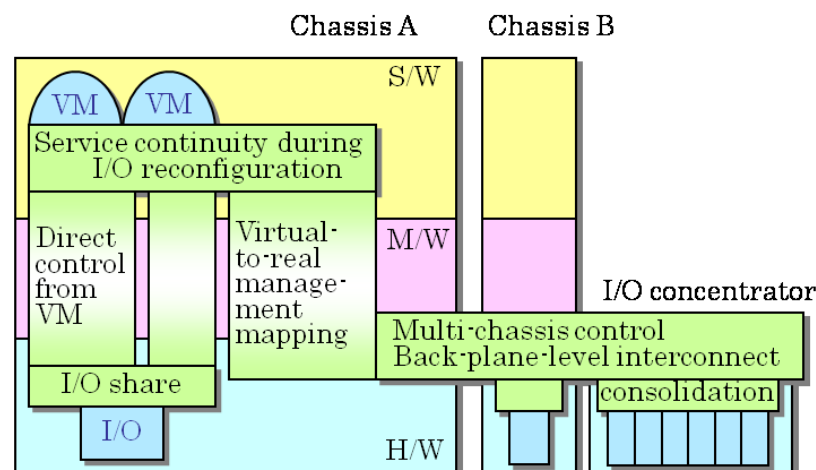


Fig. 3 I/O Virtualization Technology Mapping

5.3 Brief Summary of I/O-Virtualization Technologies

The five I/O-virtualization-related technologies mentioned above are briefly described in this section.

Table 1 provides a comparative summary.

1. The key technology of the IOMMU for the I/O virtualization is to allow the creation of multiple VM-specific address spaces for I/O access from VMs. An example of it is the VTd technology, which performs DMA/interrupt remapping in the northbridge chipset. It enables a VM to drive I/O directly, hence improving the I/O performance and simplifying I/O control in the VMM. However, software protection in the IOMMU can be bypassed by this technology. Today, the work for VM-direct I/O access continues on caching virtual configuration, off-loading address translation, and so on.

Power.org has an Embedded Virtualization technical subcommittee that is working on standards for embedded interface virtualization, including virtual I/Os [ref. 8]. Many embedded systems require real-time operation, even for VMs. For that purpose, at least one core is assigned to a VM, and, in that partition, a dedicated I/O resource is assigned to the VM and directly accessed from the VM through I/O-interrupt direct communication.

2. The key technology of the SR-IOV is making possible that multiple VMs on a single host share a standard PCI device through "Virtual Functions" (VF). The VMs use these functions to access the PCI device, and SR-IOV maps them to physical functions that the physical hardware controls and sees. This technology is implemented in the endpoint device and, in some cases, in the driver software. In order to achieve good performance under the shared I/O, the internal congestion should be avoided, and, in some cases, QoS is desirable. Although specification version 1.0 is released to establish a "standard", it does not describe its implementation so much. There needs to be some consensus on implementation among I/O hardware vendors and system vendors to assure interoperability.
3. The key technology of the MR-IOV is "Virtual Hierarchy" (VH), which enables multi host to a single I/O connection while maintaining PCI-Express's tree-topology hierarchy. This technology needs conversion in all PCI elements, northbridge, switch, endpoint, and PCIM (PCI Manager). Although many system developers need this technology, implementation and management/configuration complexity makes this a tall barrier to overcome.
4. The key technology of the I/O consolidation is in the interconnection layer to enable the system to use arbitrary I/O, not only in the local system but also in a remote location. An interconnection protocol, an interface card or bridge chip should be included in the system, which increases the cost. Therefore, low-cost standards-based interconnection is desirable. There are numerous systems using Ethernet as an interconnection. IEEE802.1Qau and 802.3ar work mainly for this purpose [refs. 8-11]. Software can also take a role in the I/O consolidation by presenting to a VM/OS logical I/O resources consisting of multiple physical I/O resources. Usually, this function is realized through CPU and storage with an implementation as an appliance [refs. 12, 13].
5. The key technology of the virtual slicing relies on the capability of a slicing switch and of its controller. This technology enables both slicing of switch and interface cards (I/O) to multi-tenant systems, maintaining secure partitioning. In some implementations such as Path Computation Element (PCE), the controller is concentrated and located remotely [ref. 15]. For this case, the security of the control interface is very important and standardization of the control API is also necessary.

Table 1 Brief Summary of Five I/O-Virtualization Technologies Today

	VTd	SR-IOV	MR-IOV	I/O Consolidation	Virtual Slicing
Technology	DMA/Interrupt remap	Virtual Function	Virtual Hierarchy	Interconnect	Resource Slicing
Location	Northbridge	Endpoint	Northbridge/Switch/Endpoint	Switch/Bridge	Switch/Controller
Merit	VM direct I/O	VM-direct I/O Share	Multi-Host I/O Share	Reconfigurability	Programmability
Weak Point	Security without IOMMU Protect	Internal Congestion	Configuration Complexity	Bridge Cost	Security at Control Interface
Future Work	Cache, ATS	Implementation Consensus	Implementation	Standard-Based Reliable	Standard Effort

6. I/O VIRTUALIZATION TECHNOLOGY

The five technologies are further elaborated in this section.

