Impact of Virtualization on Management Systems

Sven Graupner, Ralf König, Vijay Machiraju, Jim Pruyne,
Akhil Sahai, Aad van Moorsel
Internet Systems and Storage Laboratory
HP Laboratories Palo Alto
HPL-2003-125
June 23rd , 2003*

E-mail: {firstname.lastname}@hp.com

virtualization, system management, Utility Data Center, OpenView

HP's Utility Data Center (UDC) [1] is a prominent example of how applications are provided with virtualized storage and network resources. Systems like the UDC are on their way into enterprise IT infrastructures and will have impact on management systems. In this paper we review some of the implications virtualized resource environments have on management systems and what changes are needed in management systems and what changes are needed in management systems to manage virtualized resources. In virtualized resource environments, applications are not installed on specific hardware instances. Instead, resources are provisioned dynamically to applications as needed. Resource instances may change frequently underneath applications, and vice versa, applications may migrate from one set of resources to another. Management systems must track changes based on information maintained in the virtualization layer. Currently, there is no integration between virtualization layers and systems managing virtualized resources. Furthermore, several assumptions break that are built into management systems today when management systems are applied in virtualized environments. We detail the broken assumptions and show principal solutions.

* Internal Accession Date Only Approved for External Publication
To be presented at the HP OpenView University Association, 6-9 July 2003, Geneva, Switzerland

Impact of Virtualization on Management Systems

Sven Graupner, Ralf König, Vijay Machiraju, Jim Pruyne, Akhil Sahai, Aad van Moorsel
Hewlett-Packard Laboratories, 1501 Page Mill Road, Palo Alto, CA 94304, USA

{firstname.lastname}@hp.com

Abstract

HP's Utility Data Center (UDC) [1] is a prominent example of how applications are provided with virtualized storage and network resources. Systems like the UDC are on their way into enterprise IT infrastructures and will have impact on management systems. In this paper we review some of the implications virtualized resource environments have on management systems and what changes are needed in management systems to manage virtualized resources.

In virtualized resource environments, applications are not installed on specific hardware instances. Instead, resources are provisioned dynamically to applications as needed. Resource instances may change frequently underneath applications, and vice versa, applications may migrate from one set of resources to another. Management systems must track changes based on information maintained in the virtualization layer.

Currently, there is no integration between virtualization layers and systems managing virtualized resources. Furthermore, several assumptions break that are built into management systems today when management systems are applied in virtualized environments. We detail the broken assumptions and show principal solutions.

1 Virtualization

Virtualization is a set of transformation processes in the virtualization layer during which associations between virtualized entities and 'underlyings' are established and changed. We use the term underlyings to denote any resources represented by the virtualization entities—these resources can be physical or logical, grouped or otherwise constructed. For clients 'above' the virtualization layer, a created virtualized entity exposes its own 'virtual' identity, properties and behavior, while attributes belonging with the underlyings remain hidden from the client.

The virtualization layer internally maintains the associations between virtualized entities and related underlyings, often also referred to as the "indirection" to underlyings. Knowledge of these associations allows "seeing" through the virtualization layer and being able to correlate virtualized entities with corresponding underlyings.

Examples of virtualized entities include:

- *Virtual memory* provide application processes with the impression of more RAM being available than actually built into in a machine.
- *Virtual disks* in RAID systems provide the impression of—depending on RAID level—faster and more reliable disks than physical disk devices.
- *Virtual machines* provide the impression that applications operate on individual machines while actually operating on the same one. The instances are sometimes called *virtual servers*.
- *Virtual Local Area Networks* (VLAN) allow to decouple sub-networks from the actual network topology by providing individual (IP) address spaces and DNS name spaces.
- *Virtual Private networks* (VPN) add privacy by encrypting payload or entire packets in a way that is transparent for applications.

The Utility Data Center [1] virtualizes storage in form of programmatically attachable disks to machines and networks in form of programmatically connecting selected machines by individual subnets. Virtualization points in the UDC internally are switches for Storage Area Networks (SAN) and Virtual Local Area Network (VLAN). The UDC externally exposes a control interface through which these control points are programmable in a higher-level specification language for resource environments.

