Realtime publishers

New Best Practices in Virtual and Cloud Management: Performance, Capacity, Compliance, and Workload Automation

Greg Shields

sponsored by vmware

Introduction to Realtime Publishers

by Don Jones, Series Editor

For several years now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We've made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book's production expenses for the benefit of our readers.

Although we've always offered our publications to you for free, don't think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you $40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the "realtime" aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We're an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I'm proud that we've produced so many quality books over the past years.

I want to extend an invitation to visit us at http://nexus.realtimepublishers.com, especially if you've received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you're sure to find something that's of interest to you—and it won't cost you a thing. We hope you'll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Copyright Statement

Chapter 1: New Best Practices in Performance Management

Virtualization complicates, even as it optimizes.

A statement worth remembering. Also a statement too often ignored by many IT practitioners. In the giddy rush to implement virtualization for its benefits, many forget its hidden challenges. These challenges exist across the landscape of daily activities and are often so intertwined with the benefits of virtualization that the costs are easily missed. Performance management, capacity management, compliance management, and workload automation—listen too carefully to the pundits and vendors, and you'll quickly believe virtualization by itself brings huge operational improvements to each of these activities.

Impressively, it does, but not necessarily *all by itself*. Virtualization offers the potential to improve these activities. It's the next step that most people forget: Translating that potential into actual improvement. Reaching that goal sometimes requires a little extra help.

The mission of this guide is to assist you in translating potential into concrete improvement. With virtualization having become so popular—some analysts suggest that more than 50 percent of all IT workloads are now virtual—it's time for another look at the best practices in virtual and cloud management. With an eye towards virtualization's original value propositions, this guide intends to illuminate the industry's new best practices, deconstructed into four fundamental activities: performance management, capacity management, compliance management, and workload automation.

Most importantly, this guide will help you recognize where virtualization's complexities go beyond the limits of human attention. To deliver on its promise of optimization, virtualization *has to complicate* a few things. It's managing the balance between complication and optimization that's become the newest task.

PerfMon Might Be a Joke…

In my role as an IT industry author and presenter, I get the opportunity to stand up in front of a lot of people. In the past 10 years, I've presented to countless thousands of IT professionals. The experience of these individuals spans organizations from enterprise to small and midsize business (SMB) and everything in-between. These speaking opportunities present me with a lot of time to fill, so I often use that chance to poll audiences everywhere on the big questions I find personally enthralling.

How do people manage system performance is one of those questions. I'm routinely amazed by the response. In those presentations, I often ask, "How many people here have turned on PerfMon on your Windows servers?" The answer, every time: *No one*. Or, on rare occasion, some statistically-insignificant number of individuals that's very close to zero.

The audience response surprises me every time, so much so that the question's become a regular joke in such presentations. Its punch line: "So, when someone calls in and says, 'Hey, the mail server is slow today!' What do you tell them?"

Invariably, someone in the back quips, "Have you tried rebooting?"

…But It's Indicative of a Larger Problem

My story is intended to be humorous, but it's also intended to highlight a key failure that's endemic to many of our data centers: Without some measure of baseline monitoring, how can you tell what's different between today's behaviors and those from last week, last month, or last year? Simply put, you can't.

And yet out of the hundreds of audiences and thousands of people who've laughed at the punch line, still *10 years later* almost nobody gets the joke. And they wonder why their virtual environments aren't performing to expectations.

I'll admit, PerfMon isn't a great tool for across-the-data center performance management nor is it even really the right tool. That said, I see its limited use as indicative of a much larger problem: *We as an industry aren't practicing performance management.*

That problem is perhaps a result of our industry's hardware successes. Our hardware improves so fast that we've defined a law, Moore's Law, which is still relevant 50 years past its creation. The hardware improvements that are observed by the law are undoubtedly great for our workloads, but they come with consequences that become greatly apparent as those workloads get virtualized. See if you agree with this assertion: Today's class of IT professionals has "grown up" during a period where computing resources were virtually *limitless*.

Take a look at the performance statistics on just about any desktop or server and you'll surely agree. Figure 1.1 shows a representative screenshot from a sample machine's Windows Task Manager. That machine's eight processor cores and 16GB of memory are all but unused in the processing of its workload.

