

Realtime
publishers

The Five Essential Elements of
**Application
Performance
Monitoring**

Don Jones

sponsored by



Chapter 5: Bringing It All Together: Managing Application Performance Data.....	72
Case Study: The Application Is Slow!.....	72
The End User Experience Is Looking Bad—and Getting Worse.....	72
Using the Transaction Profile to Identify the Problem Domain	73
Examining the Application Model to Target Problem Application Components	74
How It Happens: A Robust APM Database.....	76
APM Visualizations: The Key to Success.....	76
Creating a Single Version of the Truth.....	76
Providing Enterprise-Wide Management Information.....	77
Leveling the Field Between IT Specialists.....	78
Broad Coverage	80
Holistic APM Requires 5D, Not 2 or 3 or 4.....	82
Creating Your 5D APM Toolset Shopping List.....	83
Business Statistics	83
Usability and Troubleshooting	85
SLA Management	86
Don't Forget the Infrastructure	87
Modern Applications, Modern APM	87
Conclusion	88

Copyright Statement

© 2010 Realtime Publishers. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers (the “Materials”) and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers its web site sponsors. In no event shall Realtime Publishers or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at info@realtimepublishers.com.

[**Editor's Note:** This book was downloaded from Realtime Nexus—The Digital Library for IT Professionals. All leading technology books from Realtime Publishers can be found at <http://nexus.realtimepublishers.com>.]

Chapter 5: Bringing It All Together: Managing Application Performance Data

In this final chapter, I'll discuss the last dimension of the 5D application performance monitoring (APM) model. This last dimension is what brings the prior four together and makes them useful, so before I dive in, I'll present an example case study from my experience (names have been changed). This will help provide context and show how the 5D model can work in a real-world environment.

Case Study: The Application Is Slow!

I'll call this application Trax. It's a complex, distributed application designed to track customer order information. New orders are added to Trax by a separate order management system; the job of Trax is to provide both customers and customer service representatives with access to current order status and information. Customers access Trax via a Web-based interface; in-house service agents use a Java-based client application. I'll refer to Trax' owner as just "the company."

The End User Experience Is Looking Bad—and Getting Worse

The company's IT department is using an integrated, 5D toolset for APM. That system uses both active and passive user transaction monitoring:

- The solution actively injects synthetic transactions into the system to measure response times. It does so for the Web application and the internal application. Active monitoring like this can result in very accurate end-user experience (EUE) metrics, but it required minor modifications to the application to be able to ultimately identify and eventually discard the synthetic transactions.
- The solution passively monitors real user transactions as well. One means of doing so is to observe a particular transaction as it progresses through the system. In some applications, this doesn't result in the same level of accuracy as active monitoring, but it does offer other advantages.

The complement of active and passive monitoring provides the company with an EUE metric that is the basis of their service level agreements (SLAs). One morning, the IT department receives an alert that the EUE performance is trending poorly, and if it continues trending in the current direction, the SLA will be breached. In other words, real users are seeing reduced performance, and while that degradation isn't yet severe, it's heading in that direction.

Using the Transaction Profile to Identify the Problem Domain

Now that the IT team knows a problem exists, they need to find out exactly which application component is at fault—with the understanding that multiple components could be contributing to the problem, or that a single degraded component could be making everything downstream perform poorly. Trax is a complex application. It relies on a farm of Web servers; infrastructure components such as routers, load balancers, and switches; and a back-end database server. It also incorporates middle-tier servers that embody much of the application’s business logic. A variety of Microsoft Windows, Linux, and IBM components are used in the application, giving the IT team a lot of domain-specific tools to have to deal with. Many of these servers are actually running in virtual machines, adding another layer of complexity and potential performance problems. Tracing the exact cause of the problem manually can be difficult.

With the APM solution, however, the process should be a bit easier. One way is to simply use a tool that will allow you to manage to specific SLAs, as Figure 5.1 shows. Here, each element that isn’t meeting its SLA is clearly highlighted—giving your IT team a quick, actionable next step for solving the performance problem.

