Compliments of riverbed

2nd Riverbed Special Edition

Application Performance Management

FOR DUMMIES®

A Wiley Brand

Learn:

- Performance monitoring from the user's view
- How to monitor an application in a hybrid cloud environment
- To apply big data techniques to derive insights and to optimize application performance

Lawrence C. Miller, CISSP

Riverbed SteelCentral

Riverbed SteelCentral gives IT and business teams the end-to-end visibility and insights they need to ensure 24/7 application performance and to deliver superior digital experiences.

By continuously monitoring and blending insights gathered from all end-user devices (mobile, physical, or virtual), networks, infrastructure, and applications, regardless of the hosting environment, SteelCentral provides an integrated and complete Digital Experience Assurance solution. With SteelCentral, the business assures, continually improves, and preemptively troubleshoots issues that can impact digital user experience. You get

Complete visibility: Bring together the complete application picture: end-user experience, user devices, application, network, and infrastructure.

Real answers: Apply "big data" analytics to expose issues, to automate analysis and troubleshooting, and dramatically accelerate problem resolution.

Cross-functional insights: Present relevant information about application performance to stakeholders within IT and the business.

For more information about what Riverbed SteelCentral can do for you, visit **www.riverbed.com/steelcentral**.

Application Performance Management

FOR DUMMIES®

A Wiley Brand

2nd Riverbed Special Edition

by Lawrence C. Miller, CISSP

FOR DUMMIES®

A Wiley Brand

Application Performance Management For Dummies®,
2nd Riverbed Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-119-30379-4 (pbk); ISBN 978-1-119-30380-0 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

Project Editor: Carrie A. Burchfield

Editorial Manager: Rev Mengle

Business Development Representative: Karen Hattan

Special Help from Riverbed: Sue Cole, Heidi Gabrielson, Russ Elsner, Dimitri Vlachos, Rebecca Salie, Jon Hodgson, Golan Shem-Tov

Production Editor: Antony Sami

Table of Contents

Introduction

Application Performance Management For Dummies, 2nd Riverbed Special Edition, introduces you to application performance management (APM) solutions and how these tools can help you monitor and troubleshoot your mission-critical applications — from the perspective of your users, as well as your systems.

About This Book

This book contains volumes of information that rival a map of the human genome, conveniently distilled into six short chapters chock-full of just the information you need! Each chapter is individually wrapped (but not packaged for individual sale) and written to stand on its own, so feel free to start reading anywhere and hop, skip, or jump between chapters (or around your office)! Here's a brief look at what awaits you.

Chapter 1: Understanding What Application Performance Management Is. I begin by exploring some application trends and different elements of APM, including monitoring and troubleshooting.

Chapter 2: From the Driver's Seat: Digital Experience Monitoring. This chapter takes a look at application performance from the user's perspective.

Chapter 3: Looking Under the Hood: Transaction Tracing. Here, you take an in-depth look at how application transaction tracing accelerates troubleshooting, and how APM helps bridge the workflow between application support and development.

Chapter 4: Horsepower and Fuel Efficiency. In this chapter, you learn to use a "big data approach" to deal with all of the data that APM collects.

Chapter 5: Exploring the Application Performance Management Terrain. Here, you learn about some creative uses for APM to help improve efficiency and effectiveness in your organization.

Chapter 6: Application Performance Management Evaluation Criteria. Here, in classic *For Dummies* style, I tell you about several important criteria to look for in an APM solution.

Icons Used in This Book

Throughout this book, you sometimes see icons that point out important information. Here's what to expect.

This icon points out information that may well be worth committing to your nonvolatile memory — along with anniversaries and birthdays!

If you're an insufferable insomniac or vying to be the life of a World of Warcraft party, take note. This icon explains the jargon beneath the jargon.

Thank you for reading, hope you enjoy the book, please take care of your writers! Seriously, this icon points out helpful nuggets of information.

Chapter 1

Understanding What Application Performance Management Is

In This Chapter

- Recognizing trends and challenges
- Looking at different APM components
- Bringing it all together in a complete solution

Today's complex business applications present new challenges that legacy application monitoring tools are ill-equipped to address. This chapter talks about these challenges and the important elements that make up a complete and effective application performance management (APM) solution.

Complex Application and Infrastructure Landscape

Increasingly, application performance directly impacts business performance. Users rely on applications to reach customers, build and sell products, provide

services, automate business processes, and perform almost every other task critical to the business. And as applications have become more critical, they have also become more complex. For most teams, application performance and availability are the most visible indicators of their success.