6.1 I/O Virtualization to Support VMM-Based Virtualization Technology

The purpose of this I/O virtualization type is to support VMM technology. Today's VMM technology is developed to share abundant processing power among multiple users. The main mechanism is implemented in the CPU scheduler in order to supply a time-slice in a certain period to a virtual CPU on which a virtual machine can be operated. The greatest benefit of this hardware-based virtualization is that each VM is securely partitioned. It enables server consolidation with multiple users/tenants on a single server.

When one considers the implementation limits and the costs, it makes no sense to assign each physical I/O resource exclusively to a VM. Naturally, the VMs must share the local I/O resources on the platform.

There are several software-level solutions to the problem of sharing a physical I/O resource among multiple VMs. One is to perfectly emulate the physical I/O in the driver software for the host OS and have the VMM show the emulated I/O to the guest OS on a VM. Because the physical I/O is emulated perfectly, the guest OS can use a normal driver for the I/O. Emulating all hardware is not realistic, however, because it requires too much development effort and because a software emulator has considerable overhead, which degrades the I/O performance.

The second solution is called paravirtualization. A special VM is dedicated to manage and control all I/Os as well as read and write data from and to the physical I/Os. This I/O-specialized VM and other VMs are connected through a virtual bus in the VMM. Because the read/write process is operated through a real driver of the physical I/O, with a pre-execution of a VM's I/O request to the specialized VM, the I/O performance is much improved. The I/O drivers in the guest operating systems, however, have to be modified to drive I/Os through the virtualized I/O bus.

The third solution, called full virtualization or direct I/O, supports the VM's direct I/O access to the physical I/Os. This solution reduces the software overhead and is thus expected to provide better I/O performance. However, full virtualization requires hardware support in both the northbridge chipset and the I/O resource itself. The former is the VTd and the latter is PCI-SIG's SR-IOV.

Because the I/O mechanism is so closely connected to the CPU and to the memory, the software can use it through normal memory access, and I/O virtualization must take into consideration the following four items: (1) discovery of I/O devices in the boot stage

and hot-plug-in stage, (2) configuration of the I/O devices (such as translation of the base address register), (3) data transfer by DMA, and (4) interruption for notification of events, such as write-completion or changes in the state of the I/O device.

6.1.1 VTd

VTd is a subset of Intel's VT (virtualization technology) series. VT provides a mechanism enabling a VM to change the CPU state directly to ring 0 because the state change for making a hypercall for handling an interrupt was becoming a bottleneck. The version for a Xeon processor is called VTx, and the version for an Itanium processor is called VTi. To support direct I/O, VTd architecture allows for DMA and interrupt requests from I/O devices to be isolated in a specific protection domain. Additionally, it provides DMA remapping and interrupt remapping mechanisms as shown in Fig. 4.

1. *DMA remapping*: Security and reliability are assured by restricting a system memory access from the I/O device to a protection domain. The system software can assign multiple I/O devices to a protection domain through a device-mapping structure that uses PCI's bus, device, and function numbers as keys to each device's address translation tables. A DMA request is applied by using the DMA virtual address (DVA) of the guest physical address of the VM to which the I/O device is assigned and is then translated to a host physical address (HPA). These address transfer mechanisms are implemented in a northbridge chipset as shown in Fig. 4.
2. *Interrupt remapping*: To support legacy interrupts, message-signaled interrupts (MSIs), and MSI-X across the protection domain, an interrupt message identifier is used instead of interrupt attributes written directly in the interrupt message issued by the I/O device. Like a MSI, the interrupt is issued as a DMA write request but contains only the message identifier. Then, the chipset remaps the interrupt by the requester ID and associates to the remapped interrupt all the necessary interrupt attributes (destination processor, delivery mode, etc.) derived from the message identifier.

