HYPERIC | All Systems Go.

Hyperic Whitepaper: **Bridging the Virtual Divide**

HYPERIC HQ

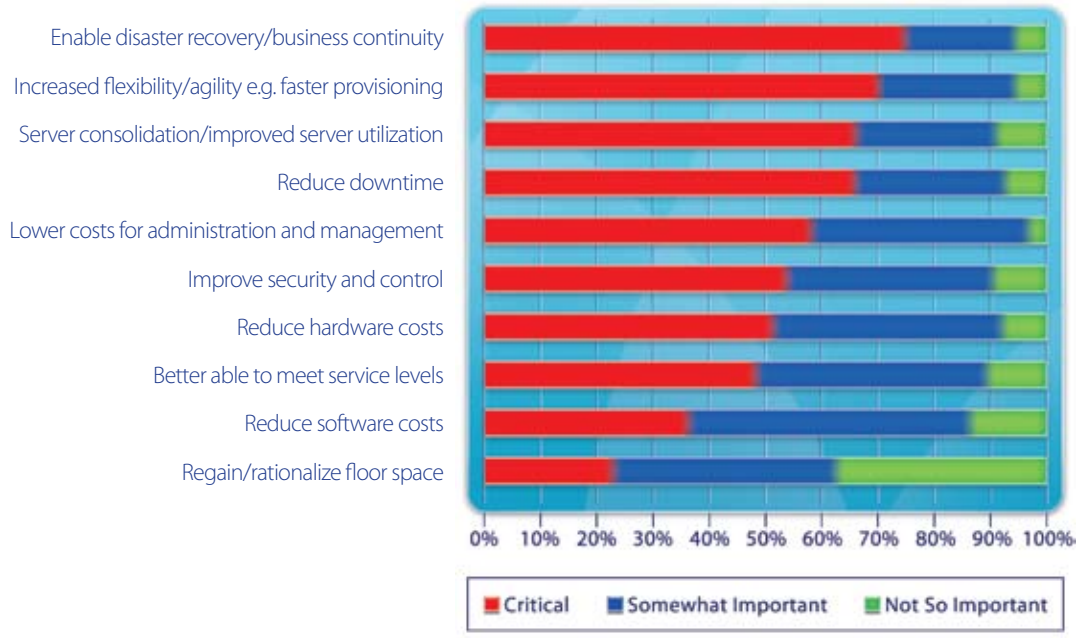
and

vmware

Understanding the Virtual Divide

Virtualization is one of the most paradigm changing trends in IT today. According to a research report by the Enterprise Management Association "Virtualization: Exposing the Intangible Enterprise" (July 2006), 75% of all enterprises have deployed virtualization in one form of another and 95% of them are planning to deploy virtualization. It's a safe bet that every data center has either evaluated or seriously considered the benefits of consolidating infrastructure around fewer physical servers. Enabling disaster recovery, increased agility, and better server utilization are some of the key drivers for companies considering whether or not to virtualize infrastructure.

Source: *Figure 24, "Virtualization: Exposing the Intangible Enterprise", a research report from the Enterprise Management Association (EMA), July 2006.*



Ironically, while virtualization infrastructure results in significant operational efficiencies, it also results in new infrastructure management challenges. Virtualization software carves out a logical, or virtual, machine from a physical one. This creates an artificially divided relationship between the applications inside the virtual machine and the physical resources they consume. This lack of visibility across the divide increases the propensity for configurations to change as administrators are forced to move virtual machines to improve performance rather then manage the performance of the Virtual Machines (VMs) directly in context with the applications running inside them.

As a result, an IT administrator can wake up to a "virtual sprawl" where an infrastructure that has shrunk from 100 physical boxes to 25, but grown to 250 new VMs —with a greater combination of operating systems, applications and changing, complex infrastructure that needs to be managed.

Not only does the system administrator have more to manage, he must also change his management approach. He must manage the physical performance of the host, typically with the virtualization software vendor's tools, and then separately, without the ability to automatically relate it to the virtualization software performance, or manage the operating systems and applications inside the hosts.

Management of IT resources across this divide requires a new type of systems management software -- software that bridges this new virtual divide.

The Quest for Virtualization Optimization

The partitioning of a single physical server through virtualization is not a new concept.