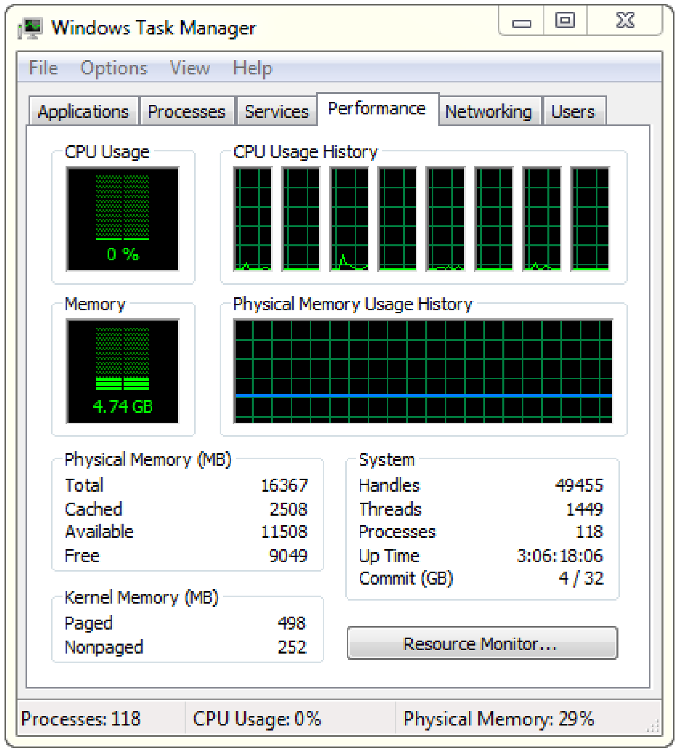

Figure 1.1: Windows Task Manager.

In fact, some analysts suggest that average processor utilization across all IT workloads, *across all industries*, lies somewhere between 5 and 10 percent. This news shouldn't be earthshattering. IT's embrace of virtualization is a direct result of the desire to consolidate these low-consumption workloads. By co-locating many workloads atop a smaller number of physical hosts, virtualization aims to eliminate exactly these inefficiencies in resource consumption.

But remember, virtualization *complicates*, even as it optimizes. It is immediately after the initial consolidation that too many data centers stop, subconsciously ignoring the downstream effects that are a direct result of virtualization's goal. Through virtualization, a data center seeks to squeeze useful work out of every resource unit. Greater optimization means greater resource sharing, right on up to the point where resource demands are perfectly balanced with those in supply. And then, all too often, *right on past* that point.

Performance Management Requires Visibility

Effective performance management first requires paying attention to the behaviors going on inside a system. A running workload requires processor attention, memory for execution space, some quantity of storage, and a bit of network connectivity for communicating with clients and other servers.

Metrics associated with the consumption of these resources can be measured with tools such as PerfMon. From a position inside a computer instance, these tools convert the behaviors they see into numbers. Those numbers can then be compared with known thresholds to identify when a workload is attempting to do too much, or its hardware resources are in too short a supply. Figure 1.2 shows the perspective PerfMon (and others like it) have when they're watching workloads on a physical machine.

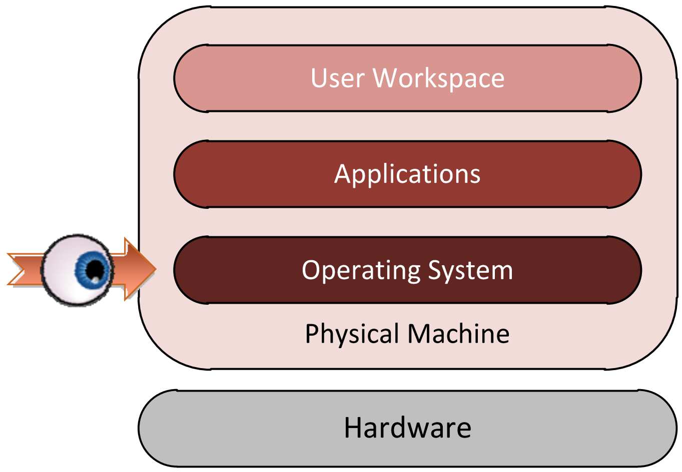

Figure 1.2: PerfMon's view into measuring system performance.

Locating the eyeball where it sits in Figure 1.2 works because of the relatively simple nature of physical machine resources. Processor, memory, storage, and networking resources in this situation are dedicated for use by a single workload. Those the workload doesn't use sit idle.