The demand for newer, more powerful business applications at an ever-increasing rate has led to the adoption of rapid software development methodologies that increase the speed of delivery for new applications. At the same time, the introduction of new infrastructure technologies has increased the complexity of the underlying infrastructure over which applications run (see Figure 1-1). Some examples of these trends include

- **Agile development and DevOps:** Agile development is a conceptual software development framework in which cross-functional teams share knowledge, ideas, opinions, and experiences to identify evolving requirements and solutions. Team tasks are divided into short timebox iterations that typically last only a few weeks and cover all development functions including planning, requirements analysis, design, coding, and unit and acceptance testing. DevOps is a rapid software test and deployment method that emphasizes communication and collaboration between development and IT operations teams.
- **Microservices:** A specialization of Service Oriented Architecture (SOA) used to build an application as a suite of small, independent, functional modules. These modules are easily replaceable, they run as autonomous processes,

they're elastic and transient, their location changes, and they communicate with each other over a network by using a lightweight mechanism (typically HTTP).

- **Virtualization, cloud, and containers:** Virtualization, cloud, and container technologies provide flexibility, rapid provisioning, operational efficiency, increased density, and many other benefits, but they also introduce new server, storage, and network infrastructure complexities.
- **Edge computing:** Multi-tiered and highly integrated architectures mean that a portion of the application runs at the "edge" — closest to the user. Of course, when an application has an issue, it's the experience at the edge that matters most to the user. But troubleshooting an issue in such a complex environment requires visibility everywhere that the application and its individual components run, including dependent systems, applications, and services.
- **Mobility:** In traditional networks, business applications run entirely on systems in the corporate network, thus DevOps teams can have complete end-to-end visibility into these applications and their performance. Users are now accessing systems and applications beyond the network — from anywhere in the world, at any time, and on multiple types of devices.

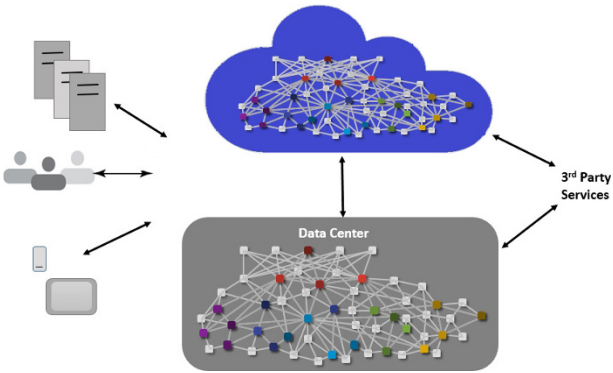

Figure 1-1: The complex world of the modern application.

In the face of these trends, DevOps teams must address challenges that include

- **Troubleshooting and recovering from business disrupting application issues:** Downtime and poor performance of critical applications have a direct impact on business. When application issues occur, DevOps teams must detect, isolate, and fix the problem quickly. But with legacy performance management tools, it can take days — if not weeks — to isolate and fix performance problems, and problems are often only discovered after end-users report an issue.
- **Failing to find and fix recurring or persistent performance problems:** Intermittent and chronic performance problems negatively impact end-user satisfaction and productivity and prevent DevOps teams from focusing on new initiatives. While

internal users may become numb to these chronic problems, external users typically have a larger selection of vendors, sites, and applications and can just switch to the competition. For both user types, this can have a significant impact on business.

- **Preventing future disruptions due to an inability to detect problems early:** DevOps teams usually find out about performance problems from end-users (according to Gartner, up to 70 percent of the time), after the business has already been impacted. DevOps teams need to detect problems proactively so that they are working on isolating and fixing the problem before the business is impacted.
- **Ensuring that IT initiatives don't cause new application performance issues:** When businesses virtualize, containerize, consolidate, or migrate their data centers to the cloud, they expect to improve flexibility, cost, and control. They don't expect to negatively impact application performance. When rolling out new applications or expanding existing deployments, it's critical to ensure that the performance required by the business will be delivered. Without the right tools to manage and predict the effects of such infrastructure or application changes on application performance, however, businesses often find themselves dealing with unforeseen performance problems.
- **Reporting pertinent information to executives and other key groups:** Application performance isn't a single group's responsibility and has broad implications across IT operations, application teams, and business owners. It's imperative in this

environment of heightened application focus that the team is able to communicate about application performance broadly and in languages tailored to technical and business audiences.

These challenges result in significant negative impacts to businesses, including loss of revenue and productivity, customer dissatisfaction, and frustration.

