On I/O virtualization management

Vitalian A. Danciu and Martin G. Metzker

Munich Network Management Team
Ludwig-Maximilians-Universität, Munich, Germany
$(danciu|metzker)@mnm - team.org$

Abstract. The quick adoption of virtualization technology in general and the advent of the Cloud business model entail new requirements on the structure and the configuration of back-end I/O systems. Several approaches to virtualization of I/O links are being introduced, which aim at implementing a more flexible I/O channel configuration without compromising performance. While previously the management of I/O devices could be limited to basic technical requirments (e.g. the establishment and termination of fixed-point links), the additional flexibility carries in its wake additional management requirements on the representation and control of I/O sub-systems.
This paper focuses on the modelling of dynamic and static aspects of the management of virtual I/O devices. Based on management scenarios and common operations on virtual machines we propose management function prototoypes and discuss the corresponding necessary information items.

1 Introduction

Consolidation of data centres and the introduction of massively virtualized system infrastructures have created new challenges to the construction and the management of I/O systems. The aggregation of data and thus I/O traffic and the need for flexible assignment of I/O capacity has given rise to the introduction of configurable I/O controllers and link schemes in order to replace the traditional, monolithic I/O setups. However, the flexibility introduced thereby necessarily introduces I/O facilities into the scope of IT management: while raw transmission capabilities were viable metrics before, management attention is shifting towards the configuration and setup issues arising from the need to apply the characteristics of these new devices to the emerging I/O usage scenarios.

Transition to virtualized sub-system elements The I/O sub-system of computer systems relies on a fixed assignment of components, especially host bus adaptors (HBA), in order to exchange data with back-end remote systems, typically storage facilities. The I/O performance available to a computing resource (CR), which is either a physical or a virtual machine (VM), is therefore directly dependent on the capabilities of the hardware components of the I/O sub-system. In virtualized installations, multiple VMs share the same HBA; typically, I/O is being identified as a performance bottleneck in such installations.

To address these issues, hardware components have been developed to include multiple controller instances and to enable hardware accelerated I/O virtualization. Figure 1 shows the structural changes to the I/O sub-system in the transition from the non-virtualized to the virtualized I/O setup. Virtual HBAs (vHBA) are provided directly by the hardware, and the burden of assignment is placed on hypervisor functions.

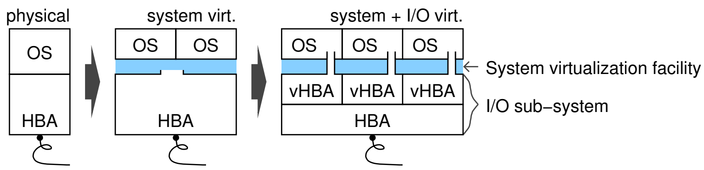


Fig. 1. Evolution from physical to virtual HBAs

Management problem The introduction of these features entails novel management tasks: while before, the I/O sub-system could be viewed as a black box from an IT management standpoint, the newly won flexibility creates the need to extend management operations into the configuration of virtualization capable I/O devices. Even without taking into account changing user/customer requirements, the I/O capacity available to a VM is dependent on the number and the state of other VMs being supported on a given physical platform instance. While in a non-virtualized setup, I/O capacity planning for a machine was performed at the time of purchase (by selecting the machine hardware according to needs), system virtualization combined with I/O-virtualization allow—and mandate—the projection of management goals onto the I/O sub-system at run-time. The management operations necessary to realise that projection are important, as are the attributes required to represent the I/O sub-system's state. The formulation of such operations and information items must take into the account the fact that I/O virtualization is being introduced in several technically different manners and that the higher-level (i.e. application-driven) management goals to be achieved differ in the requirements they place on the management of the I/O sub-system.

Structure of this paper In this paper, we derive management requirements regarding the configuration and operation of virtual I/O sub-system components from usage scenarios common in data centres. The requirements analysed in this paper and the therefrom derived model elements cover in principle both NICs and HBA. In the interest of clarity we will focus on HBAs.