The apparent simplicity here is perhaps another reason more IT professionals ignore performance management. In an all-physical data center, where each successive hardware generation is more powerful than the last, performance management is sometimes considered an unnecessary activity. Resources are generally in such great supply that a shortage is rarely a problem.[1] Now add a hypervisor, and suddenly the picture grows far more complex.

Figure 1.3 documents the limitations of the eyeball placement in Figure 1.2. It shows a graphical representation of what a system looks like after it's been virtualized and co-located with another. Determining now where resources are being consumed is a bit more difficult. Processor cycles not consumed by the first virtual machine might be used by the second, or they might be completely unused. The same holds true for memory, storage, and networking resources.

[1] With the obvious nod to those situations when they are, but that's not my point.

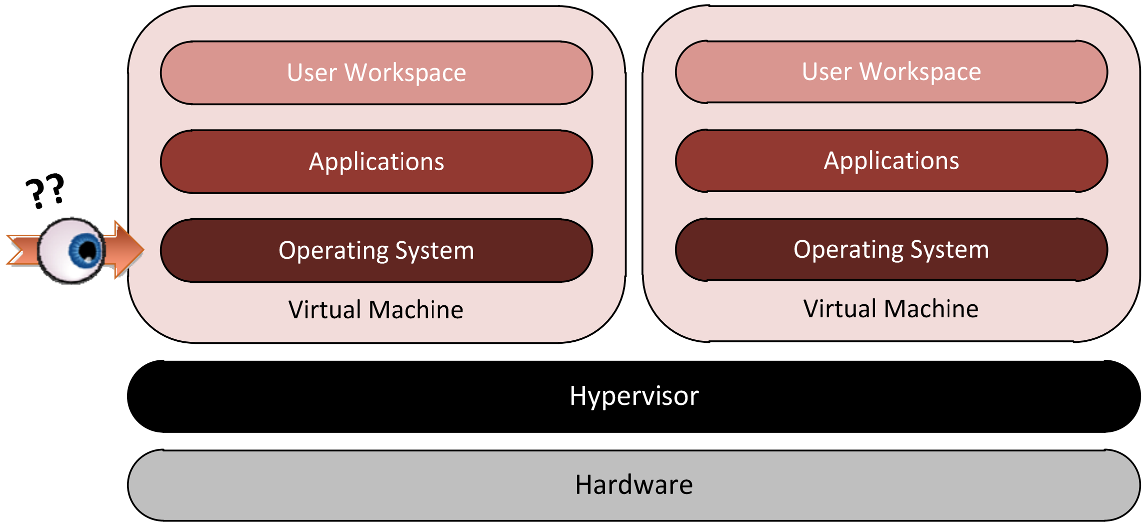

Figure 1.3: The single-system view is insufficient in a virtual environment.

"A-ha!" says the well-meaning virtual administrator, "PerfMon (and its ilk) has no value in virtual environments! It can't measure hypervisor activities! That's why I use my hypervisor management platform to monitor performance metrics across every virtual machine at once! By monitoring from the hypervisor's perspective, I can measure behaviors across every virtual machine."

Indeed, you can, and in fact, the vast majority of data centers employ a hypervisor management solution (most typically from their hypervisor's platform vendor) for managing configurations and monitoring performance metrics. What they get from such solutions tends to look like Figure 1.4.

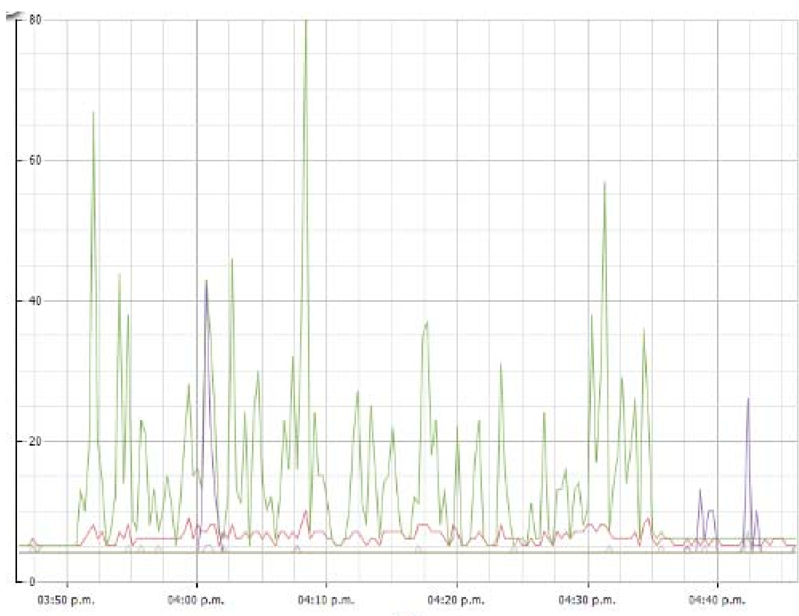

Figure 1.4: A hypervisor platform's "PerfMon."

