Realtime publishers

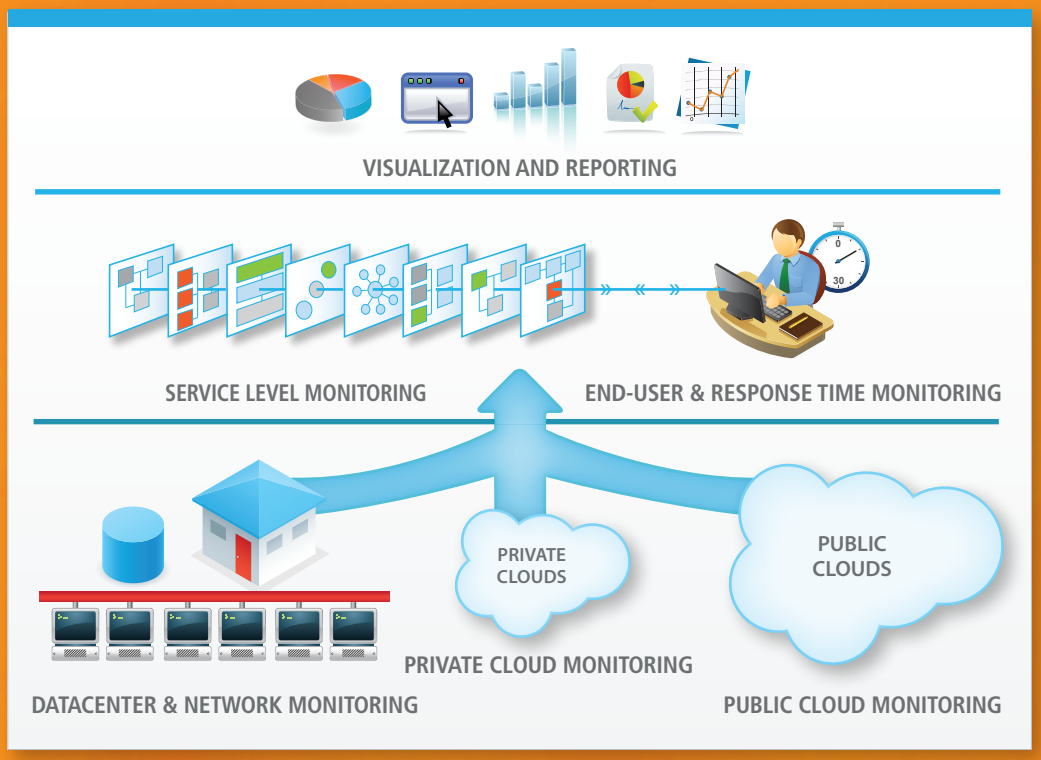
The Definitive Guide™ To

Monitoring the Data Center, Virtual Environments, and the Cloud

sponsored by

Don Jones

The Nimsoft Monitoring Solution



- **Ensures business service delivery regardless of IT platform**
- **Enables rapid adoption of new computer infrastructure such as private and public cloud**
- **Monitors the datacenter to the cloud, including SaaS, hosted, and virtualized environments**
- **Lowers TCO by 80% and delivers proven value in weeks**

Unified Monitoring

nimsoft
From the Datacenter to the Cloud

www.nimsoft.com

Copyright Statement

Chapter 5: The Capabilities You Need to Monitor IT from the Data Center into the Cloud

In the previous chapters of this book, I've covered a lot of the "why" and "how" of unified monitoring. In this chapter, I want to start focusing specifically on the "what:" What capabilities you need to bring into your environment to successfully manage and monitor a hybrid IT infrastructure and its applications. Think of this chapter as an instructional guide for building a shopping list. I'll cover not only features but also some of the finer, easily-overlooked details that can make all the difference in a successful implementation.

First, though, let's clearly define some of the major business and technical goals for this kind of evolved, hybrid IT monitoring. As I do so, I want to re-introduce the works from World Coffee, the case study I introduced in Chapter 1.

Business Goals for Evolved Monitoring

I want to cover the business' goals for monitoring first because in reality the business' goals are the only ones that matter. The business is paying for not only the monitoring solution but also the applications and infrastructure being monitored; meeting the goals of the business is really the whole point of all of this. So what might a business hope to gain from a more evolved form of monitoring?

EUE and SLAs

The business' primary concern, of course, is to have applications and an infrastructure that perform well. A problem—one I discussed in the first chapter of this book—is that many businesses have given up on simply saying, "we want our applications to perform well," and have instead gotten themselves bogged down in the minute details of application performance. But knowing that "Server5 is running at 80% utilization" isn't truly a business goal—although many businesses have accepted that this is how they have to define "good performance."

They shouldn't accept that. Instead, businesses should back off a level and concern themselves with what *really* concerns them: Applications that perform, from an end user perspective, in a way that supports the business' requirements. That's the *end-user experience,* or EUE, that I've referred to throughout this book. "We want users to be able to complete their checkout process in 5 seconds or less from the time they click 'Submit Order.'" That's an EUE-focused metric, and it can become a part of formal Service Level Agreements (SLAs), which are used to communicate the desired EUE-level performance across the business and its IT team.

Let's be clear on something: A monitoring solution that doesn't allow you to quickly determine the current EUE metrics and that doesn't help you manage to an SLA-defined EUE metric is not a monitoring solution you should be using. Different vendors take very different approaches to how they show you the EUE, how they determine the EUE, and so forth; those are technical details that are important, but the most important thing is that the solution give you some means of managing the EUE.

EUE: All That Matters to the Business

Business have been unaccustomed to dealing with an EUE metric for so long that many will resist the concept of relying solely on the EUE—simply through force of habit. I had a recent consulting client tell me that they were happy defining an EUE like, "Sales orders must be accepted by the system in 2 seconds or less after submission." But they still wanted to add other things to their SLA, like that the system must have "a minimum 99.5% uptime." Think about it: If the system is down, it isn't accepting orders in 2 seconds or less. So you didn't meet the EUE metric. There's no need to specify anything else.