We discuss management scenarios and analyse their requirements in Section 2 before reviewing different types of I/O virtualization and relevant standards in Section 3. According to these requirements we formulate management operations applicable to virtualizable I/O devices and identify the management information necessary in Section 4 to propose a sub-model aligned to the CIM. We conclude with a general discussion of this work in Section 5 and open questions in Section 6.

2 Management scenario

A data centre operator rents out parts of his infrastructure to customers, thus providing quasi-isolated, virtual infrastructures. Simply speaking, the operator creates a number of VMs and assigns them to customers in accordance to provisioning contracts and service level agreements. Additionally, the operator offers storage resources that are made available to customers' VMs.

Management of these infrastructures aims to satisfy customer requirements while efficiently exploiting the operator's resources. Thus, higher-level management goals dictated by customer requirements lead to the execution of operations at the VM level and entail, as a direct consequence, requirements on the management of the I/O sub-systems.

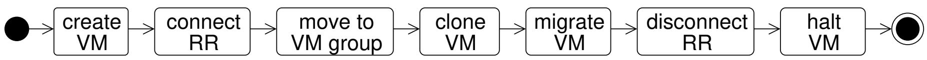
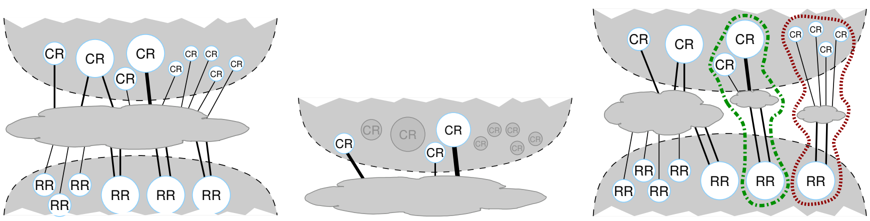


Fig. 2. Example life-cycle phases of VM

Common operations on VMs (i.e. computing resources) include their instantiation/creation, the connection of data channels and, eventually, their disconnection as well as the decommisioning of the computing resource itself. When a multitude of similar VMs are required, VMs can be cloned from an existing blueprint instance. In addition, VMs can be migrated between physical machines, either while running or in a suspended state. Figure 2 shows an example sequence of such operations, that imply action items at the I/O sub-system level.



(a) Connecting CR and RR (b) Partial shutdown (c) End-to-end isolation

Fig. 3. Data channels between computing resources (CR, i.e. VMs) and remote resources (RR, e.g. storage)

Computing units and storage space are connected by means of suitably dimensioned data channels in the operator's network (see Figure 3(a)). Thus, to provide service to a single group of computing resources, the provider must take into account the state of data channels (up, down, load etc), the endpoints of data channels and the service level constraints (gurantees regarding availability, thoughput, etc).

The provider reduces the number of running physical machines outside of peak usage periods in order to lower the energy expenditure for operation and cooling by consolidating VMs on fewer machines. However, the provider must ensure that their corresponding data channels are reconfigured to operate to the same specifications as demanded by service level agreements (cp. Figure 3(b)).

The consolidation of multiple customers' VMs on the same machine pools implies potential security threats from the point of view of customers that perform computations on sensitive data. In the same manner, shared data channels to storage resources present (or may be seen to present, by co-hosted competitors in business) a similar risk. To guarantee isolation the provider must provide separate data channels between CR and RR (in addition to providing separate CR and RR) as depicted in Figure 3(c). The isolation of data links traversing the network is achieved by using separate virtual LANs (VLANs) for channels pertaining to one customer.

Management requirements The provider faces several management challenges with respect to I/O management. It is notable that such requirements introduce the need to externalise part of the management functions to a manager component (or person).

placement and connection of CR and RR require management *information about the assignment* of VMs to I/O devices to be available, as well as *functions* to configure the I/O system.

group membership allow VMs and data channels to be included in unique logical groups, e.g. by customer, to enforce isolation requirements;

isolation requires that the assignment operations must take into account the *mapping of VM groups* to different customers, the *criticality and sensitivity* stated in customers' contracts in addition to quality of service requirements.

temporary re-location in order to lower the physical infrastructure's energy footprint requires functions for the *re-configuration* of CR–to–RR connections.