Resource virtualization is used for several reasons:

- *insufficient quantity of resources*—when underlyings are scarce (e.g. processes as virtual processors, virtual memory, etc.),
- *sharing*—when underlyings need to be shared and using entities should not or cannot be aware of sharing and coordinate accordingly. The virtualization layer then coordinates sharing transparently for virtualized resources (e.g. separate address spaces in operating systems, virtual networks, virtual machines),
- *new properties*—when new properties and behavior of underlyings are desired (e.g. RAID creating disks with better performance and reliability characteristics),
- *transparent failover*—when underlyings fail, the virtualization layer can replace failed parts without exposing applications to the failures.

Similar reasons apply to services virtualizing applications to users and other using entities.

Virtualization in general may have several effects:

- *multiplication*—a higher quantity of an underlying can be created,
- *independence*—virtualized entities exist widely independently of one another, the virtualization layer takes care of necessary coordination when underlyings are shared,
- *protection*—virtualized entities are protected, no virtualized entity can reach into another virtualized entity,
- *isolation*—virtualized entities are isolated and not aware of one another,

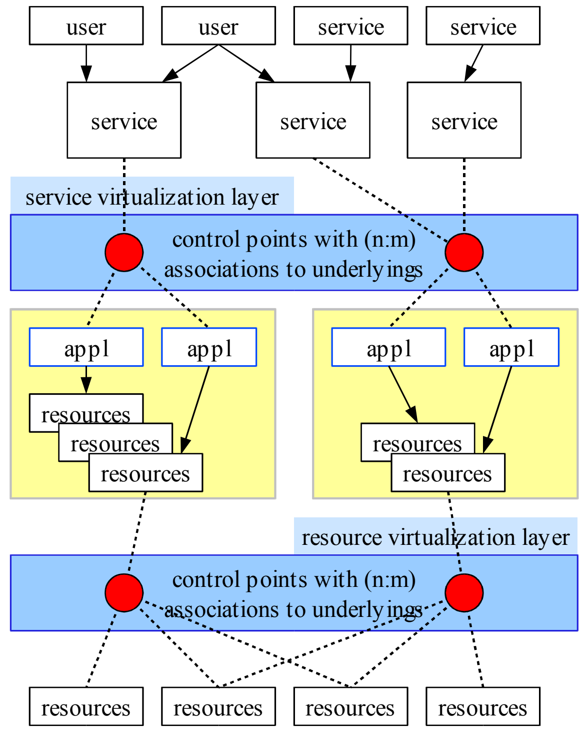

Figure 1: Virtualization at resource and service layer.

- *decoupling*—virtualized entities are decoupled from underlyings allowing the virtualization layer to change these associations without being noticed by underlyings or virtualized entities,
- *hiding*—underlyings and associations with underlyings are hidden in the virtualization layer and can hence be changed or moved without being noticed from outside. The hiding effect poses a barrier for management systems since it prevents correlating virtual with physical entities (see Section 4 on difficulties of management in virtualized environments),

Figure 1 shows the two most noticable layers of virtualization that can be observed today: resource virtualization and application virtualization in form of services. A general model would extend to an arbitrary number of virtualization layers.

2 Role of Management System in Virtualized Environments

System management is generally understood as the process of maintaining a system in an operational state, and improving and evolving that state towards an objective. One can subdivide operational management in three typical stages:

- *assess*—data gathering, processing, reporting, presenting, archiving
- *advice*—reasoning about the monitored data, drawing conclusions
- *act*—issuing control instructions back to managed elements.

A management system consists, on the monitoring side, of probes that monitor assigned managed elements and report to management servers that collect, process, present and archive monitored information. On the actuation side, a management system includes control interfaces of managed elements through which control instructions are issued to managed elements.

From the above definition of management, it becomes clear that virtualization overlaps in some of its tasks with a management system, since the virtualization layer must be aware of the resources available, and changes their state over time. As a consequence, one could imagine virtualization layers relying on management technologies to provide the necessary resource data.

In addition, the use of virtualization techniques implies the existence of some kind of 'management system' that translates customer requests into actions within the virtualization layer. Any actions that a management system wants to issue need to be coordinated with the management system that is associated with the virtualization layer.