Do you recognize this graph, at least in concept? You've probably seen something similar in your hypervisor management solution. What information can you glean from its half-dozen overlapping lines? What behaviors does it illuminate? Or, more specifically, *exactly what action* should an administrator take to resolve whatever behavior is being illustrated here?

Not entirely easy to answer these questions, is it? These are just another bunch of colorful lines. In fact, the data being communicated in Figure 1.4 looks eerily similar to another ~~joke~~ tool discussed earlier in this chapter. What do you do when someone calls in and says, "Hey, the virtual host is slow today!" *What do you tell them?*

Performance Management Requires Situational Awareness

It should be obvious at this point that the casual monitoring of raw metric data very quickly grows futile as an environment's interdependencies increase. At this point, we know that measuring performance from an individual virtual machine's perspective illuminates only a part of the story.

We are also beginning to recognize that moving the focus onto the hypervisor merely presents *a second* perspective (see Figure 1.5). The hypervisor isn't all-knowing or all-seeing. With the eyeball pointing at a virtual machine's operating system (OS) and its hypervisor, that view still misses a few key components of the overall virtual platform.

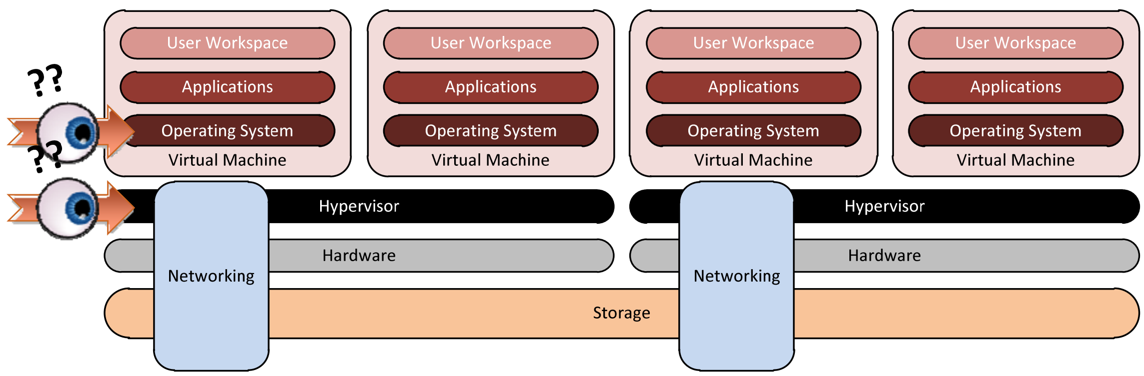

Figure 1.5: Merely a second perspective.

Storage and outside-the-hypervisor networking are two components being missed, as are the behaviors going on among hypervisors. Mission-critical virtualization requires high availability and load balancing. It requires the elimination of single points of failure. It demands redundancy at every level to ensure component failure doesn't mean system failure. Each of these interconnections on their own can be a contributor to performance problems, and each requires independent management and monitoring.

Each interconnection also introduces yet another perspective on the resources that contribute to virtual machine demands. Storage, networking, hardware, the interconnecting fabric—consider how Figure 1.5's "other" layers can impact each other:

- The storage might experience a resource shortfall that contributes to some upstream problem "felt" by the users of a virtual machine
- The hypervisor might balloon out memory that's being actively used by a needy virtual machine process
- The backplane of the switch being used for networking might become oversubscribed by storage traffic, reducing throughput for production networking

It stands to reason, then, that improving situational awareness for performance management tools requires also paying attention to behaviors at the other layers.

That's a lot of eyeballs, but the concept isn't new. As Figure 1.6 suggests, virtually every data center-class piece of hardware exposes collectable metrics. So do hypervisors and hypervisor toolsets installed into each virtual machine. Storage reports metrics, as do network components. Heck, even the servers themselves expose environmental and other at-the-chassis data that can be merged into a more situationally-aware view.