transit constraints forbid data (or VM migration) transit through a given network (e.g. over the internet); or mandate transit over a specific network;

capacity guarantees state a capacity (e.g. transmission rate) guaranteed to customer (while this requirement may be fulfilled automatically in a static environment, its fulfilment needs to be revised in the context of changes to VM deployment such as migration, cloning, re-configuration of channels etc);

redundancy guarantees state a measure of redundancy guaranteed to be provided, e.g. a customer may require double the number of effectively needed communication channels to allow for failover. Again, this last requirement breaks the encapsulation provided by I/O virtualization: if redundancy is to protect against faults in hardware, there is little point in providing two different virtual entities that are dependent on the same physical entity; a direct view into the mappings of virtual devices to physical ones is necessary to fulfil such a requirement by assigning redundant *physical* channels.

These requirements imply several information items that are relevant to the scenarios discussed, but not in the scope of the model elements describing the management of I/O devices: they are instead a source of constraints and goals (e.g. constraints/policies derived from customers' requirements, or formulated in SLAs/SLOs) with respect to that management. Methods for the detailed, formal description of the constraints themselves constitute interesting research items but fall outside the scope of this paper. We will refer to a set of constraints available to the provider, and we take them into account as a whole when treating functions and information items.

3 Background

The virtualization of the I/O sub-system borders, as a topic, on several different areas including data-centre-grade interconnects, the means of virtualization employed, and the resulting virtual structures in terms of networks, links and endpoints.

I/O virtualization techniques I/O virtualization is being realised either in the hardware platform (i.e. as CPU functions), in the system bus, within the HBA or as a specialised I/O Server, as sketched in Figure 4. An example for platform-based implementation is Intel's VT-d [2] which creates an abstraction from concrete data channels by allowing a re-mapping of DMA (direct memory access) transfers and of the interrupts generated by the I/O device. In contrast, an example for HBA-based I/O virtualization is found in the SR-IOV (Single-Root I/O Virtualisation) [9] extension of the PCI-Express [8] system bus.

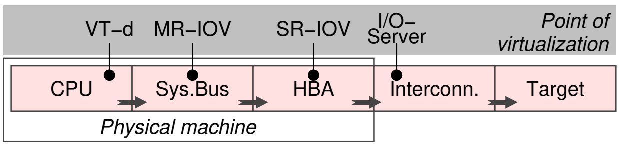


Fig. 4. Different points of virtualization on transmission path

An SR-IOV-capable HBA provides "virtual functions" that mimic multiple devices on the PCI bus that can be separately assigned to computing resources. MR-IOV (Multi-Root IOV) [10] virtualizes switch-based PCI Express topologies with multiple root entities, as found in blade installations. I/O servers externalise I/O devices into an entity outside the host machine and offer virtual HBA instances as a service. It is obvious that, while the goal of I/O virtualization remains the same, the means of realisation are quite different.

Management of I/O virtualization By using specialized components, virtualization can be implemented more efficiently. On the other hand it requires the orchestration of multiple functions of different entities to supply a (virtual) machine with an HBA, making it harder to describe a (virtual) machine's setup. Similar problems arise when creating virtual networks across large infrastructures, as described in [5], that describes a method for instantiating and deploying security and isolation mechanisms in virtualized environments. Rooney et al. describe a method for automatic VLAN creation, based on on-line measurement [11], while [3] describes a resource provisioning scheme regarding resource availability. These technologies can simplify network management when combined to automatically instantiate and control data channels in a virtualized environment.

Operations on VMs must also respect a number of management policies and constraints, including the available system resources at specific locations in the infrastructure, the adherence to isolation of customers' VMs etc. These items constitute management challenges in their own right, but they are outside the scope of this paper; an overview may be found in [6].

4 Modelling management aspects

The scenarios' use-cases and the general management requirements discussion above incorporate a number of distinct physical and logical entities (see Figure 5 and Table 1) which are the target of management operations; where available we have leveraged existing CIM classes to represent these entities.