"We want to put maintenance windows in the SLA," they told me. Well, that's fine—make it part of the EUE. Sales orders must be accepted within 2 or fewer seconds…"between the hours of 7am and 7pm; outside of those hours, sales orders do not need to be accepted by the system." That makes it clear what end users should expect of the system—it might not be available to them from 7pm to 7am. By stating things in that kind of end-user context, you're communicating not only your desires to your IT team but also your commitment to your end users. Everyone's using the same language. End users don't have "maintenance windows" after all; they have expectations for when they'll be able to do their jobs. State your SLAs in those terms, and manage those expectations.

"We want to add a clause that systems must not run at more than 80% capacity." They really insisted on that one, but I eventually convinced them that such a metric might be a good *IT management guideline,* it didn't belong in an SLA. So long as the EUE metrics were met, it wouldn't matter how burdened the systems were. If IT could meet that 2-second rule with a 95%-loaded server—more power to them! They'd be saving money by doing more work with fewer resources. And what does "95% loaded" mean, anyway? 95% processor capacity? Network throughput? Disk I/O? Don't dive into the technical details within an SLA: Try to stick with EUE-based metrics that describe your desired bottom-line performance, and make sure IT has the tools they need to manage the technical components to your EUE-based SLA.

Let's check back in with World Coffee on this. I'll re-introduce some of these folks, as we haven't heard from them in a couple of chapters.

> Ernesto is an inside sales manager for World Coffee, a gourmet coffee wholesaler. Ernesto's job is to keep coffee products flowing to the various independent outlets who resell his company's products. Like most users, Ernesto consumes basic IT services, including file storage, email, and so on. He also interacts with a customer relationship management (CRM) application that his company owns, and he uses an in-house order management application. Ernesto works on a team of more than 600 salespeople that are distributed across the globe: His company sells products to resellers in 46 countries and has sales offices in 12 of those countries.
>
> Ernesto's biggest concerns are the speed of the CRM and order management application. He literally spends three-quarters of his day using these applications, and much of his time today is spent waiting on them to process his input and serve up the next data-entry screen. He needs that process to be quick and efficient. When he looks up information or submits new information, such as sales orders, he needs the system to be responsive. Every minute he spends waiting is a minute he's not selling, and those wasted minutes can add up quickly.

Budget Control

As businesses start moving toward hybrid IT environments and applications and incorporating outsourced components, budget starts to become a very real concern. For example, consider how internally-hosted applications and services are priced: The business pays some up-front cost to acquire a solution, often some kind of recurring maintenance, and will have some amount of IT staff time spent on supporting the solution. Those costs are relatively easy to determine, and are pretty much fixed. If the business has a really busy week, the application will cost about as much to support as it would in a really slow week.

Now consider how cloud services are priced. Some, like many SaaS solutions, may be priced per user, based on storage consumption, and so forth—relatively easy to track, trend, predict, and control. Other services, however, are priced based on actual usage. Here's the current pricing, as of August 2010, for Microsoft's Windows Azure cloud computing platform:

- Compute = $0.12/hour
- Storage = $0.15/GB stored/month
- Storage transactions = $0.01/10K
- Data transfers = $0.10 in/$0.15 out/GB—($0.30 in/$0.45 out/GB in Asia)

This is a fairly typical pricing model for cloud computing; companies like Rackspace, Amazon, and others all have similar pricing models. You're paying *based on usage.* What will your monthly bill be? There's no way to know in advance.

This is where a truly unified monitoring system can help. In addition to tracking raw performance and high-level EUE metrics, a solution can also keep track of your service consumption. It can help you see how much you're using, and therefore how much you'll be paying. You'll be able to keep an eye on your costs, and relate those costs to the income produced by your hybrid applications. If an application is consuming more than it is returning, you'll be able to address the problem *before* your bills start getting out of hand.

This is an entirely new territory for monitoring software. It's made more complicated by the fact that some applications are *really* super-distributed across different hosting providers. Consider, for example, Figure 5.1. This is an example I've used before, but it bears repeating in this new, budgetary context.

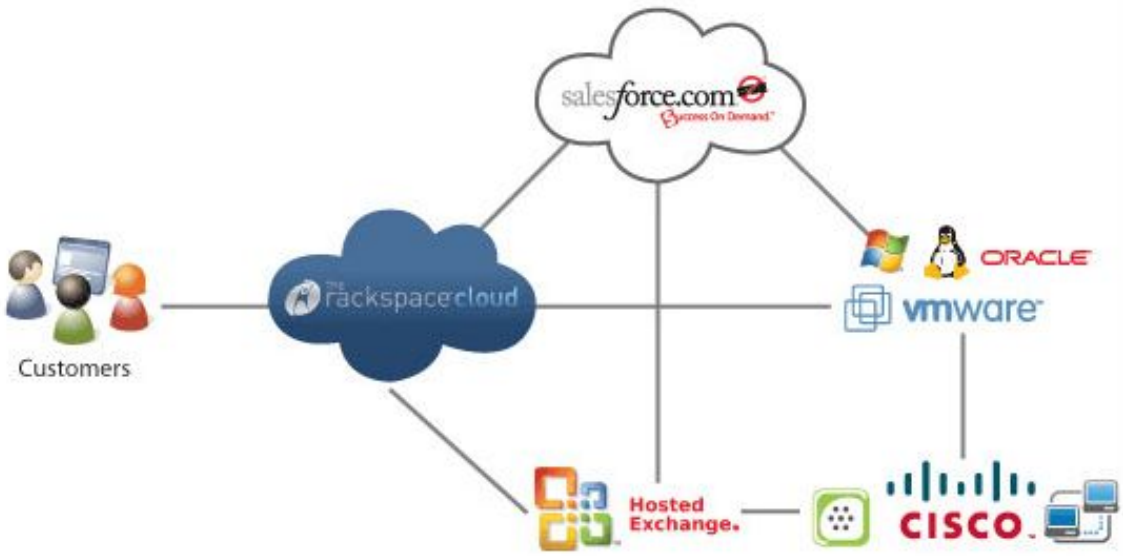


Figure 5.1: Example of a super-distributed hybrid application.

This application has a certain number of resources in your own data center—shown by the Windows, Linux, Oracle, and VMware icons. Those costs, as I've said, are relatively fixed and predictable. The application also relies on SalesForce.com, which is an SaaS offering, and on the cloud computing platform Rackspace Cloud. Your use of SalesForce.com might be per-transaction (perhaps it's being used to issue license keys to customers), and your cloud-computing costs will likely be based on actual usage as well. Being able to track that usage, and therefore those costs, across *all those different platforms* can be quite complex. The business has a clear need for a monitoring platform that can unify all of that information into a single place so that you can get a true and accurate picture of your costs.

