The Definitive Guide to Application Performance Monitoring in the Age of AIOps

Sponsored by

BROADCOM®

Table of Contents

INTRODUCTION

Application Performance Monitoring

The market surrounding Application Performance Monitoring, or APM, represents something of a paradox: On the one hand, it is one of the most well-established ecosystems within the modern IT landscape. On the other hand, it is among the fastest-changing spaces in the IT industry today.

Indeed, APM tools, which debuted in the late 1990s, have been around longer than many of the other technologies that power modern IT workloads, such as virtualization hypervisors and the public cloud. In addition, APM has consistently remained a key part of the toolset for IT operations teams, even as other trends – like waterfall-style software delivery and service-oriented application architectures – have come and gone.

Yet at the same time, despite the fact that the APM market is more than two decades old and is saturated with offerings from a long list of vendors, APM platforms are changing tremendously in response to new market demands. APM solutions designed just a few years ago are no longer sufficient for meeting the needs of modern workloads, which themselves have evolved tremendously in a short span of time.

The complexity of modern applications

One reason why traditional APM tools no longer suffice is the enormous complexity that characterizes modern applications. The emergence over the past decade of microservices-based architectures, containers, serverless functions, multi-cloud strategies and similar trends, have led to a new generation of applications that involve considerably greater complexity than their predecessors.

Rather than running as monoliths hosted on a single server, modern applications are composed of a web of microservices that are spread across sprawling clusters of servers and abstracted away from underlying infrastructure by a platform like Kubernetes. Many also depend on storage systems that exist independently from the application itself, and on APIs, which introduce another layer of complexity that must be managed and monitored to guarantee acceptable application performance.

Adding to the complexity challenge is an increase in the scale of systems that need to be monitored. Not only are existing application deployment environments increasing in size, but the rapid digitization of legacy systems – as well as the demand for reliable and efficient performance of those systems in digital form – means that there is now simply more to monitor. As a result, Gartner believes "enterprises will quadruple their APM monitoring due to increasingly digitalized business processes from 2018 through 2021 to reach 20% of all business applications[1]".

Rising expectations for end-user experience

A second factor behind the fast-changing nature of APM tools is ever-increasing standards for the end-user experience in the software industry. Whether end-users are customers who are being served by a public-facing application, or internal users of a line-of-business application that is used inside a company, customers demand applications that are fast, reliable and easy to use.

Companies that fail to deliver on these expectations risk losing customers to competitors' applications in the case of public-facing apps; or suffering business inefficiency due to performance problems with internal line-of-business apps.

APM Must Evolve

Both of the pressures described above – the substantial increase in application complexity, and the need to provide a near-flawless customer experience – necessitate a new approach to APM.

APM tools that are designed only to monitor monolithic applications, and that lack the ability to interpret a complex Web of microservices interacting via APIs, cannot effectively manage the performance of modern applications. Nor can legacy APM tools keep up in a world where virtually any amount of downtime, as well as application delays mea-

[1]Gartner Broaden Application Performance Monitoring to Support Digital Business Transformation, December 2018

sured in milliseconds, can drive users away or snowball into critical business-productivity disruptions.

One critical area of functionality that next-generation APM tools require to meet these demands is the ability to leverage AIOps, which means using machine learning and advanced analytics to identify, understand and address software performance problems. Monitoring systems that are based on hard-coded alerting thresholds, or those that require human engineers to analyze problems manually, are no longer sufficient.

At the same time, APM tools must offer functionality beyond mere monitoring. The ability to use AI-powered insight to resolve problems automatically and in real time is a critical differentiator between old-generation APM tools that merely find problems, and next-generation platforms that proactively resolve them.

In addition, monitoring the highly scalable applications that organizations deploy today requires APM tools that are themselves elastic and able to scale without limit. APM agents that must be deployed manually, or that cannot keep up with rapid fluctuations in application scale, cannot adequately address fast-changing, microservices-based applications that are hosted on elastic platforms like Kubernetes.

Understanding modern APM

The purpose of this guide is to help stakeholders make sense of the rapidly changing world of APM. The following chapters detail the new set of challenges that modern APM tools must solve, explain the role that AI and machine learning play in addressing them and describe the future of APM platforms. The sweeping changes that are transforming the APM market can be difficult to understand fully, even for teams who deploy APM tools on a daily basis, and this guide aims to offer some clarity amidst this rapid evolution.