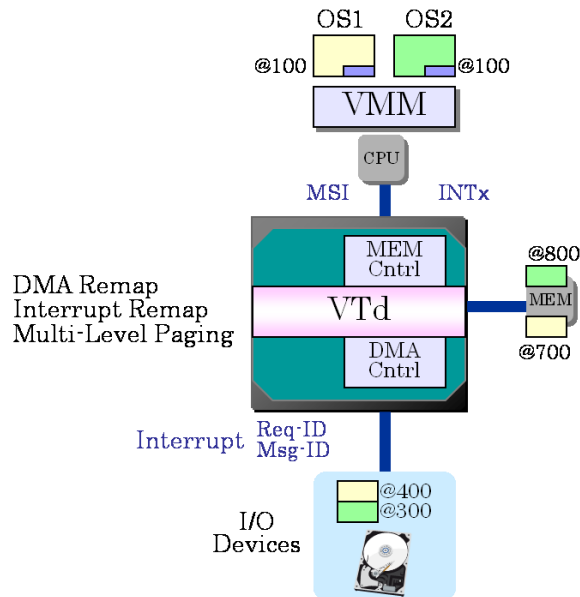


Fig. 4 VTd Performing DMA/Interrupt Remap and Multi-Level Paging

6.1.2 PCI IOV

As VTd enables a VM to directly access an I/O device, the I/O device should be prepared for simultaneous accesses from multiple VMs. Single-root I/O virtualization (SR-IOV) is the specification for this purpose, and multi-root IOV (MR-IOV) not only extends the specification to a multi-host environment but also allows a legacy OS as well as a guest OS on VM.

1. PCI-IOV: PCI Express has a hierarchy from the root complex (corresponding to the northbridge and southbridge chipsets) to endpoint I/O devices through switches, maintaining a tree-topology. This means that an I/O device is dedicated to a single system image (roughly corresponding to an OS). The main concern of the PCI-IOV is, therefore, how I/O devices from multiple system images (SIs) can be shared in a way that has the least effect on today's PCI Express standard.
2. SR-IOV: SR-IOV enables I/Os to be accessed from multiple SIs (corresponding to VMs) on a single host, as shown in Fig. 5. The tree topology is physically maintained with only one root complex at the top. Therefore, the necessary change is not so significant. The easiest way to allow this access from multiple SIs is by modifying I/Os to have a multiple interface as a virtual function (VF) and to have intervening software mediate between the multiple VMs and multiple VFs. When this method is used, only the I/O device (and its driver) should be converged. The root complex and switches can be used as they are. Making a VF does not require a big change in the I/O device because a "function" is originally defined in the PCI standard. Each function has its own configuration, memory address space, base address register (BAR) and supports INTx, MSI, and MSI-X interrupts. SR-IOV defines a physical function (PF) corresponding to an original function, and, in that field, VFs are made by sharing the PF's BAR set. The operation of the VF is almost the same as that of the original func-

tion. It responds to a PCIe configuration transaction and memory transaction targeted to the VF and also issues a PCIe transaction for itself.

VF supports the same page size of “functions”: from 4KB to 4MB. When the system is booted, there is only one PF. Then, during IOV configuration, each PF’s SR-IOV Extended capability, (VF enable, VF Migration Capable/Enable) is set and VF is created by the SR-PCIM. A functional level reset (FLR) that targets only a VF resets the VF. A FLR that targets a PF resets the PF and all its associated VFs.

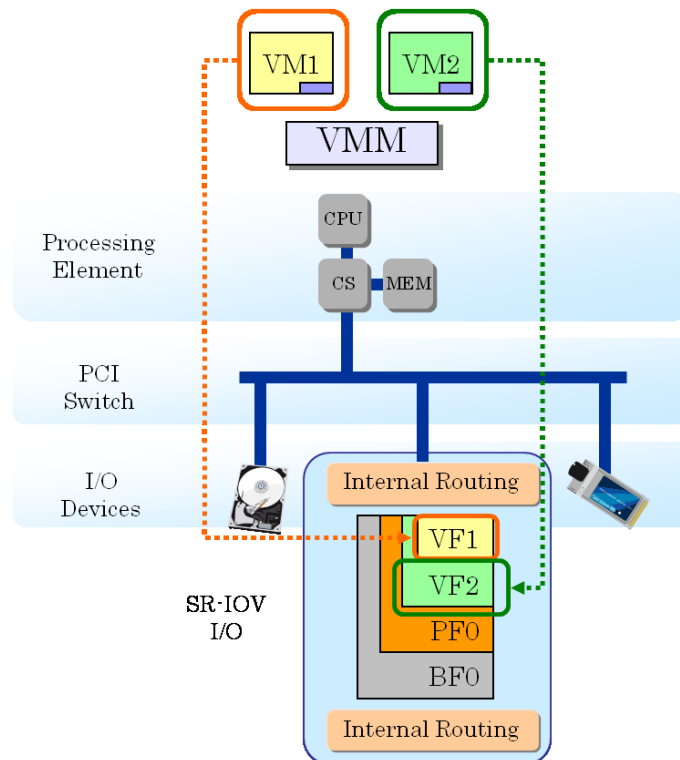


Fig. 5 Single-Root IOV

- MR-IOV: Sharing I/Os from multiple root complexes is a big change for the PCI-Express standard because it conflicts with the tree-topological hierarchy of PCI Express. Therefore, MR-IOV defines a virtual hierarchy (VH) that logically partitions the PCI Express fabric to share the same physical hierarchy, including switches and endpoints, as shown in Fig. 6. A new Transaction Layer Packet (TLP) header and a new training sequence (like the Ethernet auto-negotiation procedure) are defined. Therefore, all components must be converged to an MR-aware (MRA) one, the root-complex, switches, PCIM, and endpoints. No device vendor has announced the release of MRA endpoint I/Os yet, but some proprietary methods for MR sharing have been reported.