> John works for World Coffee's IT department and is in charge of several important applications that the company relies upon—including the CRM application and the in-house order management application. World Coffee has moved to a hybrid IT application for its in-house order management. The CRM element is now outsourced to SalesForce.com, an SaaS provider; the in-house order management application is Web-based, and runs in Amazon's EC2 cloud computing platform. Amazon is also used for customer-facing ordering applications.
>
> John needs to make sure that every user of every system is experiencing response times at or better than those defined by the company's SLAs. This task is difficult because many of the applications' elements are outside his direct control. He needs ways to directly test the EUE as well as ways to check on the direct response times for the various SaaS, cloud computing, and other outsourced IT elements.
>
> John is also responsible for providing his boss with information on how much all of this is costing the company. Since the switch to hybrid IT, the company has spent more on outsourced IT services than they anticipated. They've seen an increase in sales volume, so it's likely that the additional expenses are justified, but they need some way to closely track actual usage and charges so that they can relate that more directly to the resulting sales volume.

Technology Goals for Evolved Monitoring

IT's job is to implement the SLA that the business has agreed to. Their job is to make sure that the EUE metrics needed by the business can be delivered within whatever parameters the SLA outlines. That means IT essentially needs tools that can tell them when the EUE is starting to go wrong, and help them find the root cause of the problem so that they can fix the problem before the SLA is missed.

Centralized Bottom-Up Monitoring

Today, one of IT's biggest challenges is that they simply can't get enough information onto a single, consolidated screen. Instead, they're stuck looking at numerous consoles, as illustrated in Figures 5.2 and 5.3. One for the database. One for VMware. One for Windows. One for Active Directory (AD). One for Exchange. One for the *other* database. In addition, few of these consoles have any ability to look into outsourced systems like SalesForce.com, Rackspace Cloud, Amazon EC2, Google AppEngine, and so forth. These consoles offer *no* means of calculating the EUE; thus, they can't tell you when you're meeting the EUE metric or not.

Figure 5.2: Database performance.

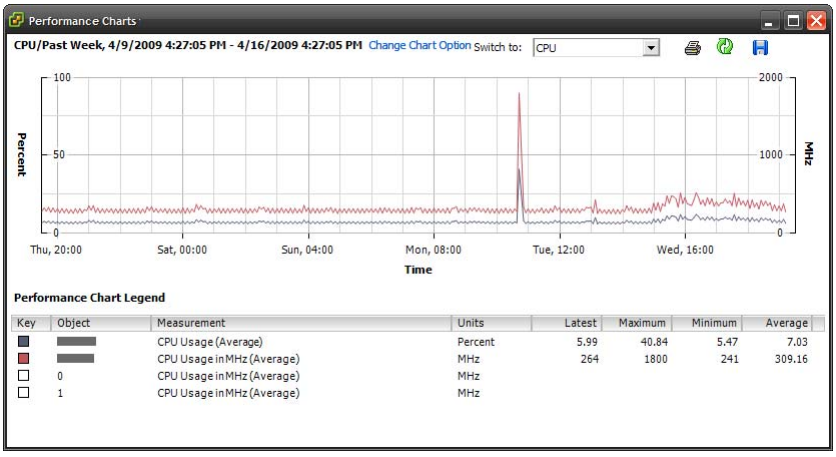


Figure 5.3: VMware performance.

EUE metrics cannot be *derived* from looking at component performance; the EUE must be directly *observed,* and it takes a central, unified monitoring solution capable of doing so to get an accurate EUE measurement. If you're not meeting your EUE metric, however, you still need to look at the individual components' performance to find out which ones are contributing to the reduced performance. Again, a *unified* monitoring console give you this capability by putting every component's performance right in front of you—including outsourced elements like cloud computing platforms, hosted services, SaaS services, and so on. So that's the technical need: Everything in one place. It's the only way to meet the business' EUE-centric SLAs.

Improved Troubleshooting

IT departments also have a need for more streamlined, efficient troubleshooting. When something does start to go wrong with application performance, the IT team needs to be able to quickly pinpoint the cause of the problem and bring domain-specific tools to bear so that the problem can be solved quickly.

A unified monitoring platform is generally regarded as the best way to avoid the "siloing" that can occur during IT troubleshooting scenarios. In other words, by getting every team member on the same screen, with the same information, everyone can agree more quickly on which major application element is causing or contributing to a problem—rather than each team member using individual domain-specific tools and independently stating that "their" component is "working fine." Once affected application elements are identified, either the unified monitoring solution or domain-specific troubleshooting tools, or a combination of both, can be used to further refine the root cause of the problem and to discover a solution.

The key is *getting everyone on the same page.* A unified monitoring solution does this by presenting similar statistics for a database server, virtualization host, Windows server, cloud computing platform, and so on. While each of these elements will obviously have a variety of different performance metrics that need to be examined, by bringing them all together into the same place, and presenting them similarly, the solution can create a sort of level playing field, providing a more authoritative starting point for troubleshooting.

A Shopping List for Evolved Monitoring

Those are our business and technology goals. Now we need to outline the exact capabilities that an evolved, hybrid IT monitoring system needs.

High-Level Consoles

It's a screen shot I've shown you before, but the first and most important aspect of a truly unified monitoring system is a high-level console that gives you a broad, dashboard-style view of your overall application. Figure 5.4 shows an example.