Virtualization, and its accompanying terminology, hails from the early mainframe time-sharing days. Through its pioneering virtualization approach, VMware has extended the benefits of virtualization to well beyond just mainframes to many other types of hardware platforms. VMware solutions separate the operating system and application software from the underlying hardware, delivering significant improvements in efficiency, availability, flexibility and manageability.

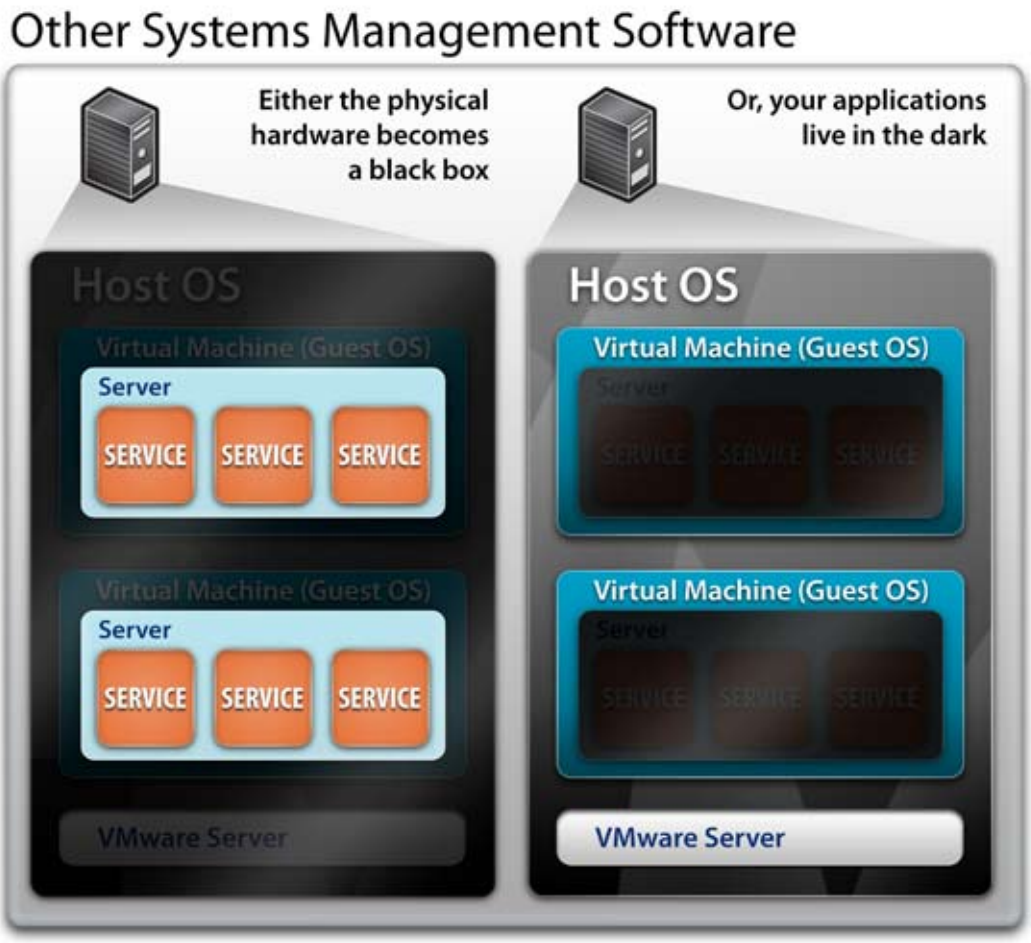
VMware solutions offer IT organizations a core benefit: squeeze as much performance and utilization out of physical servers while allowing virtual applications to be safely isolated from each other.

The isolation brings benefits, such as the ability to relocate complete VMs seamlessly within a virtualized environment (a key element in virtualization lifecycle management). However, in order to judge a virtualization implementation a success, there must be a measurable improvement in the utilization of physical resources and, at a minimum, no impact in the overall availability or performance of the entire environment. This is how the same aspects that make virtualization such an attractive proposition to businesses also make measuring its success very difficult.

In solutions offered by traditional systems management vendors, the task of monitoring virtualization performance and efficiency starts and stops with the VMware Server software itself. Operators typically judge success by measuring what the VMware Server says about how much of its physical resources are being consumed and distributed by the VM.