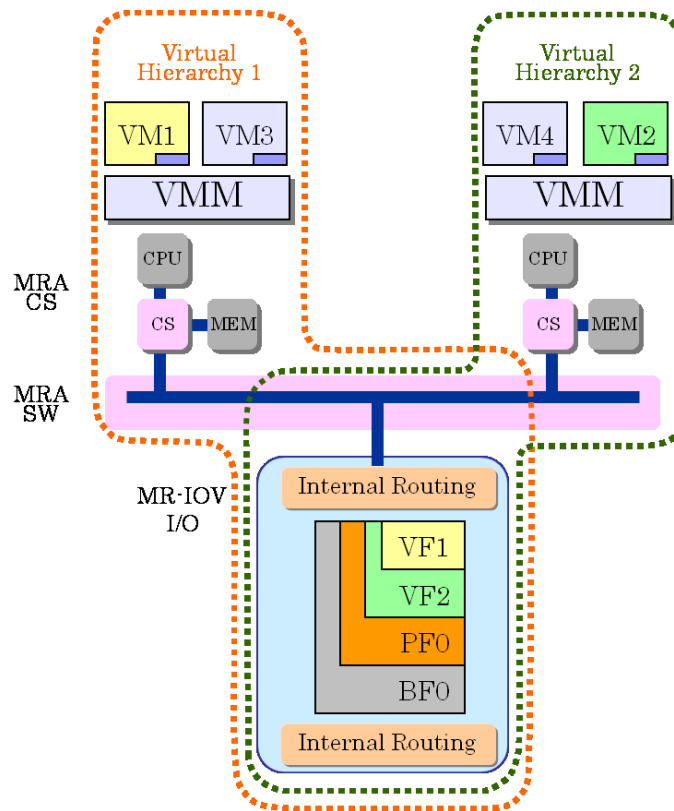


Fig. 6 Multi-Root IOV

4. ATS: Address translation service (ATS) offloads the remapping of DMA address from the chipset to the I/O itself. The root complex needs only to recognize from a flag in the TLP header whether the address is pre-translated. The system software plays an important role: it has to ensure that the I/O can bypass the protection of the memory management unit (MMU); it must also enable ATS and confirm the coherency of cache translation.

6.1.3 I/O-Consolidation-Type Virtualization

I/O consolidation is a type of I/O virtualization that uses interconnection-level technology. There is no standard work for this type of I/O virtualization on a whole system. Some proprietary systems realize I/O consolidation by software-assisted interconnection-level technology. An example of I/O-consolidated network system is shown in Fig. 7.

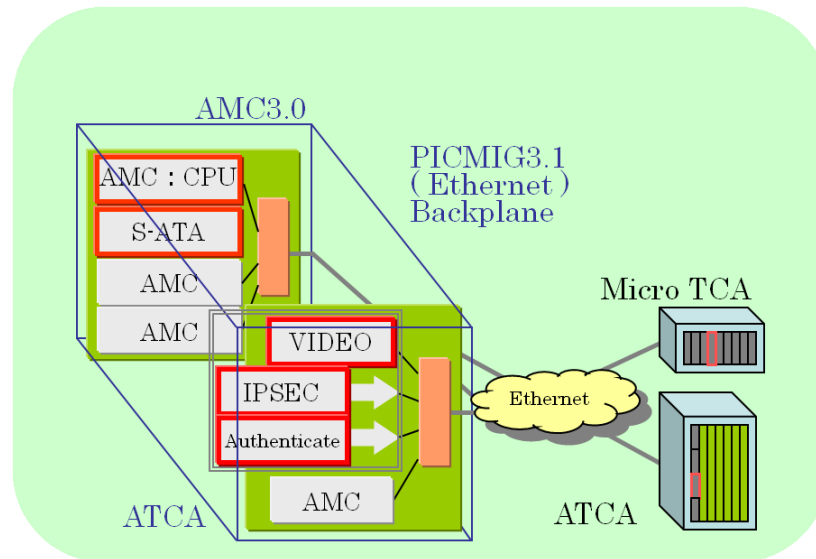


Fig. 7 I/O Consolidation-Type Virtualization

A logically single system is composed of I/O hardware in the I/O pool such as a u-TCA [ref. 6] or AMC carrier cards [ref. 5] that sustain multiple AMC I/Os. This is an “aggregation” type of virtualization. The key technology is to make a system-bus-level interconnection among I/Os resources that are distributed over multi-cards and multi-chassis. A set of I/O resources can be grouped such that they present a logical I/O resource for systems that are also attached to the interconnect. I/O resources in such a group can communicate with each other via the interconnect but are protected from invalid accesses coming from outside the group or from systems that have not allocated the group. Although it is not strictly necessary, it is desirable that an I/O resource be shared from multiple SBC (Single Board Computer) or CPU-AMC modules as hosts. Some hosts have a VMM to run network services in VMs to support legacy service in legacy OS.