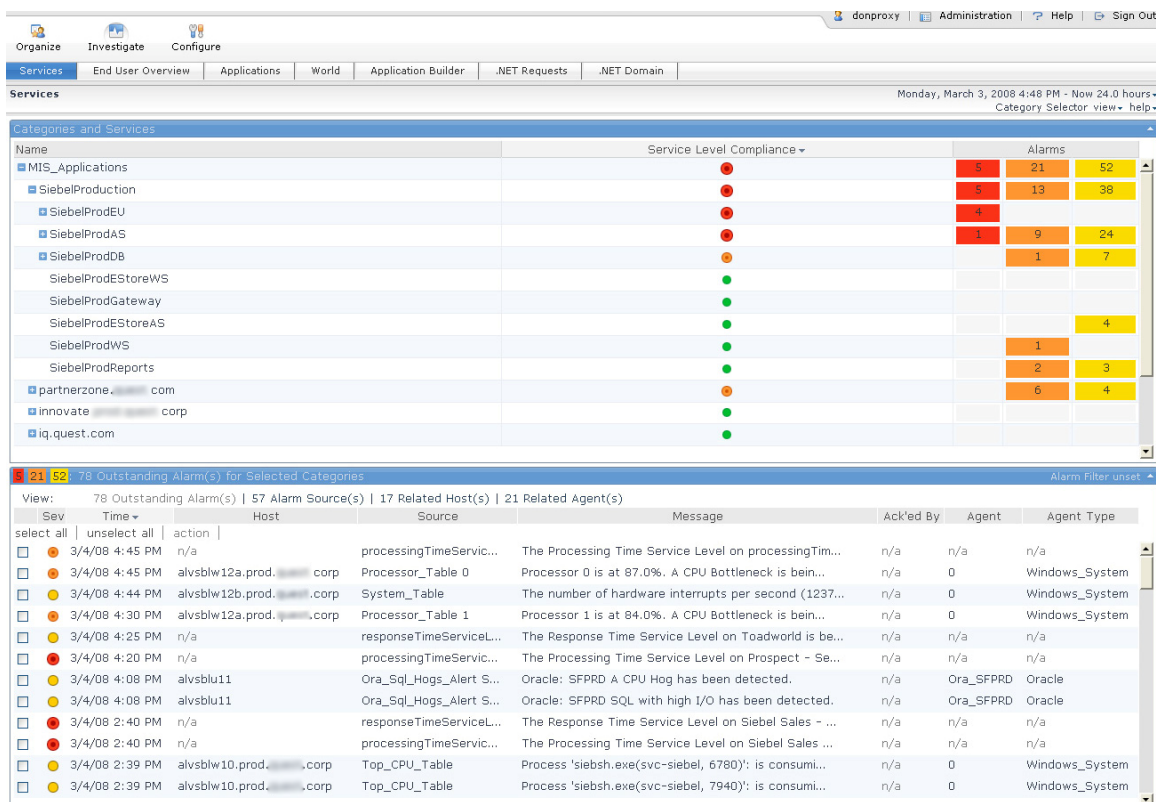


Figure 5.1: Troubleshooting by SLA.

Other types of visualizations might become useful once the general component performance problem is identified. For example, being able to switch from an application-centric view to a more physically-aligned view—as illustrated in Figure 5.2—can help identify specific components that are experiencing performance problems.

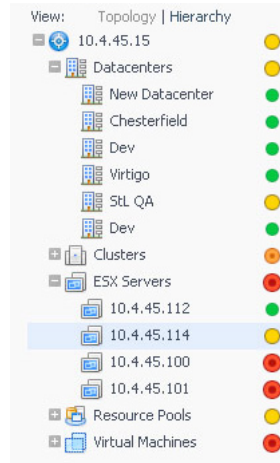


Figure 5.2: Identifying problem components.

Once a specific problem domain is identified—in this case, for example, it looks like some of the VMware ESX Server machines are experiencing performance problems—the problem can be quickly turned over to a domain expert who can troubleshoot the problem more specifically.

Examining the Application Model to Target Problem Application Components

Domain experts might immediately reach for their domain-specific troubleshooting tools—but, then again, they might not need to. A solid 5D APM solution will provide some of the domain-specific data and expertise needed to further pursue the performance problem. As Figure 5.3 shows, for example, the company’s VMware expert might now drill further to look at performance information specific to ESX Server.