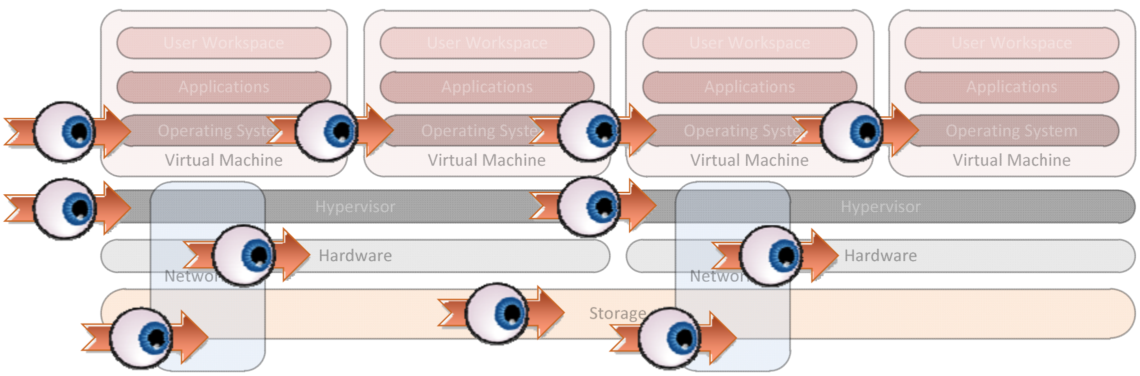

Figure 1.6: Monitors, everywhere.

Now capture these metrics with some unifying solution. Give that solution the task of collecting behaviors from every perspective, then crunching the numbers. Align the metrics by time, and suddenly that solution sees everything, everywhere, at once. A problem in the network that causes a problem in the storage which then causes a downstream problem with a database server's performance can be better correlated. As you can see, this *eyeballs everywhere* approach is better able to identify how behaviors at one level impact operations at another.

No Human Alive

Let's swing back to reality, for just a moment. There is a lot of data, and there remains a danger in simply collecting more metrics. The human challenge of having to correlate ever more metrics is what got IT in its PerfMon-phobic situation today. With that veritable fire hose of measurements coming in at every moment, there comes a point where no human alive can divine meaning out of the numbers. But where humans fail, *algorithms succeed*.

Here's the part we IT pros often forget: Computer systems, even the highly-interdependent ones driving our virtual environments, are by nature deterministic. Every behavior that can be measured exists for a reason. That reason is by definition predictable, if one has the necessary algorithms in place. Or, to put it in different terms, what we humans perceive as chaotic is in fact just a system with a lot of variables.

We can't divine meaning, but mathematics can. By replacing constant human attention with a kind of algorithmic *black box*, we gain a helping hand in processing those metrics and deconstructing the fire hose into something more manageable. What goes in Figure 1.7's black box are all the metrics from your virtual environment's layers; what comes out is a kind of actionable intelligence, or essentially "suggested actions based on actual data."

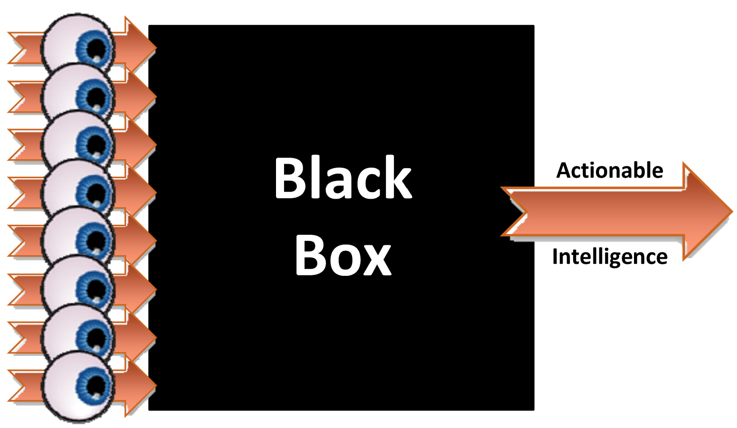

Figure 1.7: An algorithmic black box.

You cannot easily build this yourself. You can find proof of this in the multiple solutions on the market today that follow the black box approach, each with its own spin on the concept. Notwithstanding their differences, nearly all of them subscribe to the notion that monitoring a virtual platform has to happen *at a perspective outside that virtual platform*.