PCI Express is suitable for making such a close connection. However, to make a flexible multi-CPU-to-multi-I/O connection, MR-IOV or other proprietary method (such as using non-transparent port for shared memory or PCIe over Ethernet [ref. 14]) should be combined with the original PCI Express standard. Rapid I/O [ref. 16] and Infiniband [ref. 17] are also capable of DMA-type transfer to make a close connection.

This I/O-consolidated system provides the following benefits:

- **Scalability.** The system can use multiple I/O resources from the resource pool of u-TCA or AMC carrier cards to up-scale or out-scale the performance of the services. Even if the I/O resources are located remotely, the service software can use them as a local resource, because the interconnections are close connections.
- **Reconfigurability.** Performance can be scaled in-service by performing a hardware reconfiguration. The service software is not affected by hardware reconfiguration, because middleware or a VMM can hide it to sustain service processing during the reconfiguration. I/O resources can be hot-plugged into the system or removed from it even if they are located remotely. Hot plugging can be done manually or by middleware by just changing the I/O group definition.

- **Availability.** The system is highly available because it can use redundant I/O resources even if they are located remotely. The distributed and redundant architecture increases the availability because most single points of failure have little effect on other cards. For example, if all the SBC and I/Os resources constituting the system are located in a single chassis, a failure of the electrical power supplied to that chassis brings the whole system down. The example system, however, survives such a power failure by using resources in another chassis.
- **Power saving.** The system can reduce power consumption by using VM migration and I/O migration together. Similar to the case of an enterprise network, the processing power and the networking power reduce at night. Then, consolidating those services into a low-performance resource with low power-consumption is an effective way to save electrical power. This example system can use VM migration as well as I/O migration for this purpose. This means, the destination resources of migration can be chosen and powered-on just when the migration process is necessary. The distribution of I/Os also facilitates heat dissipation because all hardware resources do not have to be in a single chassis. Power consumption for cooling can thus be greatly reduced.

6.1.4 I/O Virtualization for Data-Plane Equipment

It may be reasonable for NEPs providing data-plane and user-plane equipment to consider interface cards and line cards as I/O devices. Then, a virtual router that supports resource slicing to provide a multi-tenant virtual operator solution should be included in I/O virtualization.

A typical implementation of the virtual router is shown in Fig. 8. On the top, a control CPU manages resources and treats unknown packets. For multi-tenant use, resource slicing capability is important. At the CPU level, a VMM can secure the partitioning of CPU and memory resources. Therefore, the switch fabric should also be partitioned by having separated routing tables for multiple-controller VMs. This is why the line card should have multiple interfaces for the internal switch fabric and for the outer port. The identifiers of the virtual port can be arbitrary, depending on the internal switch fabric. If it is PCI-IOV, it should be VF. Each VF, thus, has its own MAC address for the outer port.

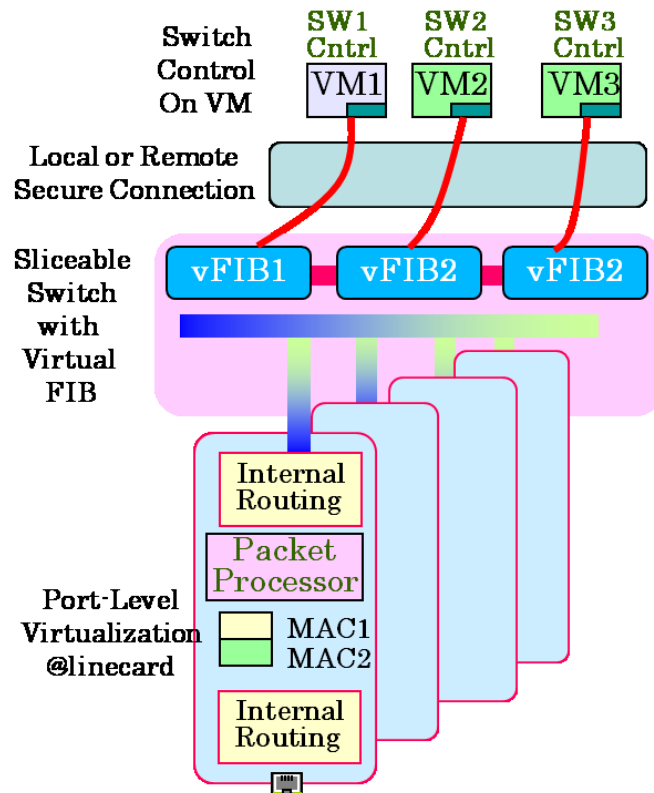


Fig. 8 Virtual Router with Sliceable Switch-Fabric and Line Cards

Although the controlling interface of the switch fabric is mostly proprietary, some switch manufacturers have attempted to define an open interface in the OPEN FLOW SWITCH Consortium [ref. 18]. Although this open interface is intended for use in experimental networks, it can be used to control virtual slicing.