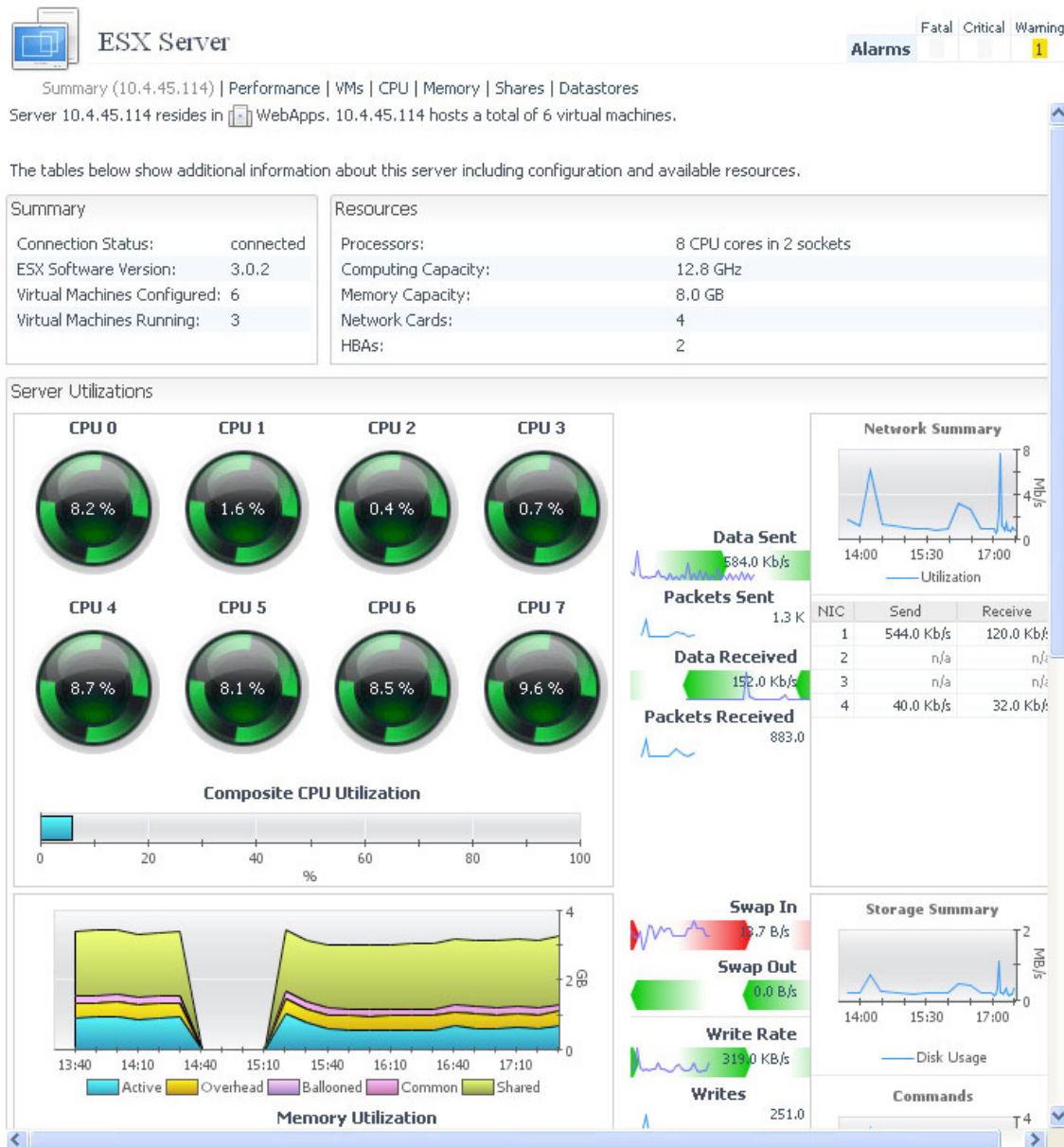


Figure 5.3: Troubleshooting with domain-specific information.

Having this domain-specific data right within the APM solution helps the IT team move more quickly through the application layers to arrive at the root cause. It might still come down to a domain expert relying on domain-specific tools to gain additional data or to actually solve the problem. However, by using a single application to quickly drive from “bad EUE” to “we know what’s broken” can shortcut many of the problems and inefficiencies typically associated with troubleshooting complex applications.

But there’s more to this solution than pretty gauges and graphs: All of that data needs to come from somewhere, and it needs to be filtered and correlated into useful information. That’s where the last, often-unseen, dimension of APM comes into play.

How It Happens: A Robust APM Database

In “Magic Quadrant for Application Performance Monitoring,” Gartner defines an APM database by first acknowledging that the first four APM dimensions—EUE monitoring, user-defined transaction profiling, application component discovery and modeling, and application component deep-dive modeling—can generate enormous data sets. That data needs to be filtered and often correlated with each other to yield useful information. They give an example:

An examination of accumulating end-user experience monitoring data determines that a response time slowdown is accelerating, and will soon hit a key customer's threshold of consciousness as he or she continues in the execution of a series of user-defined transactions. Attention shifts to the user-defined transaction-profiling dimension, and examining the data therein locates the source of rapid latency expansion to an application server running JBoss (by Red Hat) sitting in the Malaysia data center complex, while a study of the application model helps further specify the location of the server in the data center and its key configuration characteristics. An application component deep-dive monitoring tool is then brought to bear to determine which Java Virtual Machines and methods in the application server are doing the damage, while a second perusal of the application model helps the IT operations management and application support teams, working together, to determine what other elements of the infrastructure and application stack might be impacted.

The APM database is responsible for collecting this information from various elements of the APM toolset, filtering and correlating the data, and presenting it as useful, actionable information. In other words, the actual end-result of APM—properly monitoring and managing application performance—is delivered primarily by the APM database.

Ideally, you'll never even know that the APM database exists. You're unlikely to deal with it directly, but how it is built and how well it performs drives the other four APM dimensions that you *will* rely on and use directly.

APM Visualizations: The Key to Success

The real thing an APM database does for you is to get everyone on the same page, providing clear visualizations of a *single* set of underlying performance data.

Creating a Single Version of the Truth

The idea of a “single version of the truth” is a subtle part of what can make APM so successful for you. Rather than domain experts running around with their individual tools, each coming to a different conclusion about what the problem is, a *single* set of tools collects a *single*, consistent set of data, and presents it in a *single*, consistent fashion. Everyone agrees to work from that page, and so everyone can more easily agree on where problems lie and where action needs to be taken.

It's not impossible for someone's domain-specific tool to disagree with the APM toolset. However, that should be resolved by working with the APM toolset vendor to make sure they're bringing in accurate data. Don't rely on domain-specific tools for your first level of troubleshooting data because you'll be right back in the chaos of trying to decide who's right, who's responsible for making the fix, and so forth.

Providing Enterprise-Wide Management Information

An advantage of having all of that performance data in a single place is that you can more easily create visualizations beyond those that domain-specific technical tools offer. For example, Figure 5.4 is an example of an enterprise-wide dashboard.



Figure 5.4: Enterprise-wide management dashboard.

This dashboard shows performance statistics for key *applications* rather than application components. You can see the number of real users interacting with the application, see how data is flowing through the application, and so forth. This is an EUE-focused dashboard, showing information not only for *real* users but also for the synthetic transactions injected by the system's active monitoring.

This kind of information can finally raise APM awareness to managerial and executive levels, helping them feel confident that applications are working as desired—and holding IT accountable when they are not. Executives no longer have to rely on imprecise and inconsistent anecdotal user comments on the application's performance; they can instead use an SLA-backed dashboard that consistently and accurately displays the performance that end users are actually experiencing.