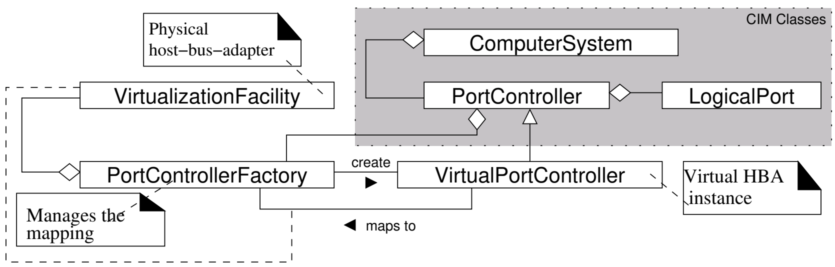


Fig. 5. Model view of the basic virtulization function

We extend the CIM PortController class by deriving a new class, VirtualPortController, to integrate a virtual I/O device into a virtual machine's model without having to modify other aspects of its model. The accompanying classes PortControllerFactory and VirtualizationFacility are used to model virtualization capabilities. VirtualizationFacility represents the device that provides virtual instances, while PortControllerFactory is the managing class which controls and monitors such instances.

4.1 I/O sub-system management functions

Figure 6 shows the sequence of basic functions for the management of virtual I/O channels as needed in the use-cases in Section 2: the creation of a VM, the initialization of a connection to a RR, as well as the subsequent disconnection of the resource and the decommissioning of the VM.

The task of creating a VM can be decomposed into single interactions. Note that a dominating aspect is the propagation of constraints. Distributing constraints top-down indicates that all key information lies with the Infrastructure-Manager, making this a very important (and busy) role throughout the VM's life-cycle. This is emphasised by

Class	*Description*
PortController	respresents a *physical HBA* with the ability to create a comm. end-point
VirtualPortController	represents a *virtual HBA*; its exposed functions correspond to, and are mapped to, those of a physical HBA
PortControllerFactory	is an entity that instatiates *VirtualPortController* instances on a *PortController* with the aid of a *VirtualizationFacility*. This entity encapsulates the basic management functions for creation of virtual I/O devices (vHBAs, virtual NICs etc). Its functions may be provided by hypervisor software.
VirtualizationFacility	represents the (physical) device that provides the virtualization function, i.e. one of the devices realising the virtualization points shown in Figure 4
Manager	an entity that enforces management policy by executing actions on the I/O devices (cp. Figure 6)
Data channel	a link between two communication end-points, managed by two (virtual) PortControllers, e.g. a VLAN
ConstraintSet	contains the conditions that I/O channel operations must satisfy (in this paper, we use group membership as an example constraint)

Table 1. Entity classes

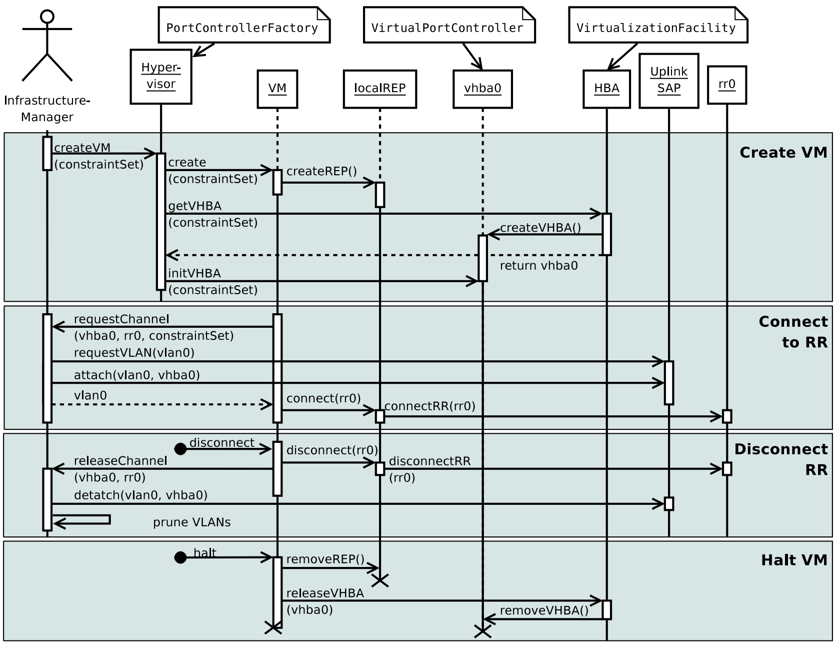


Fig. 6. Operations sequence during a VM's life-cycle

the many tasks performed by the Infrastructure-Manager when connecting to a RR. The main objective in this phase is the mapping to a virtual network.