7. MANAGEMENT AND SOFTWARE ISSUES

To provide services on an I/O-virtualized system, management and software issues have to be considered.

Although virtualization is thought to be valuable to reduce management cost, virtualized systems introduce complexity in resource management and performance monitoring. This is because the resources used for a certain service are logical resources, but the service is executed on the physical resources. Therefore, a special way to manage virtualized systems has to be considered. There is some work ongoing in DMTF (Distributed Management Task Force) [ref. 22]. It defines I/O virtualization architecture and model with a standard interface.

Fig. 9 shows the management scheme concept of the virtualized system. DMTF uses CIM (Common Information Model), which is based on UML (Unified Modeling Language). In CIM architecture, a system is described as a schemer, which consists of a group of objects and their relations. To treat virtualized systems, SVPC (System Virtualization Partitioning and Clustering) workgroup of DMTF extends its framework to have a physical resource and its abstraction as a logical resource. Resource virtualization has been defined using virtual system representation in a core CIM class that defines the

fundamental attributes of the resource, then adding information to the virtual system configuration describing resource allocation setting data (RASD). RASD includes information related to the virtual to physical association, such as resource-pool ID, host (physical) resource, and allocation units. The virtualized system configurations are made using those abstracted resources. The system service management actions defined are “define” (generate) and “destroy” the system, “adding”, “removing”, and “modifying” the settings of the system or resources. The allocated resources are derived from a resource pool, which can be hierarchically arranged.

SVPC continues to issue profiles of resource allocation, allocation capabilities, system virtualization, virtual system, generic resource, processor and memory resource allocation, network port allocation, and so on, to cover all resources and their allocation profile in a computer system.

Management middleware’s function on the virtual-to-real resource mapping framework mentioned above should be very wide. For example, the middleware runs keep-alive monitoring, and if it detects the failure of a virtual I/O, it needs to identify and isolate the physical I/O resources responsible for the failure. The management must, therefore, be performed in a complex distributed and cross-layer manner.

Hot plug-in/out management of virtual I/Os as well as aggregation and partitioning control have also to take into account virtual-to-real mapping. If the I/O resources are distributed on more than one chassis, one should carefully consider reliability, scalability, and cost when deciding whether a carrier-grade management system should operate in a concentrated or in a distributed manner.

Finally, because service software runs on such I/O virtualized systems, service continuity, even in the I/O reconfiguration process, is important. Distributed processing on a multi-blade and multi-chassis system might be useful for making scalable and reliable systems exploiting the advantages of redundancy.

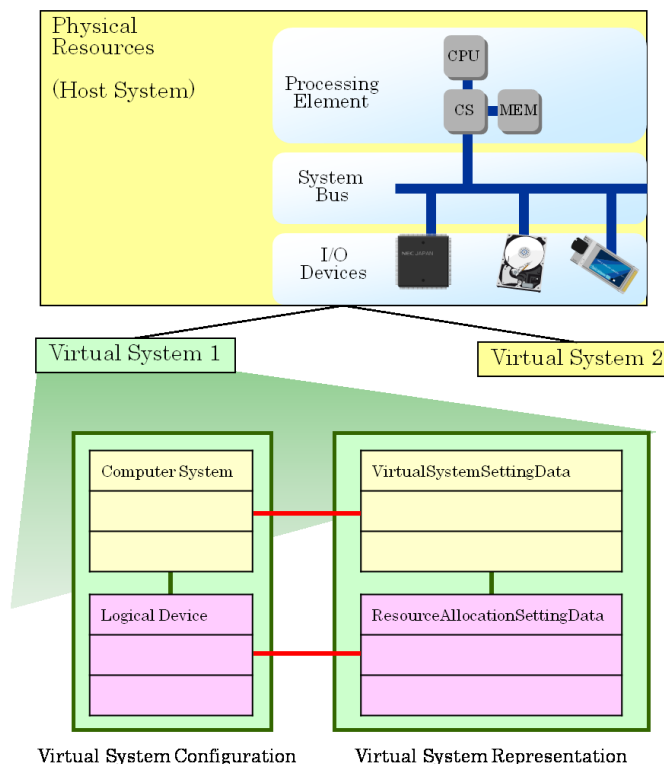


Fig. 9 Management Framework of a Virtualized System Using CIM

8. CONCERNS IN CARRIER-GRADE NE USE

This section explains management and performance issues related to the use of I/O virtualization in carrier-grade network equipment.