Different Elements of APM

IT modernization is a top driver in cloud adoption. The "cloud first" mantra is heavily used. IT organizations are also changing the way they manage their application and infrastructure — according to Gartner, by 2020 70 percent of APM suite technology buyers will reside outside of traditional IT operations organizations. Today's complex application and infrastructure landscape requires an APM solution that looks across all critical parts of the application delivery chain. Whether you are rolling out new applications, containerizing their application, consolidating or virtualizing data centers, or migrating to the cloud, managing application performance requires a holistic view that includes

- **Digital experience monitoring:** The availability and performance that digital agents (humans or machines) are actually experiencing as they interact with the application. Whether using web- or thick-client applications, and regardless of whether they are local, around the world, or use a mobile device, you must be able to monitor and troubleshoot the ultimate measure of application performance: the digital experience.

- **Application discovery, tracing, and diagnostics:** What a digital agent (human or machine) perceives as a single operation involves many distinct sub transactions across many different application components, such as different application servers or databases. How do you track, monitor, and troubleshoot the performance of each of these critical transactions, correlate them, and deep dive into the various components of the application?
- **Application analytics:** Automatic detection of the source of transaction performance problems using statistical analysis, machine learning, pattern recognition, or mathematical theories. This is the heart of an APM tool in modern and complex environments where applications frequently change, are built with microservices, and are elastic, transient, and highly virtualized.
- **Infrastructure and network performance management:** How is the performance and availability of the underlying IT infrastructure and network contributing to the performance of the application? IT infrastructure exists for one reason: to deliver applications. To really understand holistic application performance, infrastructure must be managed through the lens of the application. In particular, the network is a critical component whose importance continues to grow with cloud, mobile, SDN (software-defined networking), and network virtualization trends.

Although transaction tracing in APM tends to refer to code-level tracing within the servers, the network provides another critical perspective using packet captures to analyze network

traffic. Both approaches have value and should be part of your solution.

SDN is an emerging approach to building computer networks that simplifies networking by replacing many of the functions of traditional network equipment (such as firewalls, routers, and switches) with software components.

Monitoring and Troubleshooting Applications

An effective approach to APM must integrate all APM elements. The complex, rapidly changing, and business-critical nature of modern applications demand it. A comprehensive APM solution enables DevOps teams to

- **Rapidly diagnose the root cause of performance problems.**
  - Identify problems fast by seeing all components of application delivery including IT infrastructure and the network.
  - Identify performance problems sooner and avoid negative impacts to the business with advanced analytics that detect, isolate, and pinpoint performance problems.
  - Eliminate the "war room" that takes IT resources away from more strategic projects and reduce or eliminate "finger pointing."
- **Dramatically improve efficiency.**
  - Streamline communication among teams (faster development life cycles, lower support costs, and fewer project failures).

- Leverage dashboards and reporting that present relevant information to business, IT, and DevOps stakeholders.
- Understand how applications are performing with easy-to-digest performance views.

Proactively manage performance problems.

- Discover the landscape of application component dependencies across all IT assets to perform impact analysis and to ensure all critical parts of the application delivery chain are monitored.
- Enable faster, more accurate planning to minimize the impact of IT change initiatives.

APM and ADC: Diagnosis and cure

Imagine going to your doctor because you have a sore throat. Your doctor sticks a tongue depressor in your mouth, tells you to say "ahhhh," confirms that you have a sore throat, and sends you on your way!

This example is much like legacy application monitoring tools today. These tools provide a very focused, but limited, view of the overall problem. To effectively diagnose a problem in today's complex application and infrastructure landscape, you need to see the complete picture.

Now, returning to the doctor's office scenario, you're more likely to have a nurse first ask you about your symptoms, medical history, and any medications that you're taking, then check your blood pressure, temperature, and pulse. The doctor then gives you a complete examination, checking not

(continued)

(continued)

only your throat but also your breathing, heartbeat, and ears, before making a diagnosis and prescribing an antibiotic.

This is more analogous to a complete APM solution — except for the prescription. If APM provides the diagnosis, then Application Delivery Controllers (ADCs), as well as WAN accelerators, provide the cure. Going a step further with this analogy, ADCs are also an integral part of your wellness program for preventative care. Thus, APM and ADCs work together to keep your applications environment healthy!

Chapter 2

From the Driver's Seat: Digital Experience Monitoring

In This Chapter

- Getting to know your users
- Measuring performance
- Monitoring availability with synthetic tools

Digital experience monitoring (DEM) is an availability and performance monitoring discipline that supports optimization of the operational experience and behavior of a digital agent (whether human or machine) as it interacts with enterprise applications and services. This process includes real-user monitoring (RUM) and synthetic transaction monitoring (STM) for both web- and mobile-based end-users.