What's Inside the Box?

Although the exact detail behind each solution involves some measure of secret sauce, the overarching concept is what's important. In order to understand what's inside the black box, you must first understand the nature of the metrics themselves.

> **Note**
>
> I wrote another book titled *The Definitive Guide to Application Performance Management* (Realtime Publishers) that discusses in greater detail the framework for this black box monitoring approach. At around 200 pages in length, it provides a deeper-level discussion on this topic.

The real world can involve hundreds (if not thousands) of metrics across a wide range of potential components. To help illustrate, let's simplify things and play pretend for a minute. In our imaginary data center, we'll ignore all the actual metrics and their names. We'll throw away our preconceived notions of IOPS and CPU Latency, Memory Pressure and Network Packets Received, et al. In its place we'll focus on an imaginary metric: *Jeejaws*. It's a silly name, but that's my point. It's important here to separate the metrics from what the metrics intend to do.

> **Note**
>
> This separation of metrics from *the intent of metrics* becomes even more important in Chapter 2 as we extend performance management into capacity management.

In this imaginary world, we can assume that components reporting higher numbers of Jeejaws are simply doing more. A virtual machine, for example, that's reporting 450 Jeejaws is performing more work than one with half that number. The same holds true for the other components.

We've hooked up monitors in this world to all the appreciable places that could impact performance. These correspond to the virtual environment components discussed throughout this chapter. Figure 1.8 shows the monitoring solution in place, where metrics are being collected from the virtual environment's most important locations.

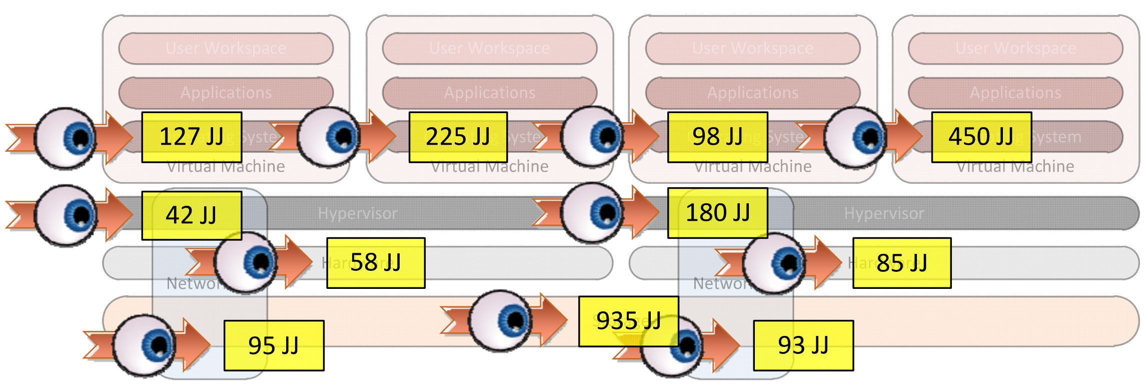

Figure 1.8: Measuring Jeejaws.

Then, one day, we get a phone call: "The Exchange server is slow!" Now what? Suddenly, the performance troubleshooting process becomes quite a bit more scientific.

> Well, virtual machine #4 over there is running at 450 Jeejaws today, but hypervisor #2 is showing double the Jeejaws of hypervisor #1. And, check out that storage metric! When was the last time it ran above 600 Jeejaws?

Replacing the prototypical administrator's gut feeling approach is a quantitative measurement that can relate actual performance to numerical values. Quantifying performance then involves identifying what's acceptable, and recognizing when a component's activities have gone past thresholds.

There's one piece remaining: dependencies. As you can image, every component in a system has dependencies on other components. Remote storage, for example, won't work well atop a poorly performing network. The final step in this process involves creating a kind of service model, a hierarchy that defines the traceability between dependencies.

Figure 1.9 shows a simplistic example of what a service model might look like. In it, you can see how the Exchange application's performance relies on an OS. That OS relies on resources from the Virtual Machine object. From there, behaviors within the Hypervisor and Hardware can create an impact, as can Storage and Networking.