The above reasoning suggests that management and virtualization are difficult to separate, and may better be dealt with in concert. We argue even stronger, that a virtualization layer should be looked at as 'just another' management system, one of many possible such systems, each with assess, advice and act capabilities. Together, these management systems form interacting control systems [1]. The remaining questions we address here are: what are the properties of these interacting control systems and how do legacy management systems fit in such future architectures. First, we discuss legacy management systems, then we address a generalized interface for virtualization layers.

3 Problems for Management in Virtual Environments by Example

Simple tasks like identification, naming and addressing of managed as well as management elements is a challenge in virtualized environments when no domain-wide, shared identification, name, and address spaces may exist since managed elements partially exist in virtualized layers with their own, private identification, naming and addressing conventions, and other parts exist in the physical layer underneath.

SNMP as the basis for many management operations is typically only supported in physical devices or resources. So far, only few virtual resources support SNMP management (see *Management interface assumption* in Section 4 for details).

Virtualization layers hide (vertical) associations of created, virtualized entities to underlyings causing difficulty for (horizontal) management systems seeing inside and seeing through virtualization layers.

The following two examples provide an impression of issues management systems are facing when moving into virtualized environments.

Example 1. Traditionally, physical server devices have been monitored under the assumption that a fixed and known set of applications is executing on a server. Monitored data thus could implicitly be associated with the applications running on that server. In contrast, under the regime of a Utility Data Center (UDC), the UDC controller may change assignments of applications to servers. Monitored data from the server devices thus can no longer implicitly be associated with applications since applications on a particular server may keep changing over time. Information from the UDC controller is needed in order to associate measurements from server devices with applications that have run on that server at certain times. If this information is not available, measurements at physical server devices have limited value.

Example 2. Another difficulty arises when management systems themselves are operated in virtualized resource environments such as in virtual networks accompanying a managed application. Monitoring and reporting is only possible within the virtual network domain of the application. Communication to data center-wide management systems across virtualized application network domains is difficult since virtual network boundaries must be crossed for management-related interactions.

4 Assumptions in Legacy Management Systems Challenged by Virtualization

Specifically the hiding effect of the virtualization layer causes problems in management systems since underlyings are not exposed to the management system, and it is hard to track associations with underlyings. These associations can even change over time. But there are more aspects that fundamentally change some of the assumptions that have been built into management systems:

Shared network assumption. Management system and managed system use the same, shared network infrastructure.

- Identification of managed elements and management elements (probes, HP OpenView spies, management servers, agents, etc.) is based on domain-wide unique (IP) addresses or hostnames in the underlying shared network.
- Managed elements can be reached by the management system from anywhere in the network using known, fixed addresses.
- Topology information is obtained (eventually discovered) and modeled in terms of the underlying shared, and rather constant network infrastructure.

In virtualized networks, sub-networks are independent from physical network topology. Since they may have own policies for identification, naming and addressing (based on own IP and DNS address spaces), global identification does not apply per se. Translations between identification, name and address spaces may be necessary.

Managed elements residing in separate virtualized sub-networks cannot easily be reached from other sub-networks. Their addresses eventually must be translated giving them different outer and inner identities, names, and addresses. Management systems must be able to follow translations of managed and management elements in different domains.

Fixed topology assumption. The assumption is that topology information is modeled (eventually discovered) in terms of physical entities in a shared network that will change infrequently.

In a virtualized environment, this assumption only holds for the underlying physical network. Virtual network topologies may occur, change, and disappear frequently, controlled by the virtualization layer and bypassing the management system. Automatically maintaining topology information with known techniques (discovery protocols) is also hard due to virtualized networks. Topology must be related with dynamic information from the network virtualization layer.

Physicalness assumption. The assumption is that managed elements physically exist in a network and can eventually be discovered.

Virtual entities are not physical entities. Their existence depends on the transformation process in the virtualization layer and is controlled there. Virtual entities are thus hard to be discovered automatically. They may not be responding to discovery protocols and may appear and disappear spontaneously.

Fixed association assumption. The assumption is that applications reside on machines for longer times, hence monitoring data obtained from that machine can implicitly be correlated with the application running on that machine.