Traditional systems management solutions have two key limitations: First, since they only focus on VMware Server-level metrics they only show the impact of the VM on the physical hardware but don't show the overall performance of the hardware. Second, they ignore the overall performance and stability of the virtualized guest and applications.

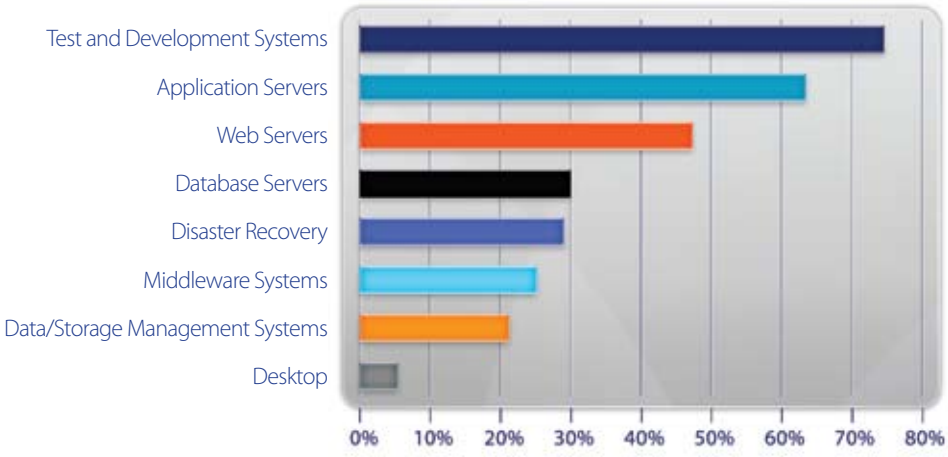
Consolidating 4 servers into 1 physical server with 4 VMs doesn't mean the applications are higher performing or more stable.



It's imperative to give equal (if not more) importance to the impact virtualization has on application performance as the impact it has on server utilization.

Why has virtualization become so pervasive without adequate solutions from traditional systems management vendors? Until recently, adoption of virtualization offerings was limited to QA and development departments, with over 74% of enterprises reporting that they are deploying virtualization for testing and development according to Enterprise Management Associates. Virtualization dramatically increases the speed of the testing lifecycle through the build, test, fix and rebuild stages because it offers a fluid, highly utilized environment to optimize testing permutations. Since testing environments typically do not have the same management strategies as production software, management has been limited.

Virtualization works. It increases the utilization and fluidity of applications used in production that are distributed, and are created as reusable services. Popular, proven examples of this include core services such as file and print, DNS, LDAP and other services that aren't powered by complex software stacks, yet provide critical services which need to be easily replicated. The only exception is "intermittent use" applications which have a heavy and intermittent use of only one of the resources such as CPU, memory or bandwidth, are optimized.



Source: *Figure 23, "Virtualization: Exposing the Intangible Enterprise", a research report from the Enterprise Management Association (EMA), July 2006.*

The adoption of virtualization infrastructure is rapidly growing beyond the traditional domains of QA and development departments. Virtualization is beginning to encompass more complex, business applications as well as nearly every layer in the infrastructure stack, as shown by the chart below from the EMA Survey. This process has already begun with custom applications built on web technologies. The rise of SOA and composite applications has created a new distributed architecture where services are discrete, in many cases stateless, and a perfect fit for a virtualized environment.

Virtualization software has proven to be incredibly effective at marshalling CPU resources across virtual machines. This effectiveness, however, can result in an application that is starved for critical resources that affect its underlying performance.

With the success of virtualization, applications are so easy to propagate in an environment and as a result, the problem could repeat itself anywhere from a dozen to a few hundred times in a given environment.

Hypothetical Case Study: Virtualized Webapp

Imagine building a new CRM application using an Apache web server on the front end of a Tomcat server all running on a Linux platform with a MySQL database collecting all the data. This application is currently running on an off-the-shelf, dual CPU machine that occupies one slot on a rack deep in some sprawling data center.

The scenario is a familiar one in today's fast-moving environment. It ended up there because a project that started off as an experiment or a rapid prototype became a customer facing, business critical application.