Leveling the Field Between IT Specialists

Unfortunately, not all technologies are created equally. Different technologies and platforms offer different tools, with differing levels of capabilities and detail. An APM solution can, to a degree, help level the playing field between those technologies. For example, suppose your organization relies on both Oracle and SQL Server as database platforms. Normally, the experts responsible for those systems will rely on vastly different native tools to monitor and troubleshoot their systems. Even though both are database platforms, it might be difficult to get an “apples to apples” comparison of their performance, simply because of the ways in which their native tools expose information. But consider Figure 5.5.



Figure 5.5: Monitoring SQL Server.

And now, take a look at Figure 5.6.



Figure 5.6: Monitoring an Oracle server.

These two views—one for SQL Server and one for Oracle’s database server—are slightly different, but they provide substantially the same information and present it in basically the same way. It becomes much easier for experts for both platforms to have an “apples to apples” comparison that will not only help resolve disagreements about which platform is doing what but also help non-domain experts more easily interpret the performance statistics for these two systems.

More importantly, having all of this information *in once place* can significantly speed the performance troubleshooting process. For example, by having the entire team accessing this single set of integrated consoles, they can quickly drill down to affected systems, change their mind, and drill down into something else if needed. For example, if the database server seems to be at fault, but a deeper examination suggests that the real problem is disk throughput, then drilling down into the underlying server operating system (OS) might be a smart next step. A good APM solution enables that kind of creative, exploratory troubleshooting—as shown in Figure 5.7—and enables experts to quickly move from application component to component, gradually refining the problem.

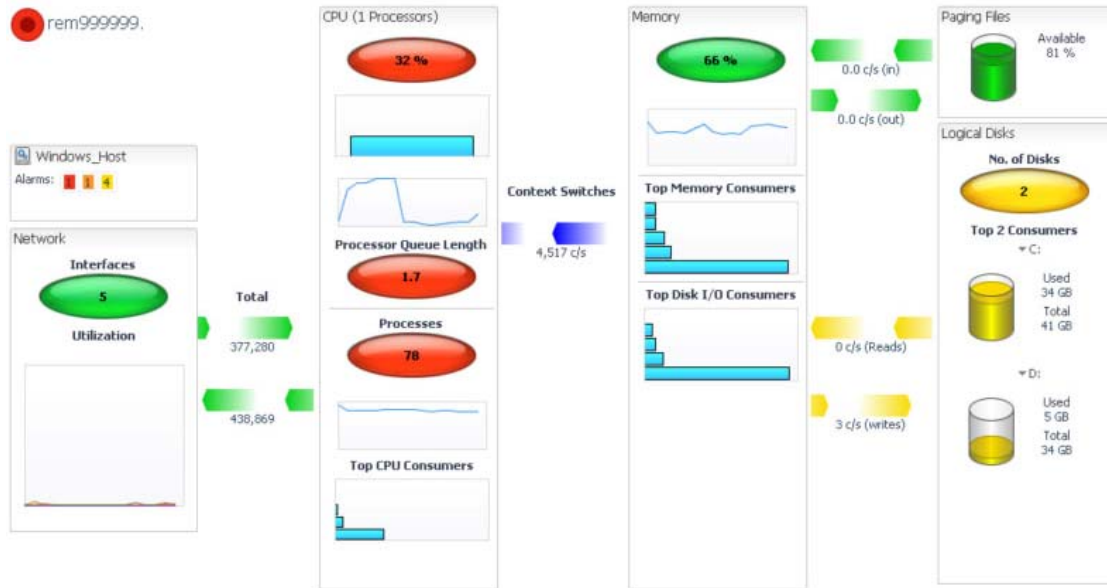


Figure 5.7: Moving to OS-level troubleshooting.

Broad Coverage

An APM solution is really only useful when it *can* put all of your application components into this integrated console. For example, it's not uncommon for modern applications to rely on external packaged software for some of its functionality. An online e-commerce application, for example, might integrate with Siebel or Oracle E-Business Suite on the back-end. Because those components are a part of the application, they have an impact on its performance. To truly do APM correctly, you'll need to be able to include those back-end components in the APM console. Figure 5.8 shows what that might look like, with an APM solution that includes support for Oracle E-Business. You might also look for support for SAP, PeopleSoft, Siebel, and so forth.