In virtualized environments, associations between machines and applications may change frequently. Changes are under the control of the virtualization layer, not the management system.

Uniqueness assumption. The assumption is that one entity (resource, application, service, etc.) exists only once.

Since virtualization has a multiplication effect, entities may exist multiple times, even under same identity (for purposes of transparent replication, for instance).

Single-layer assumption. The assumption is that the management systems only views one layer of resources and applications, only seeing the elements that are visible in that layer.

In a virtualized environment, the virtualization layer introduces a clear and enforced separation between layers of underlyings and the layer of created (transformed) virtualized entities. These boundaries must be obeyed by management systems. We propose modeling layers as separate management domains (see Section 6) taking separate identification, name and address spaces in those domains into account and performing necessary translations for cross-domain interactions.

Highest authority assumption. The assumption is that the management system has highest authority (power of control) in the system. All control is exercised from the management system and its operators.

Virtualization layers have emerged independently from management systems as control points outside of management systems. These control points must be integrated back into management systems reinforcing their authority of control (see Section 7).

Full transparency assumption. The assumption is that the management system sees everything in the management domain and has control.

The virtualization layer internally hides associations with underlyings making them intransparent to the management system,

Furthermore, the virtualization layer can alter associations anytime without notifying any other components in the system including the management system. Shared infrastructure assumption. The management system, or part of it, uses resources from the same environment as the managed system for its own operation, resources that may have been virtualized as well.

Management interface assumption. The assumption is that manageable elements provide a management interface—such as a SNMP interface—that can be reached via a shared network infrastructure and is used to exchange monitoring data and control instructions.

Providers of virtualization layer software have just begun to equip their products with management interfaces which are easy to integrate into common systems management software. For example, VMWare has built read-only SNMPv1 support into *ESX Server*, while many of the control functions can be accessed via a Perl API [2]. For VLAN management there exist proprietary solutions (CiscoWorks and CiscoView, and more specific Cisco management software such as Cisco 12000 Manager; 3Com Transcend VLAN Management Software), with limited integration into management systems.

5 Virtualization and Management Systems

The fundamental problems of management systems applied in virtualized environments are:

- Management systems and virtualization layers are disconnected since they emerged independently of each other.
- Management systems are unaware of a notion of layers. They view flat landscapes of resources or applications and cannot distinguish between physical and virtual entities and associations between them.
- Applying management systems without awareness of the virtualization layer leads to cross-layer overlaps as shown in Figure 2. The management system views underlying and virtualized entities at the same level, unaware of the virtualization performed in the virtualization layer.
- Virtualization layers are controlled separately from the management system.

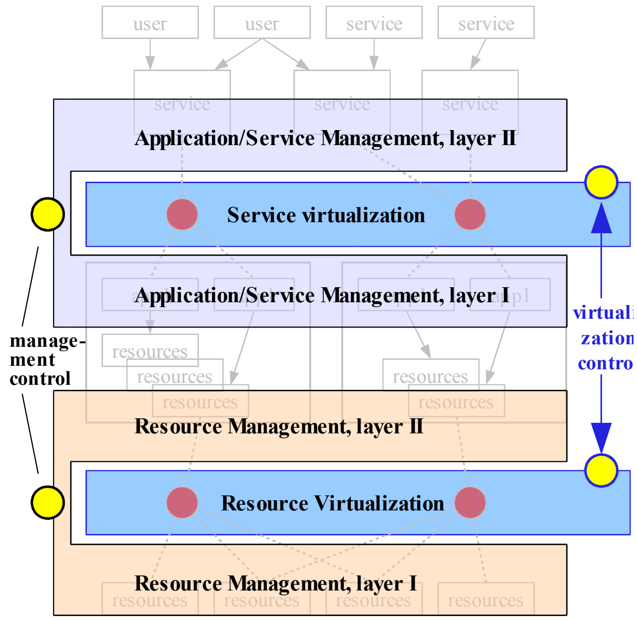

Figure 2: Cross-layer management domains spanning across two layers of underlyings (I) and virtualized (II) elements for resource and service virtualization (see Figure 1)

Figure 2 shows resource management in the lower part and application management in the upper part. Since both management domains deal with entities that are virtualized (above) and underlyings (underneath the corresponding virtualization layer), management systems are not aware of the associations between them.

