



Visibility and Control are Inside Jobs

By Dan Kusnetzky, Principal Analyst

OVERVIEW

In earlier papers in this series, the Kusnetzky Group has suggested that organizations need to adopt a different viewpoint to make the best use of virtual machine software and the virtual systems that this technology can make possible. This paper will present the fact that it is better to work from inside a virtual solution than to try to surround it with a huge management framework.

QUICK REVIEW OF THIS SERIES

VIRTUAL MACHINE SOFTWARE ISN'T A PANACEA

The first paper in this series, which can be found here: http://www.kusnetzky.net/publications/ImpactPapers/20070911_VMS_is_not_a_panacea.pdf, pointed out that virtualization in the form of virtual machine software

has been presented by the media as a panacea that will resolve all of the problems an organization has with their IT infrastructure even though the picture is really more complex and virtual machine software is only one of a number of virtualization technologies that can be found in most industry standard datacenters.

This paper presented the Kusnetzky Group model of virtualization technology and laid out all of the other layers of technology that are needed.

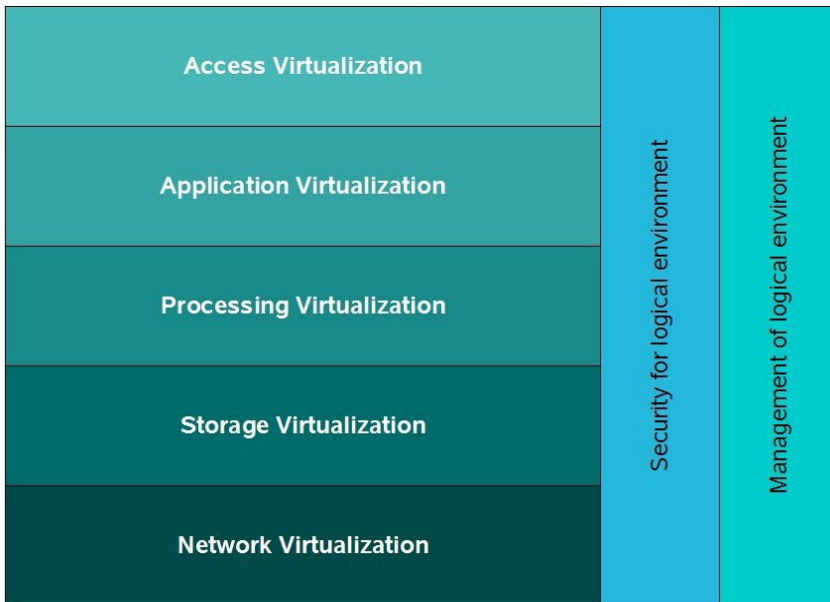
VIRTUAL MACHINES: YOU HAVE TO SEE THEM TO MANAGE THEM

The second paper in this series pointed out that virtualization can offer tremendous advantages over purely physical environments, but in the move to virtual, organizations often lost some things they took for granted — clarity of what is going on, where, and with whom.

That paper can be found here: http://www.kusnetzky.net/publications/ImpactPapers/20071023_VMS_you_have_to_see_em.pdf

VIRTUAL MACHINES: THEY MIGHT NOT HIDE, BUT CAN THEY RUN?

The third paper in this series, pointed out that the appearance of virtual systems and virtual resources in the industry standard part of the data center offered the hope of increased flexibility and reduced costs. It has also introduced challenges that were not part of the operational plans of many organizations.



Kusnetzky Group © 2007

Document #20071210

The Kusnetzky Group © 2007

The Kusnetzky Group is an independent supplier of marketing services to suppliers end user organizations and suppliers in the systems, virtualization and open source technology markets. Suppliers of hardware, software and virtualization technology are among those organizations. The opinions presented in this document are based upon our research, our personal experiences and actual use of technology regardless of whether this document or the supporting research were sponsored by one or more of the Kusnetzky Group's clients. This document may not be copied in whole or in part without the written permission of the Kusnetzky Group.

It also pointed out that virtual environments, like physical environments, need to function according to the policies set by the IT administrators.

This paper will explore that current management architectures aren't able to deliver the types of control and visibility that are really needed and that management visibility and control are best accomplished as an inside job rather than by trying to surround the environment with a large framework of some sort.