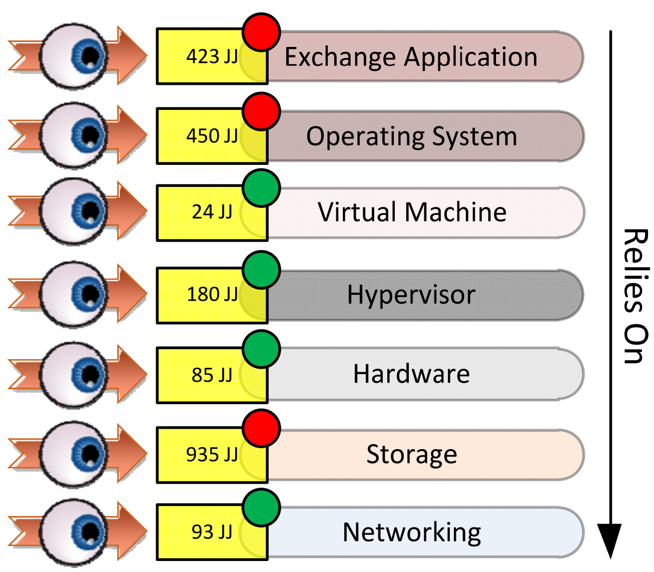

Figure 1.9: Applying thresholds to metrics.

Now, admittedly, a production virtual environment's model would look far more complex. A fully-defined model could involve an impressive branching path of dependencies with virtual machine activities impacting each other, all atop a mesh of hardware interdependencies. This example is intended merely to get you started.

With that model in place, it becomes much easier to see how the activities of one component impact another. Figure 1.9 also displays a stoplight icon on each component. With that visible notification, a troubleshooting administrator can quickly identify that a problem is occurring, then trace that problem to its contributing behaviors. That's the actionable intelligence the black box intends to deliver: *Here's where to look to fix this problem.*

Step back just a bit from the fantasy, and you can begin to add actual counters that might be valuable in a virtual environment. Knowing the number of storage utilization Jeejaws might not be enough to identify a performance problem's root cause, but the metrics that feed into that value might. Figure 1.10 shows how this could occur for the four sub-counters that reflect more detail about what's going on with the storage components.

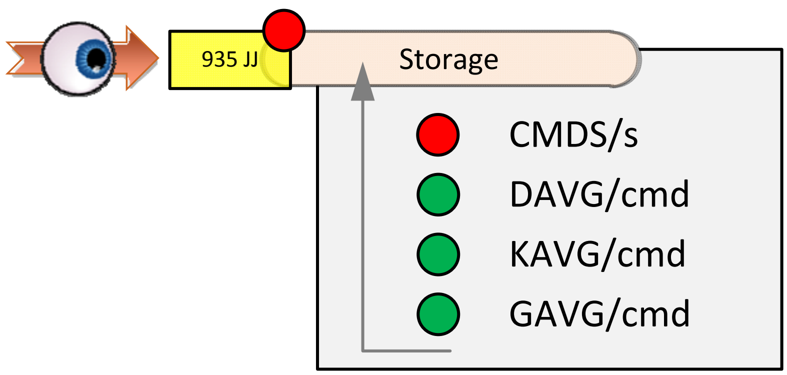

Figure 1.10: Drilling down.

That drill down shows that the performance threshold being crossed relates to storage commands per second and not command latency. These specifics further assist in finding the right action to fix the problem.

A Note on Tuning

You might be thinking at this point, "Well, that's great, but every environment is different, as is every business cycle. Tuning these models can be a nightmare of effort."

It's a valid concern, as are the false positives a poorly-tuned black box can create. Your measurement of *what's acceptable* is surely different than the next person's.

Different solutions take different spins on this modeling approach. One key difference is in how the metrics are tuned—manually, automatically, or a combination of both—over time. The solution you want gives you the flexibility to tune your metrics while automatically making adjustments to fit your IT services and your business cycle.

Extending into the Cloud

Not surprisingly, this top-down approach to performance management continues to serve a purpose as a virtual environment extends into the cloud. Its integration of component-specific metrics into an overall service model is extremely useful when managing assets atop hardware you might not own.

Figure 1.11 shows graphically one way those metrics can be exposed. Virtual machines in the hosted model tend to be driven by the same hypervisor technologies that power the on-premise Private Cloud. Being the same, those technologies expose similar APIs and/or other endpoints that can be consumed by an on-premise monitor.