They do not have access or control (or even knowledge) about these associations and the transformations performed in the virtualization layer. The management system is unaware of changes of assignments of virtualized entities to underlyings.

6 Conclusions for Management Systems

Based on the discussion, various conclusions for management systems and for virtualization layers can be drawn:

- Open the virtualization layer for management systems allowing them to access information about associations between virtualized and underlying entities.
- Integrate virtualization control into the management system reinstantiating authority of control of the management system.
- Model layers as management domains. Establish separate management domains for virtualized and underlying entities recognizing the layer boundary between them.
- Provide management interfaces for virtualized resources allowing them to be accessed for management purposes similar to their underlying counterparts.
- Develop resource models and service models that explicitly incorporate virtualizations.
- Introduce a notion of time when following associations from virtualized entities to underlyings since those associations vary over time.
- Reconsider identification, naming and addressing of managed as well as management elements based on network addresses. Translations across virtual network domains must be taken into account. An alternative is creating a separate system for element identification, naming and addressing that is independent of virtualization.
- Divide topology information into static aspects and dynamic aspects that depend on associations created by the virtualization layer and may change.

Figure 3 illustrates a view of an integrated management system where virtualization layers are integrated in management systems for resource management and service management. Domains separate underlyings from created, virtualized entities. The virtualization layer provides an interface for the management system through which associations between virtualized and underlying entities can be obtained. The details are shown in Figure 4).

7 Interface for a Generic Virtualization Layer

Due to the variety, individual virtualization layers or systems are not discussed here. Rather, a pattern for a

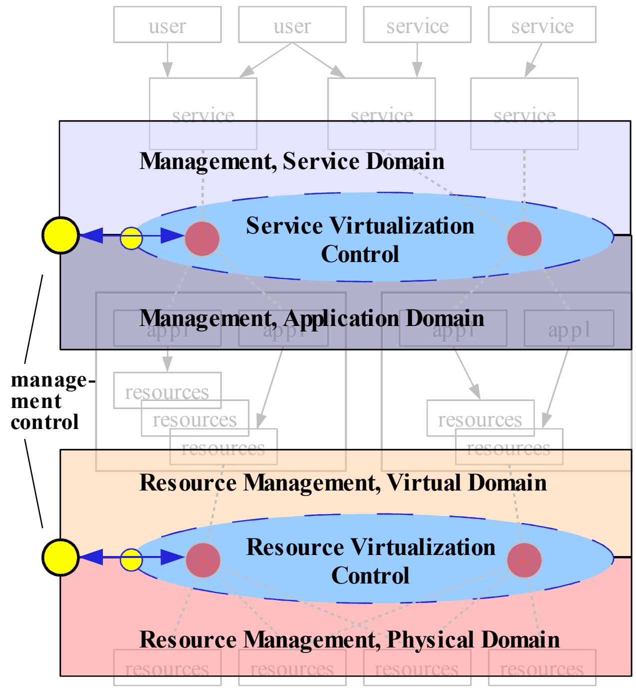

Figure 3: Integrated management system with separate domains for underlyings and virtualized entities.

generic interface between a management system and a generic virtualization layer is presented that can guide construction of resource management systems such as currently pursued in [3][4].

Figure 4 shows an abstract scenario with a set of underlyings $e \in \mathbf{E}$ in the lower layer and a set of virtualized entities $e^* \in \mathbf{E}^*$ in the upper layer. $\mathbf{E}^*$ is created by transformations in the virtualization layer at time t.

Associations $(n:m)$ between $\mathbf{E}$ and $\mathbf{E}^*$ are maintained and controlled by the virtualization layer:

$$assoc(\mathbf{E},\mathbf{E}^*) \subseteq \mathrm{P}(\mathbf{E}) \times \mathrm{P}(\mathbf{E}^*).$$

Entities in both layers are subject to management (managed objects) and are accompanied by management objects (mo for e, and mo^* for e^*).

mo^* are not virtualized mo. They are instrumentations that have to be brought into management domains separately. In both layers, entities are accompanied by separate management objects that are providing the interface to the management system.