In this chapter, you explore application performance management (APM) tools that provide insight into application performance from the perspective of your users. In other words, the view from the driver's seat!

14

Understanding User Behaviors and Patterns

In the not-too-distant past, availability (whether "up" or "down") was the key measure of performance of networks, systems, and applications for IT teams. And while availability still matters, networking and systems hardware fail less often, and critical infrastructure now regularly incorporates redundant components and designs for high availability. Modern infrastructure is also built to scale, and in most cases, it's elastic, and the number of components is growing and shrinking according to demand. Most times those components are transient and move according to geographical change in workload.

Today, users expect more. Not only do applications have to be available where and when they're needed — and on all devices — but also they must be responsive and consistently deliver a fast experience. Users and business units judge applications by their interactions and experiences with those applications. While errors and downtime still cause issues, speed or more precisely, lack of speed, measured in response time can be a major source of frustration for your users. Think about it, when was the last time your users complained that their applications were too fast?!

Performance has historically been measured at an individual component or system level, such as a network segment, a database, or an application server. The assumption was that if all the "parts" were performing as expected, then the user's experience must be fine.

But the sum of the individual parts often did not accurately represent the whole experience. Individual back-end measures, such as bandwidth and server utilization, provide a myopic view of application performance. What really matters is the digital experience, the availability, latency, execution correctness, and quality as they appear to a digital agent (human or machine). (See Figure 2-1).

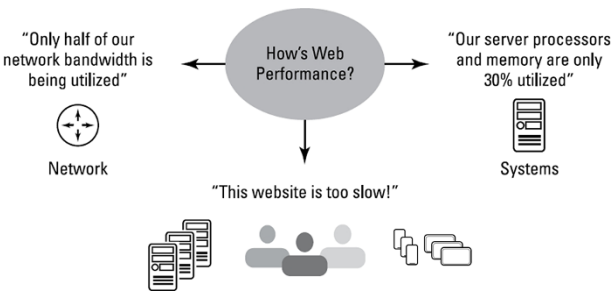

Figure 2-1: Different perspectives on performance.

DEM also contributes to a broader understanding of user behaviors and patterns with respect to applications. By monitoring real user interactions with applications, DevOps teams see how their users *actually* use an application, rather than how they are *expected* to use an application. This understanding not only improves troubleshooting and problem resolution, but also it leads to business process improvements! Thus, DevOps teams can transform themselves from a user and application support function that is reactive in nature, into a more proactive role that delivers added value to the business.

DEM

Analysis of digital experience data enables DevOps teams to observe and analyze the interactions between users and application. It gives the teams the ability to break down application response times into contributing sources, launch troubleshooting of root causes, and analyze traffic and response times among servers to manage multi-tier applications. DEM in application performance monitoring takes two different approaches: Real-user monitoring (RUM) and synthetic transaction monitoring (STM).

RUM

RUM observes and records performance data about actual live user transactions within applications to determine whether these applications are meeting the users' performance expectations, as defined in service-level agreements (SLAs).

Effective RUM measures an application's responsiveness the same way that your users experience it, thus requiring a holistic view of the application. For example, web-transaction response time should be measured at the page level rather than being based on a single HTTP request/response performance measurement.

To monitor real user experience, either an agentless, appliance-based (physical or virtual) solution is installed, or agents are deployed throughout the application infrastructure. JavaScript-based user experience measurements, both RUM and STM (discussed in the next section), within web browsers can also feed performance data about web-based applications to a management server.

A passive appliance-based DEM solution can provide additional capabilities for DevOps teams, including the ability to

- Visualize communication within the data center and across the WAN
- Measure traffic by application, user, business division, and location
- Analyze historical information for trending and capacity planning
- Measure performance of voice-over-IP (VoIP) calls while they are in progress

RUM enables DevOps teams to

- Measure application performance for internal and external users of your enterprise applications
- Monitor live performance for all users at all times, and alert against SLAs

JavaScript-based agents are a great solution for DEM of cloud-based applications, where deploying an appliance may not be an option.

STM

STM augments RUM. With an understanding of your users' behaviors and patterns (explained earlier in this chapter), you can create a series of process steps within an application that run at a continuous or timed interval.

STM provides valuable information about an application's response time and availability.

STM is only as good as its steps are accurate and relevant. You must accurately replicate the aggregate behavior of your users, and the steps being monitored must indicate the availability of the application.