We have summarised the necessary operations in Table 2 together with descriptions of their effect. The operation signatures are grouped by the entities they pertain to, i.e. the classes that implement the operations.

Application to the examplary life-cycle phases. We compose the operations on VMs (e.g. those shown in the exemplary life-cycle in Figure 2) based on the primitive function blocks from the sequence diagram in Figure 6; note that some of the life-cycle phases can be mapped directly.

create VM corresponds to the sequence shown in Figure 6.
connect RR corresponds to the sequence shown in Figure 6
move to VM group Assigning a VM to a group implies the specification of the policy of that group to be valid to the VM. The management actions required depend on whether the pre-assignment settings violate group policy. If that is the case for any connection to an RR, a disconnection and re-connection to that RR is necessary.
clone Cloning a VM implies creating an additional instance with a disjunct identity but with the same configuration settings. In our context, this means that the new clone must hold equal connections to the I/O sub-system and respect the same constraints, but that these connections may have to be duplicated. Thus, the clone phase implies a disconnection of the resources bound to the clone blueprint and a re-connection performed by the new clone, i.e. instantiating a VirtualPortController and requesting it be placed into the same interconnect group as its sibling.
migrate Migration implies that a VM is moved to another physical machine. There, it may or may not be offered the same I/O sub-system type and capabilities. To ensure the conservation of the CR–RR connection's parameters, bindings should be released before migration (i.e. disconnection sequences for all RRs) and reconstituted (connect sequences for all RRs) after migration is complete.
disconnect RR is shown in Figure 6
halt VM Destruction/deletion of a VM corresponds to decommisioning according to Figure 6, followed by the release of other resources than those of the I/O sub-system and, finally, halting and deleting the VM.

4.2 Attributes

Taking into account the different alternatives for virtualizing I/O adaptors in a generic model representation has required the introduction of new entities for which we propose partial models in the following and consider the management elements external to the virtualization function, e.g. management goals to be attained by means of I/O virtualization control and the manager actor (the latter is detailed in Section 5).

Typically, the VirtualizationFacility is a hardware-based device that allows the projection of multiple logical derived devices. To administrate these derived (i.e. virtual devices), it is important to provide a counter of instantiated devices as well as information

Infrastructure Manager	
createVM (constraintSet)	Called by the manager to create a new virtual machine. The supplied constraintSet contains information controlling the I/O channels the VM will use.
requestVLAN (VLAN)	Used by the manager to configure the infrastructure.
attach (VLAN, vHBA)	Allows vHBA to access VLAN.
detach (VLAN,vHBA)	Reconfigures the infrastructure to deny vHBA access to VLAN.
pruneVLANs (void)	Cleanup function to remove unused VLANs from the infrastructure.
Hypervisor	
create (constraintSet)	This is the internal function used by the hypervisor to instantiate a new virtual machine.
getVHBA (constraintSet)	A call to this function makes the VirtualizationFacility create a new vHBA which will be used to access the network and configures the associated physical HBA. The supplied constraintSet influences the decision which physical HBA is mapped to.
initVHBA (constraintSet)	Binds a virtual HBA to a virtual machine and initiates the vHBA's identifying properties.
Virtual Machine	
createREP (void)	Creates the virtual machine's local representation of an HBA. The resulting object is used to interface the virtual hardware, supplied by the hypervisor.
removeREP (void)	Disconnects the local HBA representation from the associated vHBA and removes the representation.
requestChannel (vHBA, RR, constraintSet)	Prompts the manager for the VLAN which will be used by the supplied vHBA instance to access the supplied RR, taking into account constraintSet.
releaseChannel (vHBA, RR)	Prompts the manager to undo all configuration changes related to the connection between vHBA and RR.
releaseVHBA (vHBA)	Instructs the VirtualizationFacility to remove the vHBA when it is no longer used.
connect (vHBA, VLAN, RR)	This function binds the VM's local representation of an HBA to vHBA and uses VLAN to connect to RR.
disconnect (RR)	Locally unmount from RR and disconnect.
OS-side port representation	
connectRR (RR)	Initiates the connection between a VM and a RR.
disconnectRR (RR)	Terminates link to RR.
HBA	
createVHBA (void)	Instantiates a new virtual HBA.
removeVHBA (void)	Delete's an instance of vHBA.