The management system models the partitioning into layers as separate management domains **MD** and $\mathbf{MD}^*$ with management objects ($mo \in \mathbf{MD}, mo^* \in \mathbf{MD}^*$).

Since monitoring and control tasks are to be performed upon entities e and e^* through management objects mo and mo^*, management objects are connected with associated entities through a management interface. Examples are SNMP or other management protocols. Management instructions received from the management system are translated in management objects into corresponding interactions with associated entities.

An additional management interface must be provided by the virtualization layer to management objects mo^* in the virtualized domain $\mathbf{MD}^*$ since part of control of virtualized entities is provided by the virtualization layer. Management objects mo in **MD** may also have access to the management interface of the virtualization layer. This is not required when the management objects mo in **MD** should be kept unaware of virtualizations created above.

The management interface of the virtualization layer should be accessible via the encompassing management system.

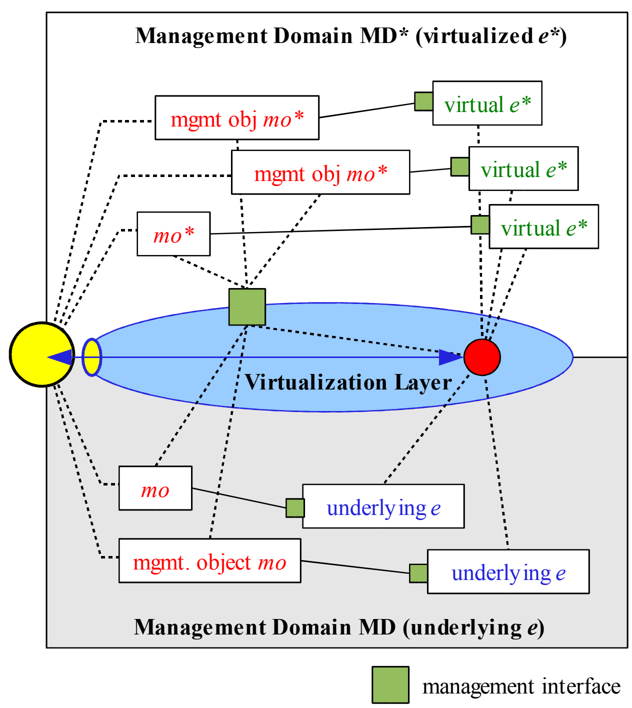

Figure 4: Management in an environment with virtualized and underlying entities.

7.1 Inner-Layer Management

Since layers established by virtualization are modeled as separate management domains, traditional management techniques apply with managed e and management objects mo within each layer.

The only specific property in $\mathbf{MD}^*$ is that management objects require control over their associated managed entities that is provided by the virtualization layer and hence has to be exercised through a management interface of the virtualization layer.

7.2 Cross-Layer Management

Cross-layer management is a more interesting case where management tasks have to be performed that span across layers of underlyings and created virtualized entities. Examples of such tasks (use cases) include:

1. replace an underlying (e.g. a machine), but take care of arrangements in the virtualized layer (e.g. affected applications running on that machine) before making that replacement;
2. identify which bindings to underlyings have to be resolved when a virtualized entity migrates;
3. how can measurements in underlyings be correlated with entities in the upper layer such as applications running on a server device at time t.

Since the virtualization layer maintains the associations between underlying and virtualized entities, cross-layer management requires tracking those relationships.

Since association may change, associations are depending on time.

7.3 Time-dependence of Associations

The first two cases only require knowledge about current associations. The third case may also include knowledge about associations of prior times, or even future times when associations have already been determined. Maintaining information about associations in the past would require that records about transitions altering associations would have to be kept in order to recall that information later for any given point in time.

7.4 Association Interface

The association interface plays an important role for cross-layer management. The association interface is attached to the virtualization layer and allows obtaining associations (including attributes of associations) between virtualized entities and underlyings. The interface can be equipped taking time-dependence of associations into account (assumed here).