8.1 Management Issues

Management costs for a carrier grade system may be reduced by I/O virtualization. I/O consolidation is estimated to reduce management costs by 50%. Furthermore, development costs can be greatly reduced by substituting CPU-based, software-handled communication by I/O interfaces, as most of the redundancy can be implemented at the hardware level (by simply providing redundant I/O devices). This eliminates the need to develop redundant software that needs checkpoints, watchdogs, and so on, to ensure synchronous operation of active and stand-by systems.

However, after reviewing today's I/O virtualization technology, it is evident that there has been little consideration of reliability and availability. Most of the important management issues are left as vendor-specific implementation matters.

Examples:

- In the PCI-SIG's SR-IOV specification, the PCIM that must enable/disable the virtual function is not specified.
- According to the MR-IOV specification, before the OS is booted, various management operations have to be performed by the PCIM (such as, for instance, operations to create a virtual hierarchy by setting all the ports of the virtual switch, by establishing virtual links, and by creating a base function, a physical function, and a virtual function). The boot-up time seems to be significant.
- No consensus on who makes the PCIM.

Other unaddressed issues are hot plugging and swapping, congestion control, arbitration of multiple accesses for shared I/Os, and the treatment of VM migration. These issues arise when the I/O resources are treated as Field Replaceable Units (FRUs). As I/O resources are closely connected to the PE through direct memory access, one needs to take the following issues into consideration when I/O resources are replaced.

- Management of the I/O device status information such as liveness check by keep-alive or heart-beat.
- Virtualization layer (logical resource) and physical resource consistency in the resource management, especially when the I/O resource is plugged and dies. Because it is I/O, the plug-on/off must be done without causing a system error.
- I/O consolidation system. It might happen here that the CPU card is dead while the I/O hardware is healthy. In this case, it is convenient that the I/O hardware continues to work after the service is switched to another CPU card.
- Limitations in the I/O error reporting interfaces. It is not clear when an error is asserted and what to do with the error report at the receiver side.
- Advertisement of the device information to the resource/service manager for a certain service. What information should be advertized? Is it broadcast or unicast?

8.2 Performance Issues

- I/O itself should implement internal routing and arbitration for multiple I/O access, sometimes including congestion control or QoS. How to stack these functions is not specified in standards.

- Secure partitioning is a very important feature but not explicitly defined.
- ATS needs some algorithm/mechanism to trust the I/O to hand off the role of address translation.
- Most server systems only have NICs, HBAs, and graphics. On the other hand, for NE, there are a variety of I/Os and various accelerators like the IPsec engine card, TOE card, and other NW processing cards, as well as various interface cards like optical transport cards, T1 cards, ATM cards, GE cards, 10GE cards, and so on. Therefore, the interfaces for NE are also various. Some are standard but most are proprietary. Unification of those interfaces in a certain layer by using I/O virtualization technology is desirable. One promising way to do that is to unify the transport layer of those interfaces by Ethernet. This is promising because the Ethernet standard is increasingly prevalent in large areas like access networks as well as in small areas like datacenters. There have been some efforts to do this, such as FC over Ethernet or PCI-Express over Ethernet. [ref. 11, 14]. The Ethernet standard itself tries to specify features needed for backplane use. Examples are the link aggregation in 802.3ad [ref. 19], high-speed electrical interface in 802.3ap [ref. 9], and congestion control in 802.1Qau [ref. 10].
- The number of I/O cards that a single system needs is much larger than it used to be, and an I/O-intensive system has to deal with I/O interrupts quickly. Therefore, the real-time feature is important. Additionally, I/O accesses must be arbitrated neatly. Preemption priorities for I/O processes are needed to prevent a single I/O from monopolizing the entire I/O processing power. A VM path-through (direct I/O) is important, because it can shorten the I/O processing time by shortcutting the proxy process in the master VM.
- Failover and switchover must be completed quickly because data is sent from an outside world that does not “care” about what happens in the NE.

9. REQUIRED STANDARDIZATION EFFORTS

This section describes three items for further consideration in terms of standardization.