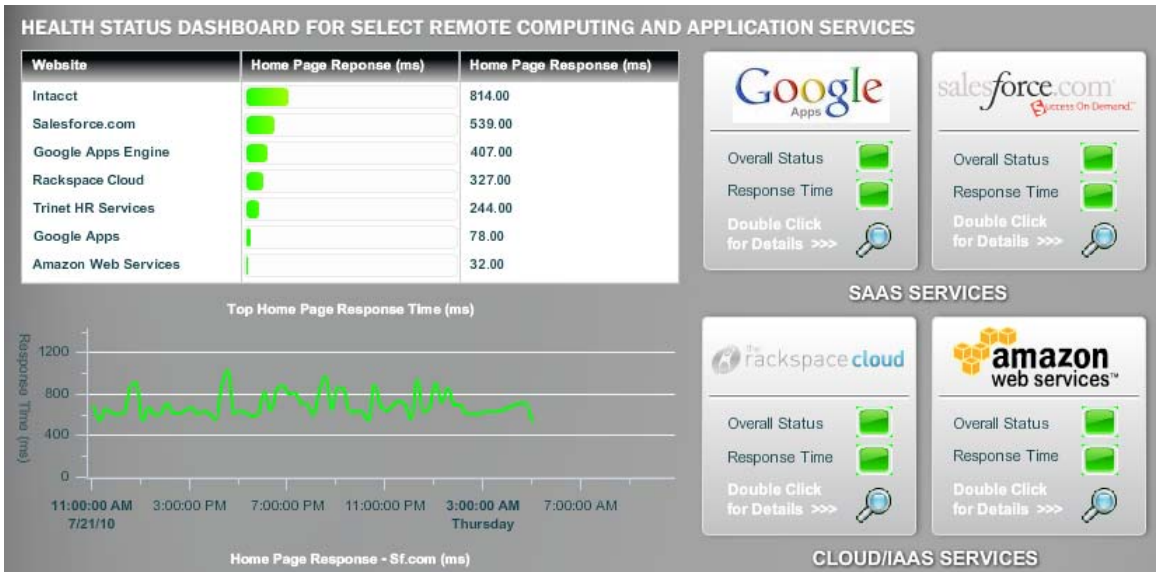


Figure 5.4: Top-level, unified monitoring dashboard.

This dashboard provides at-a-glance information for several cloud-based services, and offers an overview chart of response time from those services. A line chart shows historical response times for the past few days or hours, helping administrators quickly identify performance trends, spot weak links, and so forth. Drilling down into any of these services provides additional information. Additional panels might include your own datacenter; Figure 5.5 shows what a next-level drill-down into that data center might look like. Here, we can see a more-detailed view of what's happening in the data center. Busy routers are highlighted, and graphs show top utilization levels for storage, memory, processor, and so forth. We're essentially treating our data center as a kind of "cloud" that's fully under our control. This second-level drill-down lets us dive into the cloud and see some of the individual elements that run it.

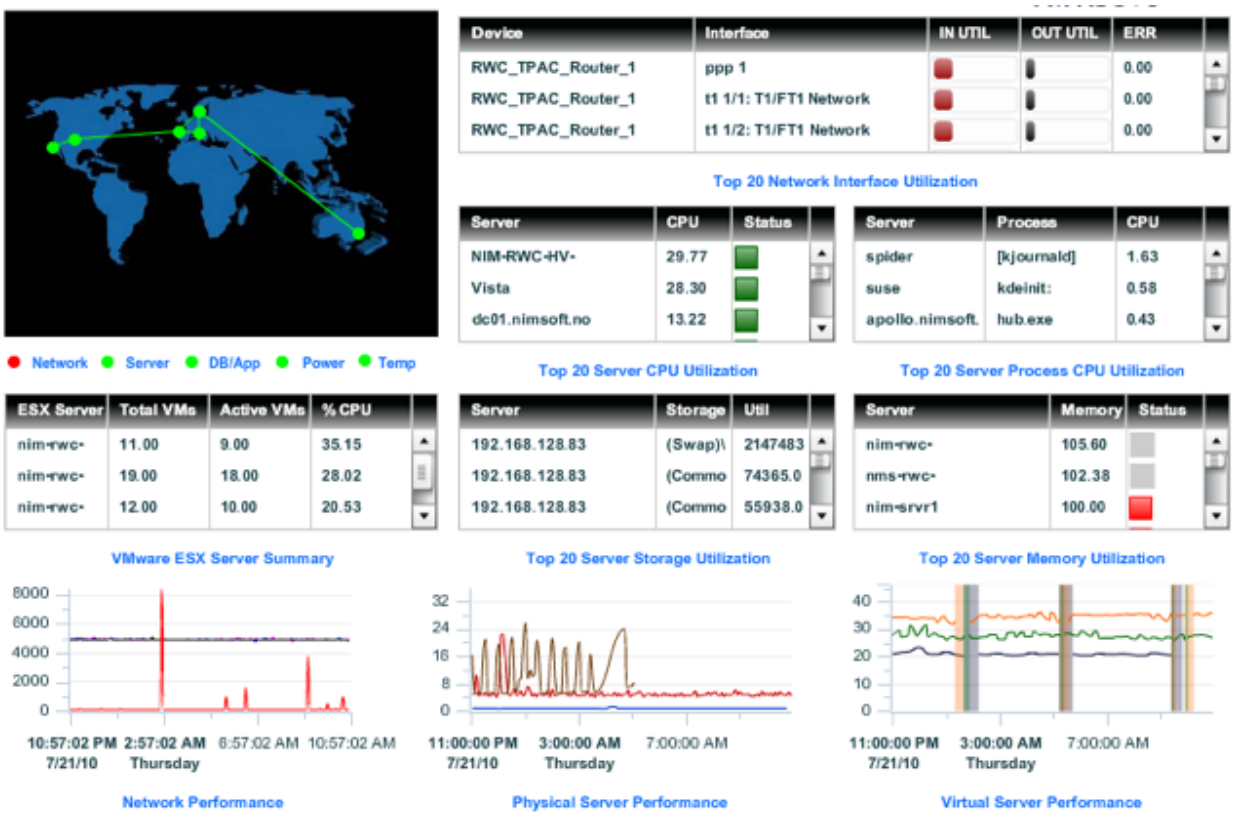


Figure 5.5: Drilling down into a data center.

For other data centers—our cloud providers, for example—we might not get that same level of detail. After all, the "cloud" is supposed to just be a big bucket of functionality and services, not individual servers. So the drill-down here might offer a different kind of detail, as Figure 5.6 shows.

Here, we're presented with information on specific service instances, response times (over time), and the number of transactions we've been sending to this provider—a key in helping us meet that business requirement of monitoring actual usage. Every cloud provider's drill-down might be a bit different, as each one works somewhat differently.

Figure 5.6: Drilling-down into a cloud provider.