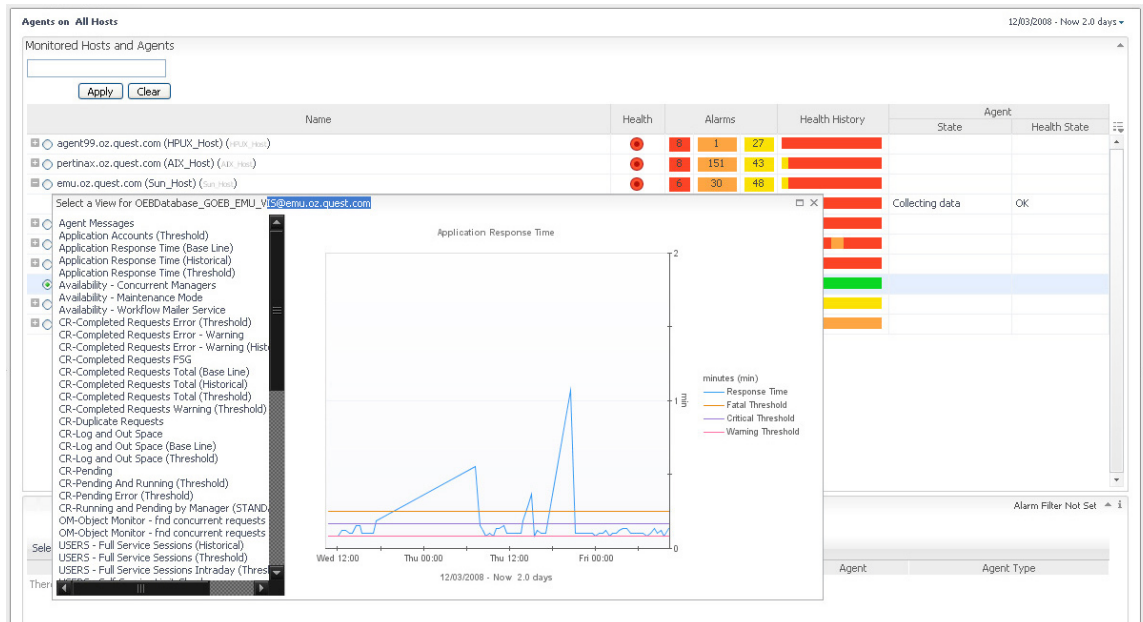


Figure 5.8: Including Oracle E-Business in your APM solution.

Most modern applications will have components written in either Microsoft’s .NET Framework or Java EE; you’ll want to be able to include details of those applications in your APM efforts, also. Figure 5.9 shows what Java APM might look like. This is a fairly detailed and deep view into the actual JVM and your Java-based components; you can see information on transactions, individual components, the JVM performance (including garbage collection), query execution, and so forth.

The message here is straightforward: Your APM solution needs to be able to natively recognize and deal with as many of your application’s components as possible in order to be maximally effective. That, in fact, should be your first and primary criteria when selecting solutions to evaluate.

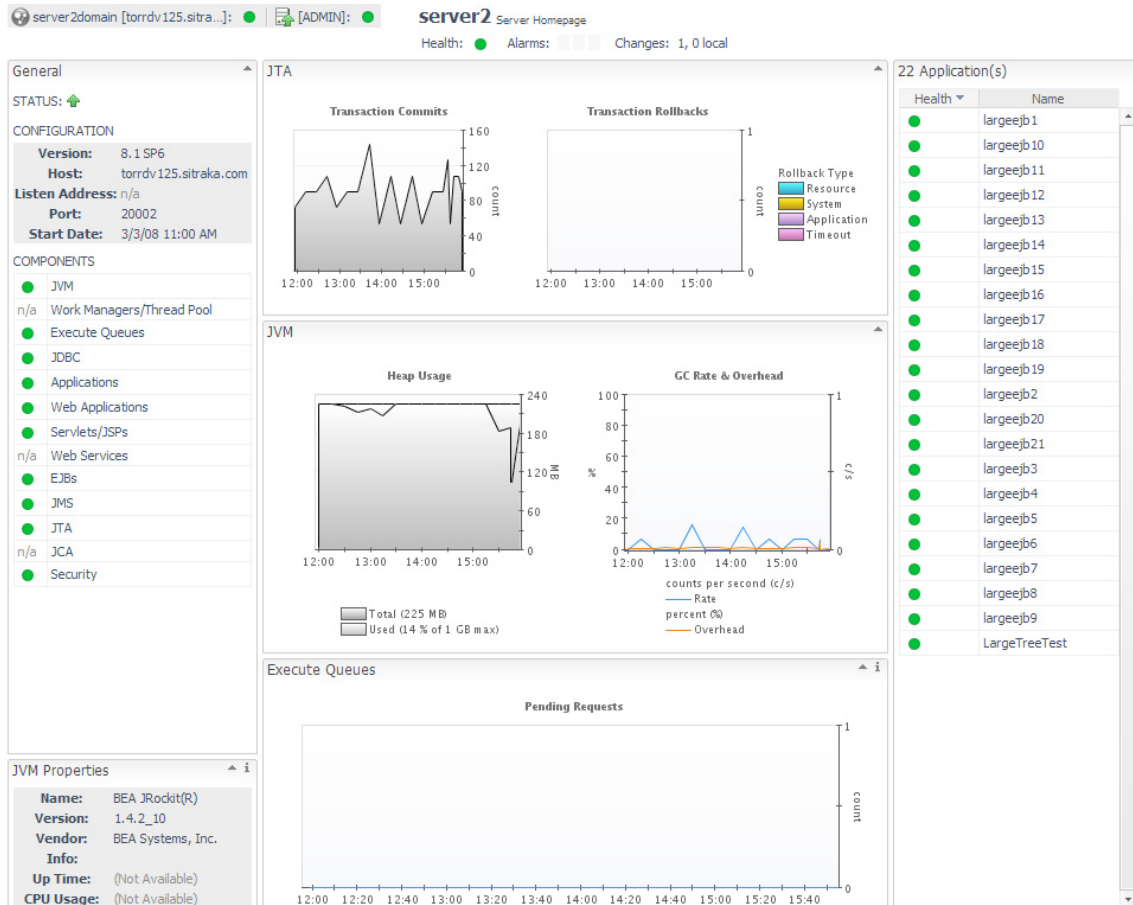


Figure 5.9: Java APM.