Chapter 3

Looking Under the Hood: Transaction Tracing

In This Chapter

- Triaging application issues with APM
- Decoding application performance issues
- Collecting and analyzing transaction data

What is perceived as a single application operation actually involves many distinct application transactions across many different application components including the backend database, web server, middleware, auditing servers, and backend third-party services.

How do you monitor and troubleshoot the performance of every component of multi-tier applications and the transactions that traverse them? Roll up your sleeves and find out how in this chapter!

Following Application Transactions Across All Components

Today's complex multi-tier business applications introduce new challenges for effectively triaging application-performance issues. These challenges include

- Monitoring applications across all tiers, hosts, nodes, and domains, and understanding how transactions traverse application components
- Locating problems during development, quality assurance, and in production, and rapidly pinpointing the source of performance issues at the component/code level

Transactions are a unit of work performed by an application. Most applications support many different transactions related to a common business area and depend on a common set of data. Each transaction is designed to accomplish a particular business purpose and exposes parameters to specify the particularities of the transaction each time it is requested by a user.

Monitoring Performance at the Code Level at Every Step

Incomplete data leads to incorrect answers. Therefore, an APM solution needs to collect data about the

performance of application components for every user, every transaction, all the time, and at a very granular level of detail. For a complete picture, an APM solution needs to monitor performance metrics that cover all application components such as Java, .NET, web servers, databases, operating systems, and storage.

APM solutions need to provide the greatest visibility with the lowest overhead, which can only be accomplished with automated intelligence. It is impractical to think that DevOps teams can manually determine what individual components should be instrumented.

Tracing Transactions

Application code-level transaction tracing bridges the workflow between application support and development, enabling cross-functional teams to identify and troubleshoot performance issues from the specific transaction step to the affected code. Examples of developer pain points related to application performance include

- Before deployment
  - Performance results are too hard to get while performing quality assurance (QA) on code.
  - Legacy code isn't instrumented (see Chapter 4 to learn more about instrumentation).
  - Performance test results aren't actionable.
  - Performance bottlenecks take days to find — sifting through load test results is difficult.

- In production
  - Code is too often "guilty until proven innocent," and developers have little or no access to production.
  - Issues can't be diagnosed to a level that tells developers what code needs to be fixed.
  - Production problems can't be easily or accurately reproduced.
  - Architectures are becoming more complex, so finding problems becomes more difficult.
  - Hosts and nodes aren't constant and infrastructure is elastic and transient, which makes correlating transaction performance issues with resource problems very difficult.

To address these pain points, transaction tracing enables application developers to see exactly what happened in any transaction across all tiers to

- Determine what application components come into play as the transaction traverses them
- Determine what hosts were involved in processing the transaction and what was their resource utilization at the time
- Understand the impact of individual instruction sets on transaction performance
- Identify any external system dependencies that may exist
- Obtain visibility into the execution of third-party code and components

Chapter 4

Horsepower and Fuel Efficiency

In This Chapter

- Collecting performance and forensic data
- Going big with big data techniques
- Automating analysis to detect performance problems

In this chapter, it's time to understand what powers APM — which is data. Here, you learn how application performance data is collected and how to get the most mileage out of this data!

Instrumentation: Keeping It Light

Instrumentation is a challenge for many APM solutions, particularly for multi-tiered applications with many application components or build with microservices. Other APM instrumentation challenges include mobility, virtualization, containerization, and cloud computing trends.

Instrumentation provides data feeds that enable an APM solution to determine the state of an application. Instrumentation types include

- Java or .NET code
- Installed agents
- Passive monitoring (agentless) via network-based, packet inspection and analysis

Different deployment scenarios for APM instrumentation may include

- Agents on desktops and servers to continuously collect detailed application and systems data
- High-speed, passive appliances to measure end-user response time at key locations
- Lightweight JavaScript automatically downloaded to browsers to capture end-user experience for web-based applications
- High-speed passive appliances to monitor application traffic within the data center, across both physical and virtual infrastructure and servers
- Low-cost virtual appliances to measure user response time at branch offices and remote locations

Dealing with Lots of Data — Every Transaction Matters

APM solutions capture copious amounts of data, including performance data and forensic data.

Performance data consists of measurements that are captured and recorded over time, such as CPU utilization, or the time it takes for a database query to execute. Performance data is monitored for problem detection, and archived to provide historical views and analysis of trends. Anomalies that indicate a problem can be obscured by traditional data summarization techniques or missed completely by collection intervals that are too infrequent. An effective APM solution captures very fine-grained performance information and preserves its fidelity.

Forensic data goes beyond quantitative metrics and is obtained by recording activity, such as a transaction. Its purpose is to provide evidence or clues about a problem or operation (hence the term "forensics") in the retrospective analysis of an event.