MONITORING THE VIRTUAL SYSTEMS

There are several techniques that allow administrators to "see" what's happening inside of a running system including:

- ☒ **Through the APIs.** Relying on the APIs provided by the developers of the operating systems, hypervisors and application framework software is inherently limited to the exposed functionality and information.
- ☒ **With an agent.** By leveraging agent technologies, loaded into the application space within each system, much better visibility can be achieved than the limited APIs typically provide. Agents must surround *all* of the running software's APIs with instrumented "shims" that feed data to the external administrative system.
- ☒ **With a low-level driver.** Different from an agent, a driver can collect information without existing obtrusively on the system, as a quieter, smaller-footprint mechanism within the system.

CONTROLLING VIRTUAL SYSTEMS

Similar choices appear when the goal is managing all of the control points of a running system once they are identified. Third-party APIs will typically expose only a subset of the types of controls needed to manage the system using external function calls. Internal approaches vary based on their access within the system. Certain controls can be affected through fairly high-level OS calls, while others require deep understanding and granular interaction with the operations of the system.

While the monitoring capabilities of a management tool may be mildly impacted by the architecture of that tool, control, and the flexibility of the control mechanisms are directly related to the implementation.

COMPARING INSIDE AND OUTSIDE APPROACHES

When evaluating alternatives to managing virtual infrastructures, then, there is a direct tradeoff associated with each choice. The criteria would include:

1. Does it provide the visibility needed?
2. Does it provide the level of control – both in terms of functionality and in terms of granularity – that is needed?
3. Is the architecture in line with the virtualization model?

API-based systems provided a limited amount of visibility and control. Clearly, the platform vendors themselves are typically able to deliver equal or greater visibility and control, as they themselves provided the APIs.

Agent-based approaches typically leverage existing systems management architectures. They do provide greater visibility than an API-based system, but often at the cost of running a bloated agent on the virtual infrastructure – directly counter to the philosophy of optimization and performance management. Their control, by virtue of their place in the stack, is less granular than one might want, as well. They can close a process or an application – but could they simply disable its network access, and leave it intact?

Driver-based approaches do seem to provide the best of all worlds. Often located within the kernel of the virtual machine, driver based systems provide the ability to see deeper into the operations and control functions of the virtual machine without imposing undue overhead. Unfortunately, embedded driver development for virtual machines typically requires a deeper level of expertise, a greater level of virtual machine internals knowledge, and of course, development skill. This means that the approach is often overlooked or merely dismissed as "being too difficult." It is clear, however, that this approach has the potential to make more granular insight and control functions possible.

VISIBILITY

Any development effort coming from outside of the system is likely to create an ongoing problem — updating the reporting software each and every time the vendors enhance or change the system. Furthermore, it would be nearly impossible to design custom API or agent technology that would *always* be able to see what's happening inside of a running system. There are times that everything is preempted while the system is executing critical functions.

It is pretty clear that getting information from inside of the system is better than trying to get third parties to support a custom API or to develop a comprehensive agent and reporting software. This approach also assures the IT administrator is able to see everything that is happening all of the time.

CONTROL

As with making internal operations visible, any development effort coming from outside of the system to control what's happening inside of the system is likely to create the same ongoing problem — updating the reporting software each and every time the vendors enhance or change the system.

It may also be impossible to control all of the functions of the running system from outside. Once again, there are times that everything is preempted while the system is executing critical functions.

It is pretty clear that being inside of a running system makes access to that system's control points much easier. This approach also assures the IT administrator is going to be able to manage functions all of the time.

S U M M A R Y

It is clear that the attempt to create a large management framework that surrounds virtual systems will have a number of insurmountable issues. The size and complexity of the task of creating either custom APIs or a comprehensive management framework is likely to make this approach much too costly. It also imposes an ongoing task of tracking the developments happening in each layer of the system and application software to make sure that the custom tools will continue to work as expected.

The Kusnetzky Group recommends that it would be better to find a supplier that can get inside of the system. An example of this is Fortisphere. They've been able to reach inside of the system in ways that would make the tasks of visibility and control straightforward.

More information can be found on the company's website: <http://www.fortisphere.com/visibilityandcontrol>.