9.1 Availability Calculation

Availability is the greatest concern in CG systems; however, it is very hard to estimate for an I/O-virtualized system. This is because in these systems the services operate on logical resources, but the performance degradation is caused by failures in physical resources. A standardization body has to show how the system-error rate can be calculated depending on the redundancy and on the level of I/O virtualization. In the system design and deployment phase, it seems appropriate to use the MTBF (Mean Time Between Failure) of each I/O. If I/Os are distributed as shown in Fig. 7, their location, connection-topology, and commonly-used platform (such as chassis-mechanicals, power supplies, and chassis management module) should also be taken into account. For example, suppose that the service is operated on a VM on a SBC with multiple AMC-type network interface cards located in an AMC carrier card in either the same or in a different chassis like a u-TCA I/O consolidated box. The interconnections between the chassis are Ethernet links with 3-hop switches in their paths. In this scenario, there are multiple levels of failure. If an AMC has an error, it is a single error. But if the AMC carrier card fails, all the AMC cards on it will also fail. Therefore, if a redundant pair is made by using two cards with the same MTBF, the system-error rate will differ depending on whether or not both cards are a pair on the same AMC carrier. If

the redundant pair is allocated over more than one chassis, the simultaneous failure rate decreases, but the error rate at the chassis-to-chassis interconnection increases. A standardization body or the SCOPE ALLIANCE itself should provide a simple model for the calculation by referring to academic works [ref. 20]. Investigation of management frameworks extending a module-level management interface like IPMI over multiple chassis is also required. PICMG specifies the IPMI-proxy approach; however, because the approach is the concentrated management model, a more distributed way of taking I/O-virtualization in account should be included in the specification.

9.2 Interoperability

Because there is a vast variety of NE I/Os, their interoperability is a matter of great concern. It is very useful that a standardization body, and the SCOPE ALLIANCE itself, specify ways to test the interoperability of products made by NEPs and ecosystem vendors. One such test specification provided by a PCI-SIG workshop is a good example [ref. 21]. The test has two parts, one is a PCI-SIG test using a typical model, and the other is an interoperability test examining the compatibility of the products of different companies that participated in the workshop.

Another interoperability issue concerns the management/control frame. Each standard specifies such a frame, but these specifications are not sufficient for CG-NE. For example, PCI-SIG specifies an Advanced Error Report (AER), which an I/O-virtualized NE using PCI-Express can use for control/management. However, for CG-NE, a detailed definition of the AERs would be helpful; the main issues are: when is an AER sent, what is transmitted, and what should be done on the receiver-side?

After these matters have been discussed by NEPs and ecosystem vendors, the results of their discussion should be submitted to a standardization body as requirements for maintaining interoperability.

For a sliceable-NE like that shown in Fig. 8, the control interface should be standardized for interoperability. The activity of such sliceable resources is still at the research level. Industries should help specifying it by sending requirements for CG use.

9.3 Reliable Transport for I/Os over Multi-Chassis Transport

The loss of a packet causes serious problems in a system. Reliable end-to-end transport must therefore be assured. The TCP transport protocol is very famous for doing that, but because it uses the CPU and software for flow control, packet transport performance is often limited by the CPU performance. As performance is very important for network equipment, a low-level flow-control without CPU/software operation is desirable. To decrease packet loss at layer 2-level, IEEE 802.1 standardized congestion control is needed because most Ethernet packet losses are caused by congestion and by the consequent packet disposal at congested switches. Even if the congestion control works well, it does not guarantee lossless packet transmission. Therefore, end-to-end flow control should be implemented; moreover, it should be implemented in a low layer to provide high performance. In Ethernet transmission, for example, a flow should be defined in such a way that the Ethernet frame contains the flow ID, sequence number, and time stamp. The frame header must be standardized as a flow label; however, how it controls the flow should not be standardized, as that will depend on the system configuration.

10. CONCLUSION

I/O virtualization in terms of network equipment is first defined in this paper as an I/O issue to support VMMs. As network equipment performs more I/Os than servers, I/O virtualization is more important in the network equipment industry than in the server industry. In addition, I/O consolidation and resource slicing are treated here as another type of I/O virtualization.

Five I/O virtualization technologies are described: VTd, PCI SR-IOV, MR-IOV, I/O consolidation, and Virtual Slicing. These technologies are the keys to scaling system performance and to increasing availability by aggregating and sharing I/Os.

As to the I/O virtualization in NE, there are many implementation and management issues, especially concerning the treatment of I/O resources as FRUs, which is complicated because of the close connection and limited topology between I/O resources and processor element (CPU/memory). One possible approach to address this issue is to use DMTF profiling as a management object and to build systematic associations to map the FRU operation to SA Forum's control interfaces, such as hot swap.

Then, a variety of issues (both management and performance issues) that need to be considered when I/O virtualization is used in carrier-grade network equipment are explained. Today's standards do not specify solutions in detail and leave their implementation to vendors. Therefore, these issues must be discussed among NEPs if the merits of I/O virtualization, such as lower system cost, simpler system configuration, better performance, and greater scalability are to be achieved.

The description of all the technology and issues in this paper presents the landscape of I/O virtualization and is intended to raise awareness among NEPs (Network Equipment Providers) and ecosystem vendors on this technology.