Table 2. Management operations, arranged by entity

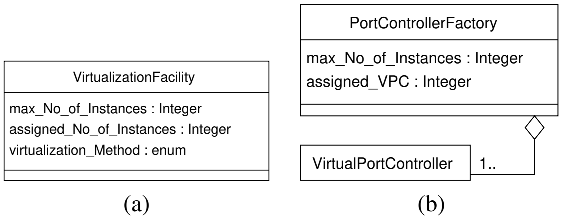


Fig. 7. Class attributes

regarding the virtualization method used and the maximum number of virtual devices supported by the VirtualizationFacility (Figure 7(a)).

The PortControllerFactory (PCF) is responsible for creating virtual PortController instances based on the abilities of the VirtualizationFacility. In simple cases, the PCF may be a part of the hypervisor code; in other cases it may be a management function of a product supporting I/O virtualization (e.g. a physical HBA). Its primary task is to create and administrate VirtualPortController instances and to bind them to LogicalPort instances. There are conceivable environments in which machines equipped with virtualization-capable I/O devices and machines without those capabilities are mixed. Thus, to provide a robust representation of the PCF, it must be assigned at least one PortController, whether it be a physical or a virtual instance. In case of a virtual instance, that first PortController will simply serve as the first (perhaps of many) instances created; in case of a physical instance, it will ensure an appropriate interface to (an equally physical) LogicalPort. In the more interesting case, i.e. when I/O virtualization capabilities are present, the PCF must keep a list of managed VirtualPortControllers as well as suitable counter variables (Figure 7(b)).

5 Discussion

Substantial challenges to the management of the I/O sub-system originate in the variety of the infrastructure partitions, the differences in their use, the variance of usage over time ("burstiness") and, finally, the service levels assured to the customers. These are understood challenges in network management. Several common-sense principles may be employed to simplify infrastructure design and management [4]. *Uniformity* regarding the view on devices and configurations requires the bridging of heterogenity. The introduction of *roles* helps ease the configuration of components by splitting configurations into small units that can be reused on multiple components, thus reducing the number of special cases. Our approach attempts to serve these two principles by mapping the operations on VMs onto operations on the I/O sub-system in a generic manner, agnostic of the concrete I/O virtualization technology and of the point of its introduction on the data tranfer path.

Another important principle is the organisation of components into a *tiered structure* (instead of in a flat one) to enable operators to assign tasks to groups of elements, thus reducing the number of tasks per elements. The organisation of VMs (and VM groups) and virtual and physical I/O devices lends itself to such organisation inherently. A principle more difficult to preserve is that of keeping *short dependency chains* to allow a reduction of the side effect of management operations. This is due to the conditions imposed on the management of virtual I/O devices by external entities (provider's policy, customers' requirements etc). We have localised the responsibility for creating the link between internal and external aspects in the Infrastructure-Manager role.

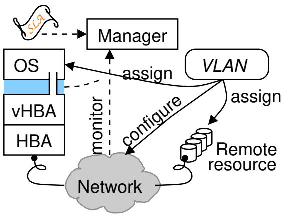


Fig. 8. Projecting management goals

The Infrastructure-Manager coordinates the steps when managing virtual infrastructures. While the effective configuration and orchestration can be automated, the Infrastructure-Manager has to supply the components with data on which resources to allocate and use. Often a human role is involved when grouping virtual machines and assigning resources. In particular, the Infrastructure-Manager role must have access to the policy set applicable to an infrastructure partition, for example a VM group, associated back-end storage, and the entities creating the data channel between the two. Thus, the manager's characteristic task in the context of virtualized I/O devices is to provide information *external* to that context. The application of that information is enacted by the *choice* of operations and attributes, instead of by relaying selected information to the acting entities (VirtualizationFacility, PortControllerFactory etc). Figure 8 shows the data flow to and from the manager entity and reflects the projection of (formalised) management goals on the execution of operations, by means of the example of VLAN assignment (from Figure 6).