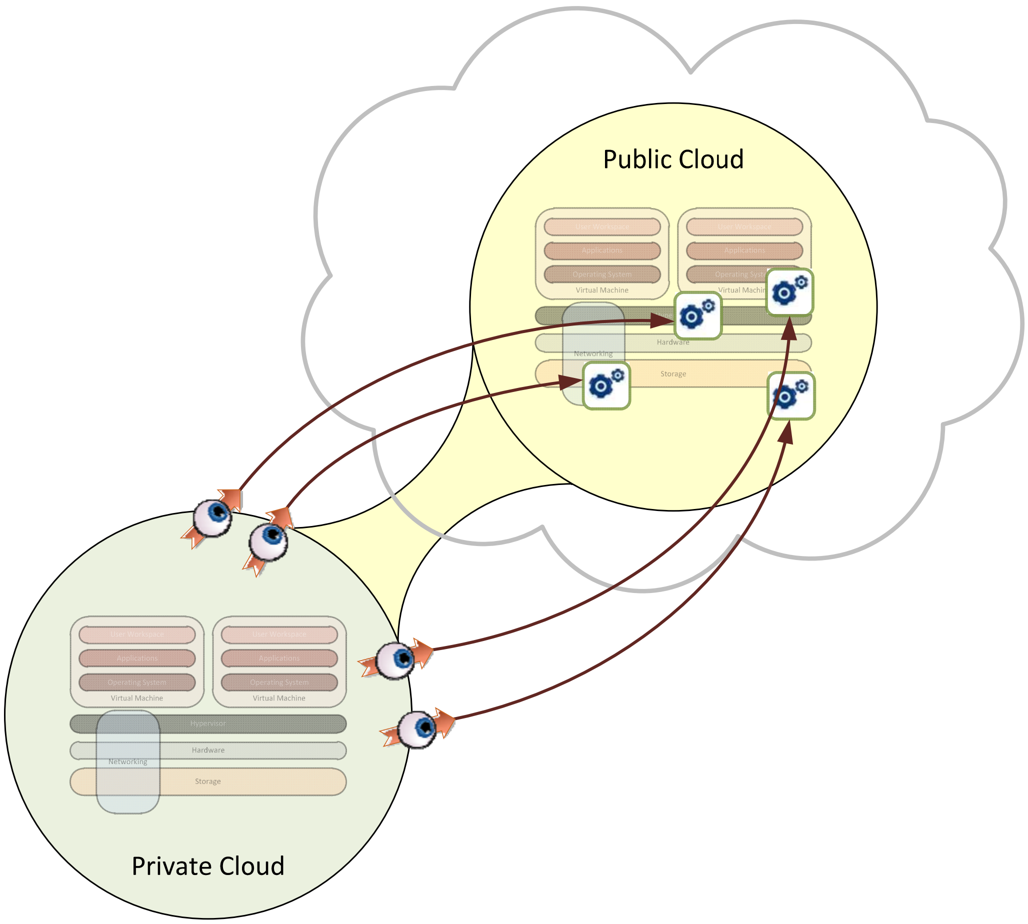

Figure 1.11: Local monitors; remote metrics.

As before, using a unifying solution to consolidate metrics across multiple components—and in this case multiple locations—enables correlating behaviors across an entire distributed system. In Figure 1.11, metrics in each virtual machine, at the hypervisor, and within the provider's storage and networking are collected and processed by the local performance management solution.

Performance Management Requires a Solution

Performance counters are great. They give you an idea of what's going on inside a computer system. But all by themselves, performance counters are an absolute fire hose of information. Unless you're watching them constantly, and making sense of how each impacts the others, it becomes easy to get overwhelmed at the data they present.

That systemic overload can be a key reason many IT professionals today aren't doing performance management (either effectively or at all) in their data centers. The activity is simply something no human alive can accomplish without assistance.

In a world where unlimited resource supply is now considered a waste of hardware investment, today's virtual environments are striving to make best use of every dollar spent. That desire for optimization, as you've learned in this chapter, mandates a change to our old ways of thinking. That change arrives in the solutions that now exist to automate much of the number crunching. Implementing such a solution for performance management in your virtual environment *is the new best practice*.

Quantifying the activities of performance management isn't the only way these solutions bring an important assist to today's virtual infrastructure. They're also extremely handy at measuring capacity. Although performance management and capacity management are very different activities, the types of new best practice solutions you'll want to implement are absolutely critical for accomplishing both tasks. The next chapter will focus the discussion on capacity management. You might be surprised at how the right focus simplifies answering the question: What do you have versus what do you need?