When there's a performance problem with a cloud provider, this is likely as far down as we'd drill; the next step is to get them on the phone and find out what's happening. Within our own data center, however, we'd likely want to drill a bit deeper.

Domain-Specific Drilldown

Within our own data center, having additional levels of detail can help focus troubleshooting efforts. For example, in Figure 5.7, we can see alerts generated from a specific server as well as summary information for key performance metrics from a variety of servers.

By configuring performance thresholds, we can receive alarms when something looks wrong. These should be from across all our servers, whether they're running Windows, Linux, Unix, or whatever. In fact, in the figure, you can see that both Windows servers running SQL Server and Red Hat Linux computers are included in the list of alarms. Getting all of this information onto the same page will help direct efforts to resolve these alarms.

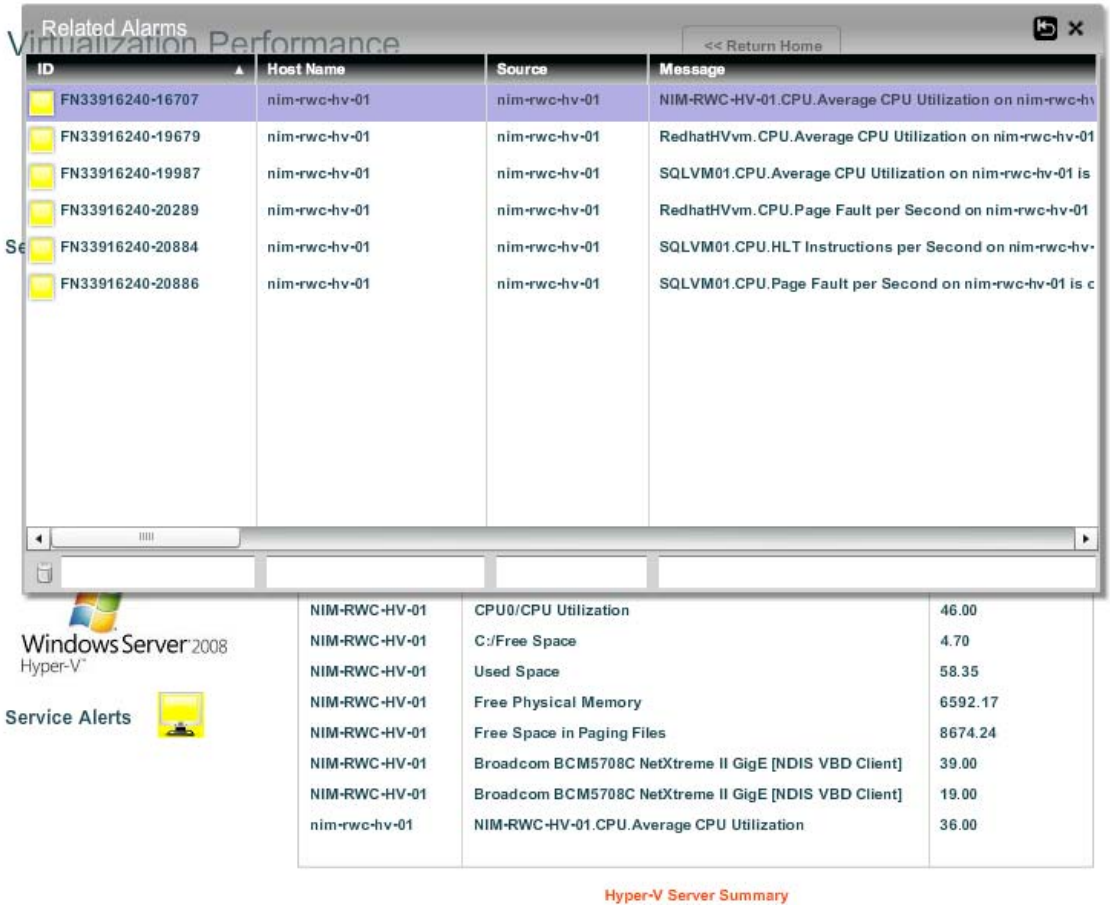


Figure 5.7: Drilling deeper into specific servers.

Performance Thresholds

Ideally, a monitoring solution will come preconfigured with performance thresholds based on the vendor's experience with the technologies involved. Commonly, you'll also be able to define your own thresholds, perhaps defining a larger pad between "good performance" and "bad performance" to give your team more time to react. As Figure 5.8 shows, these thresholds should be used to create instantly-readable visual displays: Indicators, graphs, and gauges that help draw your eye to elements that need the most immediate attention.

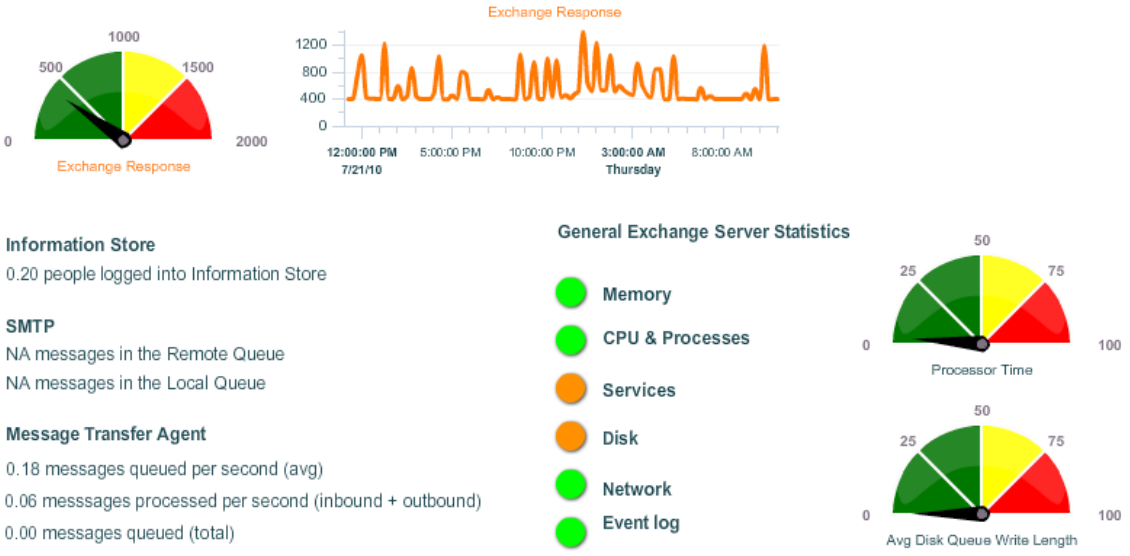


Figure 5.8: Performance thresholds help drive graphical displays.