Automation of the Infrastructure-Manager's tasks would imply their execution under a chosen control paradigm. Several approaches lend themselves to this task, among them policy-based and constraint based management. The formulation of rules with respect to I/O sub-system configuration appears to be a powerful means to manage data channels; however, the well-known issues of rule refinement and conflict handling must be considered.

6 Conclusion

The virtualization of I/O devices is a viable technology candidate to solve the I/O performance issues encountered in the heavily virtualized computing environments. We have analysed the management challenges posed by HBA virtualization scenarios and addressed the resulting requirements by identifying the management information necessary to perform the operations needed in the scenarios. We have covered a number of characteristic use-cases of HBA virtualization to identify the management functions and information items necessary at different points in the infrastructure.

This work ist but a step on the way towards consistent management of virtualized I/O devices: hardware products are still in development, and it remains to see which approach(es), as described in Section 3, will become dominant in the market. The multiple possible combinations of network protocols allow (virtual) I/O sub-systems different capabilities which in turn reflect on the management operations; an obvious example is the use of protocol stacks with QoS capabilities that allow effective management of data channel characteristics with respect to data rates, throughput and so on.

Acknowledgment

The authors wish to thank the members of the Munich Network Management Team (MNM Team) for helpful discussions and valuable comments on previous versions of this paper. The MNM Team directed by Prof. Dr. Dieter Kranzlmüller and Prof. Dr. Heinz-Gerd Hegering is a group of researchers at Ludwig-Maximilians-Universität München, Technische Universität München, the University of the Federal Armed Forces and the Leibniz Supercomputing Centre of the Bavarian Academy of Science. http://www.mnm-team.org.

References

1. Host LAN network port profile. Specification DSP1035, DMTF, April 2006.
2. Intel Virtualization Technology for Directed I/O (VT-d): Enhancing Intel platforms for efficient virtualization of I/O devices. August 2008. http://software.intel.com/en-us/articles/intel-virtualization-technology-for-directed-io-vt-d-enhancing-intel-platforms-for-efficient-virtualization-of-io-devices/.
3. Syed Zubair Ahmad, Muhammad Abdul Qadir, and Mohammad Saeed Akbar. A distributed resource management scheme for load-balanced QoS provisioning in heterogeneous mobile wireless networks. In *Q2SWinet '08: Proceedings of the 4th ACM symposium on QoS and security for wireless and mobile networks*, pages 63–70, New York, NY, USA, 2008. ACM.
4. Theophilus Benson, Aditya Akella, and Dave Maltz. A case for complexity models in network design and management. Technical Report 1643, University of Wisconsin, August 2008.
5. Serdar Cabuk, Chris I. Dalton, Hari Govind Ramasamy, and Matthias Schunter. Towards automated provisioning of secure virtualized networks. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 235–245, New York, NY, USA, 2007. ACM.
6. Vitalian A. Danciu. Host virtualization: a taxonomy of management challenges. In *SPIRIT 2009, Proceedings of the 2nd Workshop on Services, Platforms, Innovations and Research for new Infrastructures in Telecommunications*. Gesellschaft für Informatik, September 2009.
7. Martin Metzker. Design and analysis of techniques for virtualizing I/O-channels (in german). Master's thesis, Technical University Munich, April 2009.
8. PCI-SIG. *PCI Express Base Specification Revision 2.0*, December 2006.
9. PCI-SIG. *Single Root I/O Virtualization and Sharing Specification Revision 1.0*, Sept. 2007.
10. PCI-SIG. *Multi-Root I/O Virtualization and Sharing Specification Revision 1.0*, May 2008.
11. Sean Rooney, Christian Hörtnagl, and Jens Krause. Automatic VLAN creation based on on-line measurement. *SIGCOMM Comput. Commun. Rev.*, 29(3):50–57, 1999.
12. Storage Network Industry Association. *Storage Management Initiative Specification (SMI-S)*, April 2009. Version 1.4.0, Revision 4.