While this application is heavily used, it doesn't consume the full set of resources the physical host provides it. This excess of resources goes unnoticed for a year is not noticed. The next year, an assessment of the amount of space and physical resources in the data center reveals that this same thing has happened 20, maybe 50 times.

Applications get developed, deployed, and racked by the various teams that support them creating a maze of 1U boxes running singleton applications. Time for a new plan: contain the sprawl through virtualization.

The team notices that the CRM application fits the criteria for virtualization, and therefore it is slated for the new virtualized application stack. The team leading this effort creates a nice little VM for the application very quickly and efficiently and it passes basic QA testing. The VM then gets deployed on a machine running virtualization software such as VMware ESX.

In typical virtualization fashion, it gets deployed alongside 4 other VMs each running different applications with different technologies, including a J2EE app, a .NET app and mail server—each of which has its own appetite for resources.

The VMware's internal monitoring tools give the operators the impression that there's adequate capacity for all the VMs that are deployed on it. Until one day, a usage change occurs. Perhaps its quarter end and there's a huge sale, or maybe a new product release has a serious bug and the call center is suddenly flooded.

This little CRM application happens to be a critical function of the company's ERP system. End users are busy creating opportunities, managing accounts, and doing their usual activities, but at a much more rigorous pace. The heat is on and suddenly the application is struggling to keep up. Inevitably, the question becomes: "Is virtualization the cause?"

Unfortunately, this question isn't easy to answer. On the lower, system-based level, it's difficult to decipher how the VM monitoring tools choose to allocate resources to this now very busy VM. The original assessment of whether the CRM app fit within this particular physical host was based on a completely different load profile.

On a higher, application-centric level, the team lacks visibility into the performance of the different pieces of the supporting stack to determine if the application itself is the problem. This hinders their ability to anticipate performance issues before users start complaining, regardless of whether the app was running in a VM or on a dedicated host. The only solution to this problem is to dynamically re-provision the VM to another, less taxed virtualized host machine using VMotion.
While this alleviates the problem for the moment, the scenario can turn into the IT equivalent of a game of "whack-a-mole"--where lack of end-to-end visibility, the dynamic

nature of web application load, and the sheer number of moving parts in an environment force applications to be moved from one side of virtualized infrastructure to another without solving the underlying problems. Virtualization makes the challenges of problem management more complex than before, and that's why a robust and sophisticated systems management capability is critical in virtualized environments.

This scenario undermines the purpose of virtualization in the data center.

Introducing Hyperic HQ for VMware to Virtualization

Hyperic HQ for VMware is a pioneering new solution that enables data centers to consolidate complete discovery, monitoring, analysis and control of all application, system and network assets, both inside and outside of the virtual machines powered by VMware.

The Hyperic software is purpose-built for the web infrastructure powering the adoption of production virtualized environments. Hyperic starts with an inventory model that represents a server, all its applications and components within an intuitive set of relationships that apply to any combination of physical or virtual hardware, software, and services—regardless of the technologies involved.

Hyperic HQ for VMware provides a set of management features essential to the operation of modern infrastructure:

- **Inventory Auto-Discovery –** automatically populate every aspect of the physical and VMware virtual machines, applications, and services running in a given environment as well as the relationships between virtualized components and the physical resources they depend on.
- **Real-time and Historical Monitoring –** collect and store any availability, performance, and utilization metric for any resource in the inventory repository.
- **Event Management -** track logs and configuration changes from any resource in inventory.
- **Control –** take corrective or pre-emptive action on any application, service, or script in inventory.
- **Analyze –** combine as much information from logs, performance metrics, and infrastructure events into a set of correlated views that help you diagnose problems and determine capacity.

These core management functions are essential to the proper operation of any production IT environment, virtualized or not. The challenges of virtualization put added strain on legacy solutions designed for similar purposes.

The CRM application example above illustrates how virtualization allows for rapid provisioning, deployment, and relocation of complete application environments. Hyperic HQ's inventory auto discovery capability can detect every layer of a virtualized environment and react accordingly when any of those resources are relocated.