Thresholds will be different across different technologies. This example is for Exchange Server, so it includes information about message queues and transfer agents in addition to more generic metrics such as memory, CPU, disk, and network measurements.

Broad Technology Support: Virtualization, Applications, Servers, Databases, and Networks

A unified monitoring solution is only as useful as the number of your technologies it can unify into a single console—and ideally, you want a solution that can handle everything you've got. There are subtle differences in how solutions work. Here are some considerations:

- Virtualization
  - Look for VMware, Hyper-V, Citrix, Sun, and IBM support
  - Consider support for agentless monitoring, which requires less impact on your infrastructure and means less long-term support and maintenance
  - Look for templates that provide a starting point for Quality of Service (QoS) metrics
- Applications
  - Look for Messaging and Directory Services support for Exchange and AD at a minimum, along with any other technologies you have in house, such as Lotus Notes/Domino
  - For Web servers, Internet Information Services and Apache support are both desirable
  - Collaboration platforms should be supported, too, including Lotus Notes/Domino and Microsoft SharePoint

  - You might need support for Voice over IP (VoIP) services, such as Cisco VoIP components
  - Also consider support for application services, such as those from Citrix, IBM WebSphere, IBM WebLogic, JBoss, Tomcat, and Sun's Java Virtual Machine

- Servers—You probably have a variety of servers in your environment, and you want to make sure they're all included in your monitoring solution:
  - AS/400 (for example, Figure 5.9 shows how a monitoring solution might raise alarms for IPL or reboot conditions)
  - Linux
  - Unix
  - NetWare
  - Windows

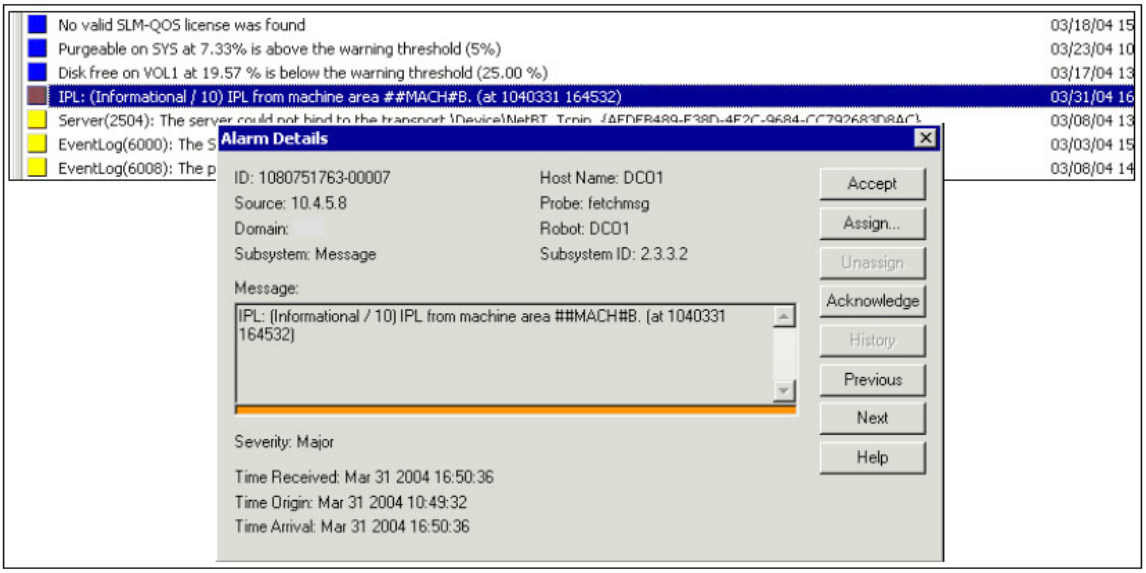


Figure 5.9: Viewing AS/400 alarms.

- Databases—These form the backbone of most modern applications, and you don't want to be tied to just one or two simply because they're all your monitoring solution supports; get maximum flexibility by looking for support for:
  - Sybase
  - Informix
  - Oracle
  - Microsoft SQL Server
  - DB2
  - MySQL

- Networks—The infrastructure that connects everything can play a critical role in your applications' performance; look for a solution that can monitor:
    - Cisco's IP SLA—Figure 5.10 shows a monitoring dashboard for Cisco SLA, which is critical on modern converged network carrying voice, data, video, and other traffic
    - Core DNS, DHCP, and LDAP services
    - SNMP management information
    - Routers and switches
    - Raw network traffic