Holistic APM Requires 5D, Not 2 or 3 or 4

Gartner states that APM vendors have come to realize that “...all five dimensions are equally critical.” In other words, to have a truly robust and useful APM solution in place, it needs to implement *all five* of the dimensions that Gartner identified:

- EUE monitoring
- User-defined transaction profiling
- Application component discovery and modeling
- Application component deep-dive modeling
- APM database

Having a two-dimensional (2D), three-dimensional (3D), or four-dimensional (4D) solution simply doesn’t provide the capabilities needed. That’s not a fact being driven by vendors; Gartner points at that, “...it has been technology buyer insistence...that has led to the almost-universal adoption of the five-dimensional model for APM.” This is perhaps the most important point to remember as you begin evaluating potential APM solutions: Most experienced APM users know that you need all five dimensions to be successful.

If full coverage for all of your application components is your first selection criterion, here's your second: a complete 5D model. Independent experts tend to agree that only a complete APM solution implementing all five dimensions should be under consideration.

Note

It's unusual for vendors to offer five specific components that each map to one of the five dimensions. It's far more common for vendors to create integrated, modular toolsets that each provides different coverage for the dimensions. Ask your vendor to demonstrate how their product covers the five dimensions. You don't, for example, need a product that *just* provides EUE monitoring; some vendors may include that capability as part of a product that also handles user-defined transaction profiling and application modeling. Any mix is fine, so long as the *capabilities* outlined by the five dimensions are present.

Creating Your 5D APM Toolset Shopping List

I've already identified two key criteria, and these should be right at the top of your shopping list. First is full coverage for all, or as many of, your application's component technologies. Be sure to consider:

- Development frameworks
- Database servers
- Network infrastructure components
- Third-party packaged software or application servers
- OSs
- Virtualization hosts
- Clients—both thin and rich
- Web servers

Second is complete provisioning of all five dimensions of the 5D model, as I've outlined it in this book.

In addition to those criteria, each of the five dimensions has specific, more granular criteria that you might want to keep in mind. The importance of these will differ from organization to organization, so you'll need to filter as is appropriate for your circumstances.

Business Statistics

At a minimum, an APM solution needs to provide for either active or passive monitoring of end-user transactions; as a reminder, *active* monitoring is generally used to describe the injection of synthetic transactions into the system, whereas *passive* monitoring simply watches existing real-user transactions. *Both* techniques can be employed in complement.

The EUE monitoring should, again at a minimum, provide you with the ability to detect an application whose key end user response times are starting to get longer than you'd like (and ideally, whatever "you'd like" would be defined in an SLA, which the solution could help you manage directly).

However, the steps involved in monitoring the EUE can enable a powerful APM solution to deliver a lot more than just assurance that you're meeting your SLA. For example, the dashboard in Figure 5.10 shows how a geo-aware APM solution can help highlight geographic areas that contain more customers, that are not responding well (which is great in a highly-distributed application like a global Web site), and so on.

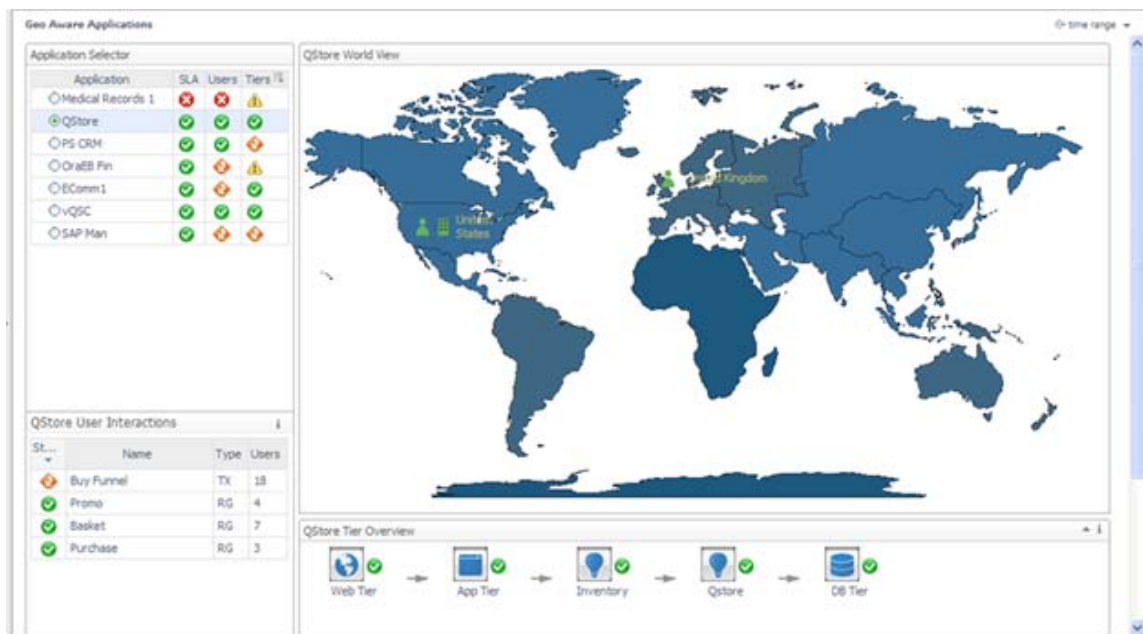


Figure 5.10: Geo-aware APM solution.