In that scenario, Hyperic HQ would have automatically discovered the VMware ESX Server host along with the Linux guest with the Apache server and the MySQL installation and its corresponding tables and indexes. These resources would have been mapped with "parent-child relationships" automatically to their counterparts on the physical VMware ESX Server. This mapping would provide operations with a complete view of the CRM application, as well as how it relates to the physical host on which it lives. In the event that the VM for our CRM application is relocated, Hyperic HQ would be immediately aware of the change and simply update its inventory to reflect the new parent of the VM, leaving its internal components like the apache server intact.

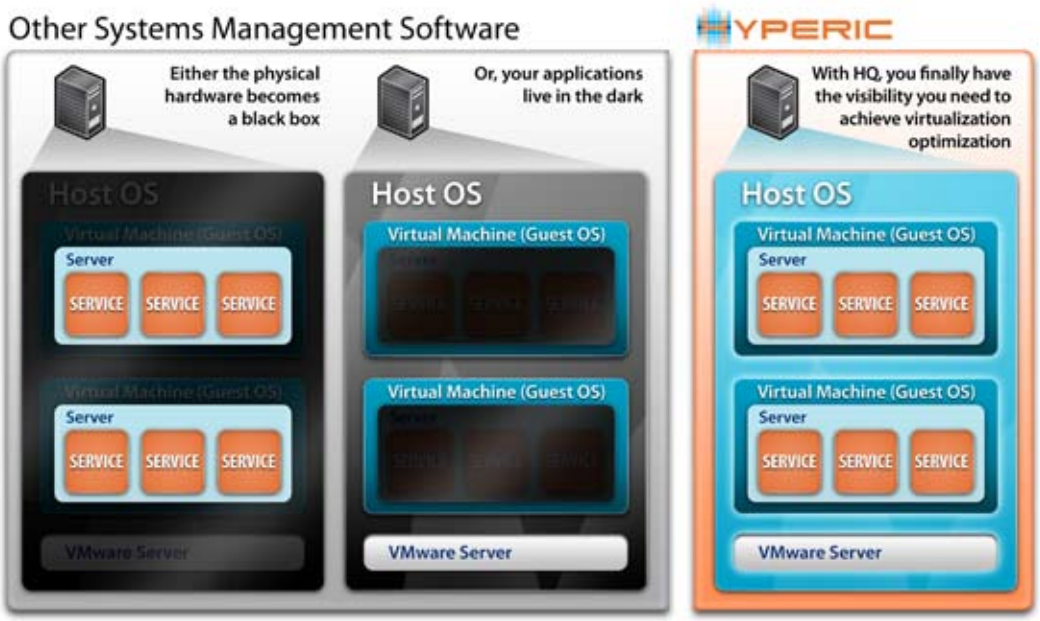
Once the virtualized environment is properly mapped, the next challenge is to provide monitoring visibility into every layer. Hyperic HQ enables this by providing out-of-the-box support for collecting real-time and historical values of all performance and health statistics at every layer of the infrastructure. In the CRM example, Hyperic HQ would provide monitoring for the virtualization software itself. (For a complete list of the available metrics and supported products, see the Appendix.)

In order to properly assess the performance of our CRM application it's important to see all of this data in concert with each other. Before, it was known that the CPU utilization had maxed out, and it was moved to another location with more CPU without any information or analysis on what caused the change. It can only be assumed that the increased transaction load increased the appetite for CPU resources. The only fix suppressed the problem, but it didn't uncover or address the root cause, and certainly didn't optimize the virtualized environment.

By using Hyperic HQ for VMware, an administrator could see the trend of the CPU utilization on the CRM VM moved towards the maximum threshold. It's clear that the Tomcat server consumed all the free memory on the OS and stopped responding. In this case, the administrator could correlate the memory usage from the OS with the memory usage from the Tomcat server and the Tomcat webapp services. It's evident that there is one particular webapp service that corresponds to the number of requests served by the webapp. This classic example of a memory leak was always there, and was exacerbated by the increased usage.

Armed with this information, action can be taken. First, the webapp developers can be alerted to a probable memory leak in their application, and are directed to a specific service to fix. Next, instead of moving the VM to a larger location, the Tomcat server can be restarted to clear out the memory.

This is a faster course of action, and will satisfy the users while the development team estimates the fix. Additionally, alerts can be set to beep the administrator's pager in case the Tomcat's memory usage gets too high again. This solution is a good one because the long term and unexpected outages can be limited and the VM can likely continue in its current location once the fix has been applied.