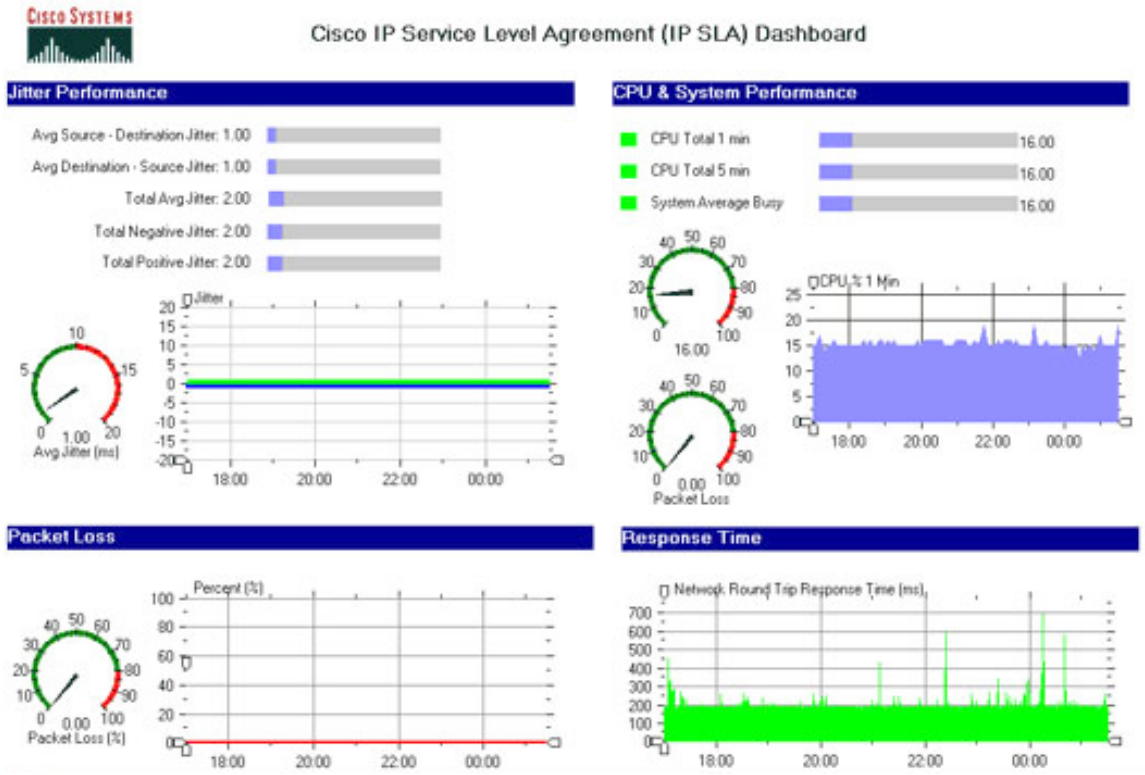


Figure 5.10: Monitoring Cisco IP SLA.

End-User Response Monitoring

There are two main kinds of EUE monitoring: Active and Passive. With Active, you can actually set up "synthetic" transactions to feed into your system. The monitoring solution can trace those, and accurately report on response times. You might end up with a screen like the one in Figure 5.11, showing granular detail for user transactions.

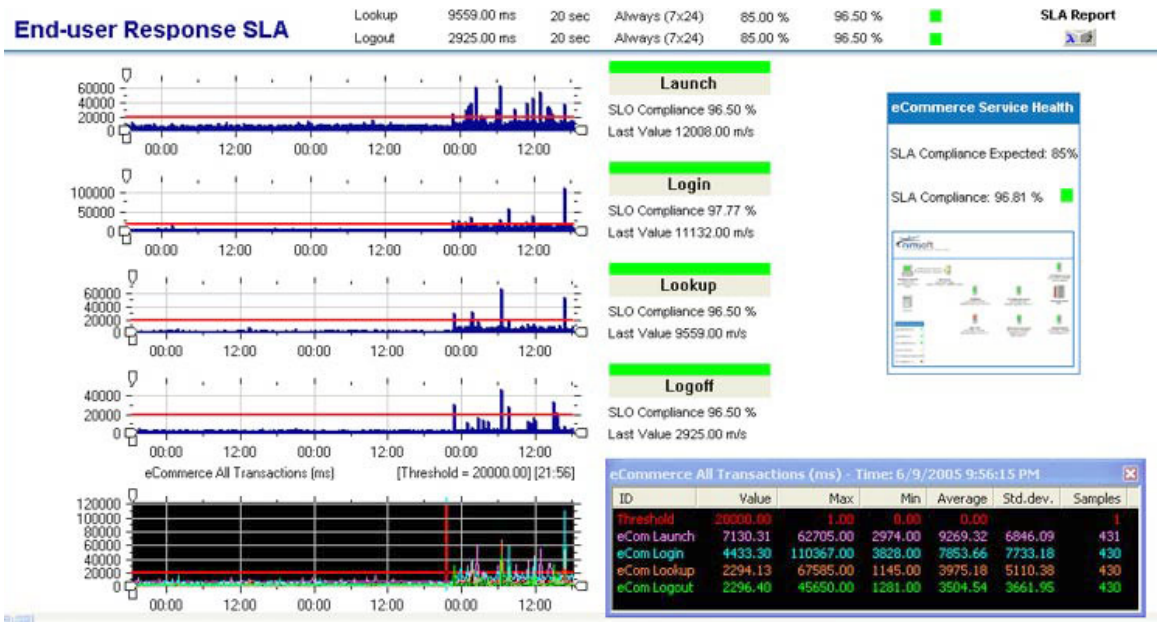
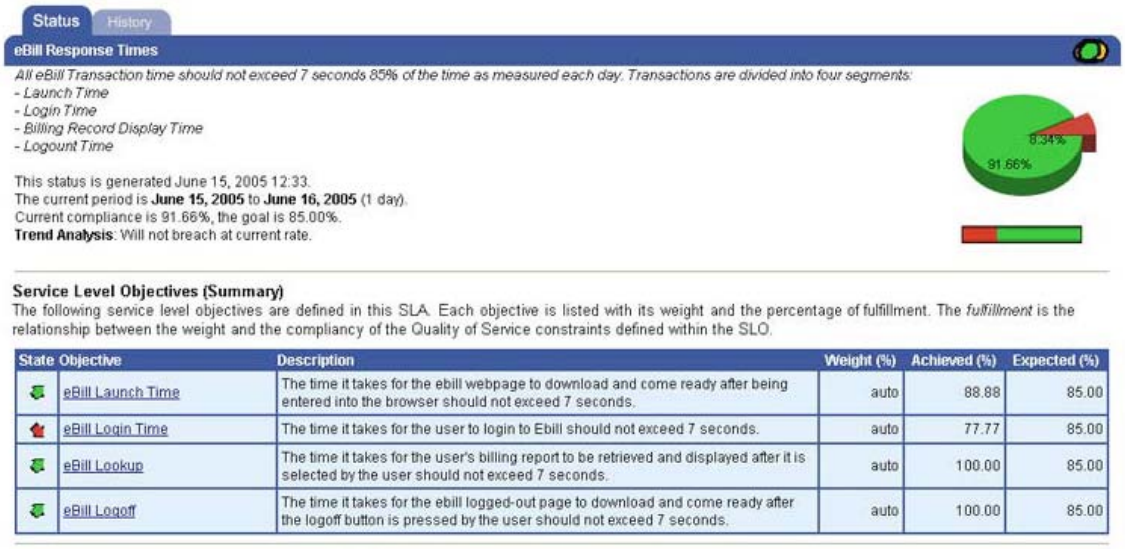


Figure 5.11: Active user transaction monitoring.