Forensic information is generally obtained from lower-level techniques, such as code-level tracing, detailed resource tracking (for example, memory consumption), and deep packet inspection of network traffic. The data is used to drill down deep into the behavior of the application, determine how components are interacting with each other, and break down all activity related to an individual transaction or other events of interest. Rich forensic information is fundamental to root cause isolation.

Applying Big Data Techniques

Big data techniques offer deep insight into application dynamics, thereby speeding problem diagnosis, increasing the DevOps team's efficiency, and improving application service levels.

The full potential of APM lies in exploiting large volumes of detailed data about the performance and behavior of all transactions. Big data capabilities go far beyond traditional statistical performance summaries by enabling massive retention of transactional data. These capabilities also enable powerful unstructured searches that give IT the ability to quickly detect patterns, and to find and analyze the specific transactions that definitively answer questions. Equipped with these capabilities, DevOps teams can rapidly pinpoint the root causes of performance problems and minimize the impact on end-users and the business.

The scale of data required to comprehensively manage application performance quickly overwhelms traditional performance management solutions. Virtualized, containerized, dynamic infrastructures and highly distributed application architectures further contribute to the exploding volume of data. Analytics are essential to pinpoint the specific data that is relevant to a problem, or to indicate an emerging problem. Historically, performance management solutions have dealt with this volume of data by sampling, averaging, or otherwise reducing the granularity of data they collect. These techniques leave DevOps teams with inadequate data and insight, losing the ability to track the complete application performance picture for every user, every transaction, all the time.

In contrast, big data techniques enable application support teams, developers, and operations to mine massive amounts of high-fidelity data to extract meaningful information. The result is faster troubleshooting, increased uptime, and the opportunity to proactively identify application issues before they impact business.

Leveraging Analytics to Detect Patterns and Problems

Analytics refers to a set of automated analyses that process and leverage performance and forensic data for performance management purposes, including monitoring, alerting, and troubleshooting. Once data is gathered, analytics operate on the data to provide functions such as

- Adaptive baselining to establish "normal" behavior and uncover "abnormal" behavior to support dynamic thresholding and alerting
- Intelligent information storage for rapid retrieval during problem investigation or performance analysis
- Identification of metrics that are deviating from normal, based on complex variables such as time, day/week/month, location, and user community, among others
- Correlation of performance anomalies to uncover cause-and-effect relationships, and patterns of deviation that occur together to determine if they are related and what they have in common (for example, shared code used throughout an organization) to provide a starting point for forensic analysis
- Derivation of new information to answer specific questions, such as long-term trends, peak operating periods, and top consumers of resources

- Automation of complex administrative tasks, such as selectively instrumenting application components to maximize visibility while minimizing overhead

Gartner has updated its definition of application performance monitoring suites and acknowledged the importance of application analytics (AA) to APM. AA is listed as one of the main functional dimensions in an APM suite.

Chapter 5

Exploring the Application Performance Management Terrain

In This Chapter

- Ensuring satisfactory service levels
- Diagnosing and assigning responsibility quickly
- Troubleshooting performance issues methodically
- Understanding complex application dependencies
- Keeping an eye on applications in the cloud
- Managing changes proactively

Application performance management (APM) gives DevOps teams visibility and actionable insight to deliver the application performance that users and businesses demand. In this chapter, you learn how APM helps DevOps teams manage service-level agreements (SLAs), perform triage quickly and effectively, troubleshoot problems, discover and map application dependencies, and monitor applications in the cloud and through major infrastructure changes such as microservices, containerization, virtualization, and consolidation.

30

Monitor Application Performance and SLA Compliance

APM enables DevOps teams to monitor SLA compliance beyond simple uptime measures. Although "five nines" (99.999 percent) availability remains an important SLA metric for service providers, end-to-end responsiveness has quickly become a more relevant bellwether of service levels for end-users.

Although individual components in an application infrastructure may be up and running within established performance thresholds, the digital experience may be slow or unresponsive, making the application or website appear, in effect, unavailable or "down." For example, transactions of a website application will typically rely on the efficient operation of multiple steps and components such as:

- Web servers and services
- Application servers and services
- Database servers
- Network connections
- Routers and switches
- Firewalls and load balancers
- DNS (Domain Name Service) lookups

A failure or delay in any one of these steps or components results in an inability to use the application. Additionally, although each of the individual steps or components may be performing as expected, the resulting performance may be unsatisfactory. Therefore, an APM solution must monitor the overall digital experience.