An APM solution can also provide business-level statistics. Figure 5.11 illustrates this, showing a "buy funnel" for a Web site. Because the APM solution is tracking real user transactions, it can show you how many users are in each step of the overall purchasing process, from adding an item to a cart to completing the order. Illustrating the conversion rate in this fashion is something normally relegated to a Web analytics package, but having access to it from within the APM solution can make conversion ratio more visible to the entire company—including executives.

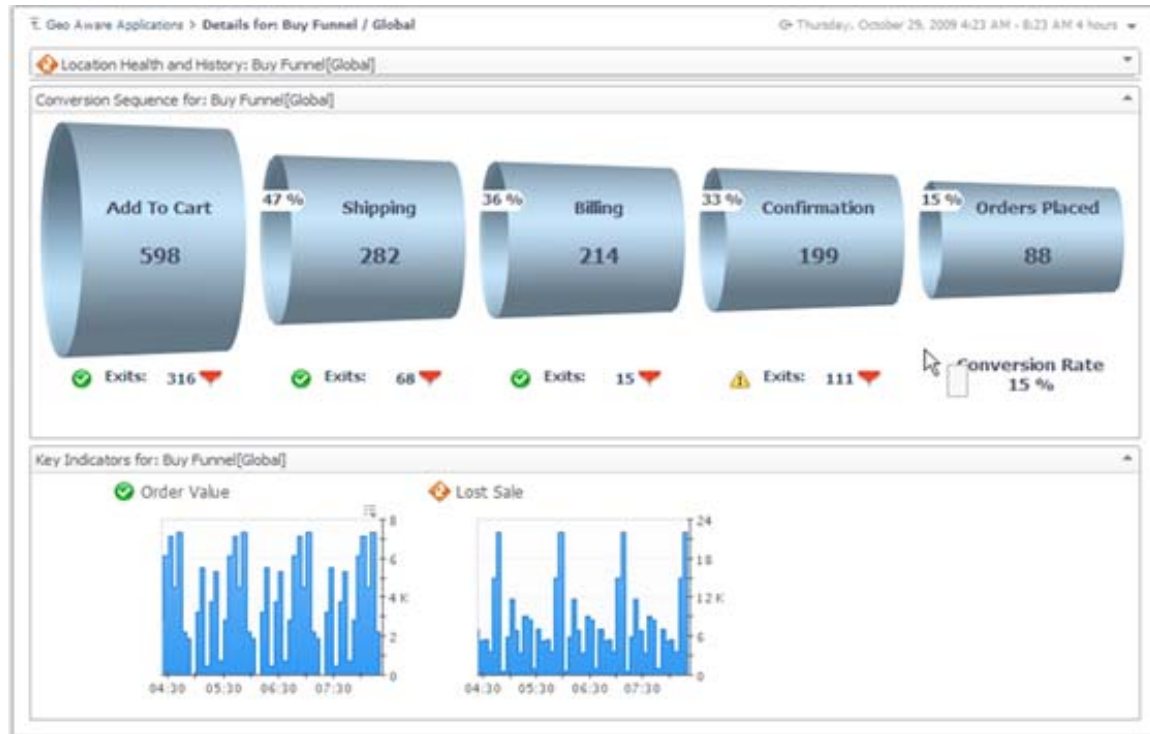


Figure 5.11: Tracking a buy funnel from an APM solution.

Based on this information, managers can begin trying to increase the conversion ratio, looking for reasons that users fall out of the funnel at various stages and looking for ways to keep them in the funnel through to completion.

Usability and Troubleshooting

It's often difficult to predict *exactly* what users will do with your application. A good APM solution, however, can *show* you, by capturing real-user transactions and behaviors, and then enabling you to “replay” those on your own. You'll have the experience of looking over the user's shoulder, enabling you to re-create exactly what the user was doing. That's an invaluable tool for troubleshooting and fine-tuning. Using that same data, you should even be able to search for specific user transactions—all the way down to HTTP requests and responses, as Figure 5.12 shows.