Passive monitoring doesn't inject transactions into the system but rather monitors the individual system elements and creates an aggregated EUE metric. It isn't always as accurate as active monitoring, but it's completely non-intrusive. Most organizations use both active and passive monitoring, and you should look for a solution that present both because the information they generate is largely complementary.

SLA Reporting

This is something we'll dive into more in the next chapter, but SLA reporting is absolutely a capability your monitoring solution must provide. Even a straightforward report like the one Figure 5.12 shows can be tremendously useful. It shows a breakdown of specific elements of the application—launch, login, lookup, and logoff—and indicates which elements are meeting their SLA. It also gives you the good or bad news, indicating whether, at current rates, you are trending toward a breach in your SLA.



Status | History

eBill Response Times

All eBill Transaction time should not exceed 7 seconds 85% of the time as measured each day. Transactions are divided into four segments:
- *Launch Time*
- *Login Time*
- *Billing Record Display Time*
- *Logount Time*

This status is generated June 15, 2005 12:33.
The current period is **June 15, 2005** to **June 16, 2005** (1 day).
Current compliance is 91.66%, the goal is 85.00%.
Trend Analysis: Will not breach at current rate.

Service Level Objectives (Summary)

The following service level objectives are defined in this SLA. Each objective is listed with its weight and the percentage of fulfillment. The *fulfillment* is the relationship between the weight and the compliancy of the Quality of Service constraints defined within the SLO.

State	Objective	Description	Weight (%)	Achieved (%)	Expected (%)
	eBill Launch Time	The time it takes for the ebill webpage to download and come ready after being entered into the browser should not exceed 7 seconds.	auto	88.88	85.00
	eBill Login Time	The time it takes for the user to login to Ebill should not exceed 7 seconds.	auto	77.77	85.00
	eBill Lookup	The time it takes for the user's billing report to be retrieved and displayed after it is selected by the user should not exceed 7 seconds.	auto	100.00	85.00
	eBill Logoff	The time it takes for the ebill logged-out page to download and come ready after the logoff button is pressed by the user should not exceed 7 seconds.	auto	100.00	85.00

Figure 5.12: An SLA report.

Public Cloud Support: IaaS, PaaS, SaaS

Finally, because so many applications are becoming reliant on cloud-based services, your monitoring solution must be able to include them. Whether you're using Infrastructure as a Service (IaaS—think cloud computing like Amazon's), Platform as a Service (PaaS—think cloud computing like Microsoft's Azure), or Software as a Service (SaaS—think SalesForce.com). Popular choices you should absolutely be able to include in your monitoring are:

- Amazon EC2 and S3
- Rackspace Cloud
- Google AppEngine
- Windows Azure
- SalesForce.com CRM

The Provider Perspective: Capabilities for Your Customers

As a Managed Service Provider (MSP), you become a part of your customers' IT teams. Typically, that means you have a dual problem that your customers don't often face:

- You need to maintain, monitor, and manage your own network for your own reasons. After all, you want to provide excellent services to your customers.
- You need to help your customers include your network and applications *in their own monitoring.* There are a number of ways in which you can do this (many of which we'll discuss in the next chapter), but the bottom line is that you need to provide your customers with some visibility into your infrastructure so that they can treat *your* systems as a true part of *their* systems.

> Li works for New Earth Services, a cloud computing provider. Li is in charge of their network infrastructure and computing platform and is working with World Coffee, who plans to shift their existing Web services-based order management application into New Earth's cloud computing platform.
>
> Li knows that he'll have to provide statistics to World Coffee's IT department regarding New Earth's platform availability because that availability is guaranteed in the SLA between the two companies. However, he also knows that he'll need to provide them with more detailed insight into certain aspects of New Earth's infrastructure. After all, World Coffee is essentially making New Earth's network a part of World Coffee's network through their hybridized IT applications—so as a customer, they deserve some insight.
>
> Li plans to search for monitoring applications that can be used by his internal network engineers that will also allow him to provide dashboards and reports directly to customers like World Coffee. That way, he doesn't have to build his own monitoring and data-provisioning mechanism.

MSPs often look for monitoring systems that have *multi-tenant* capabilities. In other words, the MSP can buy a monitoring system that lets them monitor their *entire* infrastructure while also providing monitoring capabilities directly to their customers—restricting each customer so that they can only see *their* portion of the MSP's infrastructure. Such monitoring systems are obviously more complex.

In some cases, the monitoring *itself* might be something that the MSP offers as an additional service. Imagine a conversation between Li, who works for an MSP, and John, who works for one of Li's customers:

> John: Will we be able to monitor the servers that we're using on your network?
>
> Li: It's possible. Let me ask, what sort of monitoring tools do you have now?
>
> John: We use a lot of different ones. We have some for Oracle, others for VMware, and others for Microsoft Windows. We don't have anything specifically designed for monitoring cloud-based resources.
>
> Li: You know, in addition to the cloud computing that we're providing you, we can provide you with a complete unified monitoring solution. It's basically a Software-as-a-Service offering. It can monitor your entire infrastructure in one console, including your Oracle, VMware, and Windows systems. It can also include monitoring of our systems so that you'll get your entire network—including those bits you've outsourced to us—on the same screen. There's minimal impact on your network, and you wouldn't be responsible for maintaining or patching the monitoring system—it's just a service we'd provide to you.
>
> John: I never knew such a thing was possible.

It is possible. Vendors are becoming increasingly creative and efficient at handling this kind of hybrid IT environment, and the ability for service providers to offer monitoring as just another service to their customers—well, it's compelling.

Coming Up Next…

In the next and final chapter of this book, I'll focus on one last set of capabilities that your monitoring solution should offer: reporting. It's very easy to get caught up in the actual details of monitoring, concerns about performance alerts, and setting thresholds and forget that management reporting is equally important. I'll look at different kinds of reports that can be used to help manage SLAs and keep the business on-budget as well as reports that show component-level health, usage trends, and so on. I'll also look at newer kinds of reports, including dashboards, in addition to ways in which you might want to leverage performance information elsewhere, such as data stores and application programming interfaces (APIs). For the MSP perspective, I'll also look at how things like multi-tenant capabilities can help deliver added value to your service offerings.

Download Additional Books from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this book to be informative, we encourage you to download more of our industry-leading technology books and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.