Perform Triage

When a critical business application is down or performing poorly, teams often lose valuable time pointing fingers at other teams (such as development, networking, and systems) or third-party vendors. Although such "blamestorming" may seem petty, it is often actually the result of logical, albeit flawed, deductive reasoning based on incomplete information from one-dimensional monitoring tools. The systems team sees that all servers are up and running optimally; the networking team reports that bandwidth utilization is nowhere near peak and latency is low; and the development team says that the custom application worked fine yesterday and no recent code changes have been introduced.

A comprehensive APM solution provides the entire team with a "big picture" view of the entire application and infrastructure landscape from end-to-end, including all transactions and components. DevOps teams can then drill down into the details of the problem and quickly triage and assign the problem to the appropriate team or third-party vendor.

Troubleshoot Application Performance

Troubleshooting an application performance issue is often a tedious and frustrating process that can take days or even weeks to resolve. The troubleshooting process is often further complicated by a lack of meaningful information about the problem and limited tools for diagnosis and analysis.

A Forrester study found that only 25 percent of availability and performance problems is satisfactorily diagnosed and resolved within 24 hours. This represents a huge delta between reality and the SLAs by which organizations are measured (discussed earlier in this chapter), which often requires resolution of a critical problem within two hours. It is also a costly delta: The same Forrester study reported that more than 50 percent of respondents say the business cost of one hour of brownout (slow responsiveness) or downtime of their most critical application would cost from $100,000 to more than $1 million.

APM provides DevOps teams with the tools needed to effectively and methodically troubleshoot vexing performance problems, beginning with a "big picture" top-down approach that enables them to pinpoint and then dive deep into the individual transactions and components to isolate and resolve the root cause of performance issues.

Map Application Dependencies

Mapping application dependencies is an important process, particularly for complex, multi-tier applications. Such applications often have unexpected dependencies that even the application owners themselves may not understand. An incomplete understanding of business-critical applications makes troubleshooting difficult and increases the risk associated with application and infrastructure changes, such as cloud migrations, virtualization, and containerization (discussed later in this chapter).

APM provides the capability to map an application to the application components and underlying infrastructure elements that deliver it. This understanding of the application architecture enables more effective troubleshooting and helps to reduce unknown risks associated with application and infrastructure changes.

Manage Application Performance in the Cloud

Cloud computing provides opportunities for greater operational efficiency. The on-demand computing capabilities of a cloud can be more flexible and cost-effective — but several application management challenges aren't so easily vaporized.

Just because you've moved your applications to a cloud doesn't mean you don't have to keep your customers happy and your vendors honest! APM challenges in the cloud include

- Measuring digital experience when it may not be possible to instrument in traditional ways, such as via a network-based appliance
- Monitoring application components that are executing in the cloud
- Troubleshooting performance issues when the extended IT team includes the cloud provider
- Dealing with the complexity that is characteristic of hybrid environments that encompass multiple cloud vendors, as well as in-house infrastructure and applications

While cloud computing holds significant promise and claims to address many of the traditional challenges of application performance management, in reality, an organization's transition to the cloud will be gradual. There will always be a mix of application deployment models, and it would be wise to employ a management system that can handle them all. Important characteristics of such an APM solution include

- Measures the key aspects of application performance, including digital experience.
- Provides a unified view in a hybrid environment comprising multiple cloud services and data center applications.
- Integrates application knowledge from multiple perspectives — including application components, network, server, database, and web services.
- Offers a holistic approach that treats the application as more than a sum of its parts, when possible. ***Note:*** In a SaaS environment, it may not be possible to get an end-to-end view of the application.
- Includes application analytics.
- Provides appliance- and agent-based instrumentation options for a cost-effective approach to monitoring end-user experience.
- Fosters collaboration among IT teams and with cloud vendors.

Manage Performance with Application and Infrastructure Change

As applications and infrastructures change, critical business applications must be monitored to ensure that these changes don't unexpectedly impact performance.

Server virtualization and consolidation are key trends that are revolutionizing the modern data center. In many organizations, virtualization and consolidation initiatives are primarily driven by financial considerations that include reducing management and administrative costs.

Transitioning applications to microservices and the usage of containers are adopted rapidly and change the way application teams work. Using small functional modules that are transient, elastic, replaceable, and constantly changing offers a lot of power, flexibility, and cost savings that are vital to modern applications.

However, application performance is often an afterthought. This can be a costly mistake as these application and infrastructure changes inherently reduce visibility and increases application complexity. To proactively address these performance issues, these initiatives should include implementation of comprehensive APM capabilities. Without adequate visibility, these types of projects may be significantly delayed, eliminating or reducing potential cost savings.