The association interface consists of two functions f and f^{-1}. One function is resolving a given underlying entity e into a set of virtualized entities e^* to which associations exist at time t. The other function performs in the reverse direction resolving a given virtualized entity e^* into a set of underlying entities e to which associations exist at time t.

$$f(e,t) \rightarrow \{e^*\} \text{ at time } t,\ e \in \mathbf{E},\ e^* \in \mathbf{E}^*$$
$$f^{-1}(e^*,t) \rightarrow \{e\} \text{ at time } t,\ e^* \in \mathbf{E}^*,\ e \in \mathbf{E}.$$

When no time-dependence is supported, the current association is referred to.

Each management domain ($\mathbf{MD}$ and $\mathbf{MD}^*$) has additional functions to resolve (managed) entities e into associated management objects mo_e and vice versa:

In $\mathbf{MD}$:

$$g(e,t) \rightarrow \{mo\}, \text{ with } e \in \mathbf{E},\ mo \in \mathbf{MD}$$
$$g^{-1}(mo,t) \rightarrow \{e\}, \text{ with } e \in \mathbf{E},\ mo \in \mathbf{MD}.$$

In $\mathbf{MD}^*$:

$$h(e^*,t) \rightarrow \{mo^*\}, \text{ with } e^* \in \mathbf{E}^*,\ mo^* \in \mathbf{MD}^*$$
$$h^{-1}(mo^*,t) \rightarrow \{e^*\}, \text{ with } e^* \in \mathbf{E}^*,\ mo^* \in \mathbf{MD}^*.$$

With these primitives, association chains between virtualized and underlying entities can be tracked across management domains for cross-layer management purposes.

For example, in order to identify all management objects in $\mathbf{MD}^*$ that represent virtualized entities depending on a given underlying entity e, the following invocation chain resolves e into desired $\{mo^*\}$:

In virtualization layer: $f(e,t) \rightarrow \{e_e^*\}$,

in $\mathbf{MD}^*$: $\forall e_e^* \in \{e_e^*\} : h(e_e^*,t) \rightarrow \{mo_e^*\}$,

or combined: $h(f(e,t),t) \rightarrow \{mo_e^*\}$.

Determining management objects mo^* in the virtualized layer based on entities e from the underlying layer is useful for case 1 in Section 7.2 when management operations have to be performed in entities e^* in $\mathbf{MD}^*$ that depend on e in the underlying layer, and when the underlying entity e has to be changed or replaced affecting virtualized entities above.

This example shows how cross-layer management tasks can be performed referring to information about associations in the virtualization layer. The virtualization layer was extended by the proposed interface for this purpose. Similar techniques can be used for translating identities, names and addresses of entities and accompanying management objects between management domains.

Summary

In this paper we have reviewed the impact of virtualization on management systems and some of their fundamental assumptions that are challenged with virtualization moving forward. Reasons have been discussed why virtualization is on its way into data centers. It has been shown how management systems can be applied in virtualized environments, and what changes and additional functions and interfaces are needed.

To summarize, as virtualization has fundamentally changed concepts, operation and management of operating systems 30 years ago, virtualization moving into data centers will change operation and management of data center resources. Management systems must be prepared for those changes, which had given the reason for this paper to outline some of the implications virtualization has on management systems and how those can be addressed.

References

[1] Hewlett-Packard: Utility Data Center, `http://www.hp.com/go/virtualization`, 2003.

[2] VMWare: White Paper VMware ESX Server, Systems Management, `http://www.vmware.com/pdf/ESXSysMgt.pdf`, 2002.

[3] Sahai, A., Graupner, S., Machiraju, V., van Moorsel, A.: Specifying and Monitoring Guarantees in Commercial Grids through SLA, Proceedings of 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2003), pp. 292–299, Tokyo, Japan, May 12–15, 2003.

[4] Graupner, S., Kobiyama, J.: The Road to the Enterprise Grid: Enabling Technology for Managing Data Center Resources Using HP Utility Data Center, Industrial Track Presentation at CCGrid 2003, Tokyo, Japan, May 12–15, 2003.

[5] Machiraju, V., Rolia, J., van Moorsel, A., Quality of Business Driven Service Composition and Utility Computing, HP Labs Technical Report, HPL-2002-66, `http://www.hpl.hp.com/techreports/2002/HPL-2002-66.html`, March 2002.