Continued on next page - **Virtualization Optimization Success**

Virtualization Optimization Success

The team has a complete picture of the entire VMware virtualized stack, and can rationalize performance and availability of applications with the overall goal of optimizing the physical infrastructure. The team can stay ahead of the chaos by implementing a management system that reacts seamlessly to the dynamic provisioning activities common to virtualized environments.

Hyperic HQ for VMware provides a quicker path to virtualization optimization, and a better chance at closing that physical to virtual divide in IT management once and for all.

Using Hyperic HQ for VMware, your organization can reap the following benefits:

- Install, set up and start managing your VMware infrastructure with one click
- Auto-discovery capability gives you a continuously accurate inventory of all the hardware, software, and services inside and outside your VMware infrastructure
- Organize the Operations Dashboard, metrics, charts and alerts to view your infrastructure the way you do
- Understand the real performance of your virtualized environment, from the both the host and the guest perspectives
- Compare performance data with configuration changes, log data and security events for fast problem ID and resolution
- Define intelligent alerts, acknowledgements and response actions to help anticipate problems before they cause outages
- Execute control operations to stabilize environments directly through the HQ user interface
- Accurately model and display the relationships between your hardware, software and services
- Analyze real-time and historical metrics from your production hardware, network, virtual machine and application layers

Appendix

The following table outlines a sampling the monitoring capabilities HQ provides at each layer in our CRM application example:

Layer	Supported Types	Metrics
Physical Host	x86 x86_64 Sparc	Load average CPU Physical Memory Disk I/O Disk Utilization/Capacity Network I/O
Guest Operating Systems	Linux Windows XP/2003 Server/Vista Free/Open BSD Sun Solaris	Load Average CPU Network I/O Disk I/O Memory activity Disk Capacity
Guest Applications	Web Servers MySQL Database Mail Servers J2EE Application Servers .NET Messaging Servers Tomcat	Availability Request activity Response time Memory consumption CPU Consumption Throughput Capacity
Virtualization Software	VMware GSX VMware Server VMWare ESX	GSX 3.x Server 1.x ESX 2.x ESX 3.x
Virtualization Software	General Server Metrics	Availability Memory Available for VMs Memory Used by VMs CPU Used CPU Used per Minute CPU Idle CPU Idle per Minute Number of CPUs
Virtualization Software	VMware ESX 2.x and 3.x VM NIC Metrics	Availability Packets Transmitted Packets Transmitted per Minute Packets Received Packets Received per Minute Bytes Transmitted Bytes Transmitted per Minute Bytes Received Bytes Received per Minute

Layer	Supported Types	Metrics
Virtualization Software	VMware ESX 2.x and 3.x VM Disk Metrics	Availability Reads Reads per Minute Writes Writes per Minute Bytes Read Bytes Read per Minute Bytes Written Bytes Written per Minute
Virtualization Software	VMware ESX 2.x and 3.x VM Metrics	Availability Process Virtual Memory Size Process Resident Memory Size Process Page Faults Process Page Faults per Minute Process CPU System Time Process CPU System Time per Minute Process CPU User Time Process CPU User Time per Minute Process Uptime Process CPU Total Time Process CPU Total Time per Minute Process CPU Usage VM CPU Wait VM CPU Wait per Minute VM CPU Used VM CPU Used per Minute VM CPU Sys VM Memory Shares VM Memory Minimum VM Memory Maximum VM Memory Size VM Memory Ctl VM Memory Swapped VM Memory Shared VM Memory Active VM Memory Overhead VM Uptime

Layer	Supported Types	Metrics
Virtualization Software	VMware GSX 3.x and Server 1.x VM Metrics	Availability Process Virtual Memory Size Process Resident Memory Size Process Page Faults Process Page Faults per Minute Process CPU System Time Process CPU System Time per Minute Process CPU User Time Process CPU User Time per Minute Process Uptime Process CPU Total Time Process CPU Total Time per Minute Process CPU Usage
Virtualization Software	Control Actions	Start Stop Reset Suspend Resume createSnapshot revertSnapshot deleteSnapshot saveScreenshot