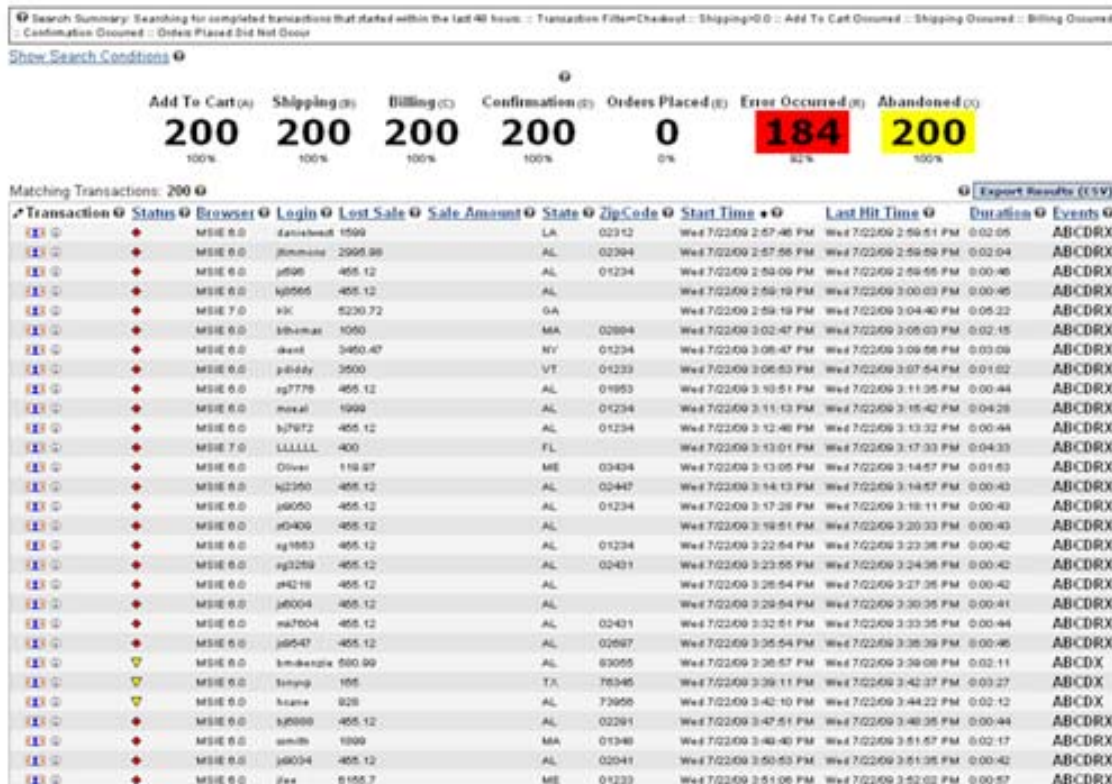


Figure 5.12: Searching for specific transactions.

SLA Management

A lot of APM solutions I’ve seen will let you manage the EUE, but they won’t necessarily help you align that to a specific SLA that your business has defined. A good APM solution should do so. You should be able to summarize all your IT components into a definition of an application that a customer or user would recognize: “the e-commerce Web site,” for example, or “the ticket-tracking application.” You then define SLAs on those user-centric applications, and monitor the quality of those services.

This approach helps to align APM to management-level practices, such as management frameworks (like ITIL), to Business Service Management (BSM) or IT Process Automation (ITPA) initiatives, and so forth.

This approach has a side benefit: If the monitoring system detects a performance problem in a deeply-nested technology component, it can surface that alarm all the way up to the SLA/application level. That lets you know not only which component to start fixing but also which users are potentially going to be impacted. You can then, in parallel with fixing the problem, attempt to mitigate end-user impact by communicating the outage to them, providing workarounds, and so forth.

Don't Forget the Infrastructure

I don't like seeing allegedly "all-inclusive" APM solutions that completely disregard the physical network infrastructure. Without switches, firewalls, routers, and other physical elements, applications don't work; it makes sense, then, to include those physical elements in the application's overall performance monitoring.

At a minimum, an APM solution should help you auto-discover network components, help you include those components in application models (so that the network components are properly described as dependencies of the application), and help you monitor the performance of those network components. If a network component fails or degrades, you should be alerted and be given at least a basic set of tools for troubleshooting the problem and engaging a domain expert to help solve the problem. Applications and SLAs potentially impacted by the problem component should also raise an alert so that you can begin considering workaround or mitigation plans.

Modern Applications, Modern APM

Our applications have been getting more and more complex for years. Yet in many ways, our ability to manage their performance hasn't kept up. It's really only recently that a good selection of fully-five-dimensional APM solutions have become widely available.

I find, however, that getting people's heads wrapped around 5D APM can be difficult. The bottom line is that for years, businesses would focus on very technical details like server utilization or network bandwidth as their primary performance metrics. They would have loved to be able to accurately and directly measure the actual EUE—but there was really no practical way to do so. After three decades of focusing on how the technology is performing at a component level, it's not very difficult to convince people to do something different. Those of us in IT just find the EUE to be a foreign, nigh-unreachable concept—but in reality, it's what we've always wanted. We're used to this:

"Help Desk. The application is slow? Well, the server seems to be performing okay. It's only at 50%. You don't know what that means? Well, I'll get someone to take a look."

We've been doing that for years. We're *driven* by the EUE—or at least some inconsistent, ad-hoc version of it—but we can't really validate it. We have to try to correlate that touchy-feely end user perception with the hard performance numbers we *do* have.

Now, with the right APM toolset, we can have this:

"Help Desk. The application is slow? Where are you located? Virginia. I'm checking. No, the application is performing well within its service level agreement from the Virginia office. We're monitoring transactions from other users there and they are being completed well within the service level agreement."

Put some numbers on the EUE. Monitor it. Turn that touchy-feely user perception into hard, scientific, unambiguous numbers. You'll be telling *them* when the application is slow, not the other way around.

Conclusion

There you have it: The five essential elements of APM. With the right tools, implementing all five dimensions that I've identified in this book, you can truly make APM a more consistent, useful, and powerful part of your IT environment. I should note that although a good APM solution will support all five elements, *you* don't necessarily need to tackle all five at once. It's entirely possible to utilize one or two elements on your first project, then integrate other elements as you learn how to use the system and modify your practices to integrate it.

With the right APM solution, you can begin to manage application performance primarily from the end-user perspective, and you can complete performance management and troubleshooting tasks more quickly and efficiently. With the right solution, you can practice effective APM across today's modern, distributed, multi-component applications. Of course, it all requires the right solution—and by now, you should have the details needed to identify the one that will work for your company.

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>.