Chapter 6

Application Performance Management Evaluation Criteria

In This Chapter

- Looking at digital experience monitoring
- Checking transaction tracing
- Providing application discovery and mapping
- Detecting application analytics
- Discovering dashboards and data visualization

This chapter reviews several of the important capabilities of APM and provides convenient checklists to help you evaluate different APM solutions.

Digital Experience Monitoring (DEM)

Digital agents (human or machine) expect instant access and a consistent experience with the business

applications they need. You must be able to monitor and troubleshoot the ultimate measure of application performance: the digital experience. Use Table 6-1 to compare APM solutions.

Table 6-1 DEM Checklist

Criteria	*Yes*	*Partial*	*No*
Monitors live performance of real users continuously			
Measures web transaction response at the page level, the same way users see it			
Establishes a baseline and generates alerts for poor digital experience			
Provides complete visibility into digital agent geography, platforms, and usage trends			
Determines the share of end-to-end delay that is attributable to each of the major components (for example, network, server, client, and others)			
Monitors the availability of applications even when there is no user traffic, using synthetic transactions			
Provides visibility in digital experience for specific locations, applications, and other business-oriented criteria			

Transaction Tracing

Code-level transaction tracing and application component performance monitoring enable you to see deep inside your applications to diagnose the root causes of performance problems. Table 6-2 lists several important criteria.

Table 6-2 Transaction Tracing Checklist

Criteria	*Yes*	*Partial*	*No*
Traces user transactions across all application tiers			
Records and indexes all transactions, for all users, all the time, not just samples or averages			
Monitors performance data at a fine-grained level to ensure that anomalies are captured			
Supports multiple application environments, including Java, .NET, popular third-party software, and custom applications			
Monitors system and application metrics			
Has very low operating overhead			
Configuration is simple/automated			
Uses a highly-scalable, big data approach to analyze transactions			

Application Discovery and Mapping

An application dependency map provides a complete picture of your application and infrastructure components, and their relationships. See what's important in Table 6-3.

Table 6-3 Application Discovery and Mapping Checklist

Criteria	*Yes*	*Partial*	*No*
Produces an accurate map of application dependencies at the time of execution			
Maps dynamic operating environments, such as microservices, containers, and virtualized servers			
Leverages auto-discovery and does not require manual data input or maintenance			
Finds and maps all applications, including custom developed applications			

Applications Analytics

Analytics monitor system and application metrics to automatically detect anomalous behavior, correlate information, identify the root cause of problems, and predict events and performance trends. Refer to Table 6-4 to assess APM strengths in analytics.

Table 6-4 Analytics Checklist

Criteria	*Yes*	*Partial*	*No*
Detects and correlates performance anomalies			
Captures and indexes every user transaction			
Extracts answers from vast amounts of performance and forensic data			
Utilizes a natural search language capability to quickly pinpoint meaningful information			

Dashboards

Dashboards provide a big picture view of your applications and infrastructure in a "single pane of glass" that provides different teams with actionable alerts and information. See what's important in Table 6-5.

Table 6-5 Dashboards Checklist

Criteria	*Yes*	*Partial*	*No*
Allows fully customizable views and role-based access controls for individual users or user groups			
Provides high-level views and alerts for fast triage and diagnosis			
Facilitates root cause analysis with seamless drill-downs			
Enables creation of new dashboards using drag-and-drop workflows			

Notes

Notes

Full-Stack Performance Monitoring On and Off the Cloud

Build and deliver high-performing applications, infrastructure and networks.

Continuously monitor them to get end-to-end visibility and end-user experience insights around-the-clock.

Try for free at **SteelCentral.net**

Application performance *is* business performance

Assuring 24/7 application performance is critical to delivering the digital experiences that users and businesses rely on. To do so, organizations need end-to-end visibility to understand user satisfaction and to troubleshoot problems. In this book, you will:

- ***Understand digital experience monitoring* — and why it's important**
- ***Apply big data techniques and analytics* — to derive insights and to quickly expose and resolve problems**
- ***Evaluate performance management criteria* — and choose the best solution to meet your needs**

Open the book and find:

- **How moving to the cloud and a containers-based environment affect APM**
- **Why digital experience monitoring is a key performance metric**
- **How to solve the complexity of a microservices-based architecture**
- **Why tracing all transactions for all users all the time is so important**

Making Everything Easier!™

Go to Dummies.com®
for videos, step-by-step examples, how-to articles, or to shop!

FOR DUMMIES®
A Wiley Brand

ISBN: 978-1-119-30379-4
Not for resale

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.