In this way, this guide gives decision-makers the insight they need to understand why one of the IT industry's best-established markets is undergoing disruption, and which APM functionality they should look for when selecting tools to safeguard the performance of modern application environments.

CHAPTER 1

What is APM?

Let us begin with the basics: an overview of what APM means, how it has evolved since it originated more than two decades ago and why it is so crucial within the realm of modern software delivery.

Defining APM

Different analysts offer somewhat different takes on APM; some definitions are broader than others. However, most analysts would agree that APM refers to any tool designed to monitor application availability and performance in order to help optimize user experience and maximize application efficiency.

APM tools come in a variety of forms. They can be as simple as a tool that pings an application or server to check whether it responds, and as sophisticated as an AI-powered system that aggregates and analyzes volumes of data from disparate sources in order to predict and react to performance problems.

In addition, application tracing and diagnostics, which help developers to get to the root of software crashes or performance issues, fall under the category of APM. So do cost-optimization tools designed to help IT operations teams strike the right middle ground between under- and over-allocating infrastructure to workloads.

Origins of APM

APM has a long history. By the late 1990s, the increasingly critical role that digital systems played in business operations led developers to build the first generation of performance monitoring and management tools. Exemplified by platforms like Nagios, these tools worked in a relatively straightforward fashion: They collected basic data about whether systems were up or not, as well as how many spare compute, memory, storage and other resources they had available. If these characteristics fell below a certain level, the monitoring systems sent alerts to IT teams so that they could investigate further.

These systems evolved over the course of the 2000s and early 2010s to support new types of infrastructure, such as virtual machines, cloud-based environments and containers. Yet beyond compatibility with more types of infrastructure, most APM platforms saw little change with regard to monitoring functionality during this time. They remained essentially reactive tools, dependent upon IT engineers to analyze and respond to the alerts they generated. They also lacked the ability to evaluate the significance of one problem relative to another, or to trace problems to their root causes.

It is only in the past half-decade that APM platforms have begun expanding beyond this basic set of functionality. Using AI, a new generation of tools has gained the ability to perform intelligent, data-based assessments of complex application environments, thereby moving beyond the simplistic threshold-based alerting that characterized earlier tools.

Why APM matters

In the most basic sense, APM is important because it automates the monitoring and management tasks required to track the availability and performance of applications – and, by extension, to guarantee a positive user experience. To monitor even simple applications manually would require significant time and effort on the part of IT engineers, and is not feasible at scale. And it is certainly impossible for teams contending with highly complex, constantly-changing applications.

It's important to note, too, that although APM in its most basic sense is about monitoring applications, modern APM encompasses a much broader range of functionality. Its purpose is not merely to generate alerts when something goes wrong, but to help IT teams know which alerts to prioritize by using AI and machine learning to understand which incidents are most critical. It also plays a critical role in helping teams to monitor the user experience across disparate channels by tracing users' behavior as they move between applications – a task that would be very difficult to perform manually, given the multiple interfaces and data sources involved.

In addition, although APM is most often used by IT operations teams, the data and insights that APM tools collect play a central role in helping developers, too, to identify opportunities for improving the performance and reliability of the applications they write. In this respect, APM tools that can use data to assess complex application behavior and make analytics-driven recommendations play an important role in bringing IT operations teams and developers together and achieving the goals of DevOps.

Finally, modern, AI-powered tools are important because they enable automated, real-time response to many application performance issues. Rather than waiting on IT engineers to receive an alert, investigate the problem and formulate and execute a remediation plan, the emerging generation of APM tools can leverage machine learning to resolve issues automatically.

When combined, these various facets of AI-driven APM are often referred to as AIOps, which the next chapter discusses in detail.

CHAPTER 2

AIOps: Integrating AI and Machine Learning Into the Foundation of APM

The previous chapter described what APM tools do and which types of challenges AI-powered APM platforms can address. Now, let's take a deeper look at what it means to integrate AI into APM platforms in order to achieve what has become known as AIOps.

What is AIOps?

As defined by Gartner, "AIOps combines big data and machine learning to automate IT operations processes, including event correlation, anomaly detection and causality determination."[2] Within the context of APM, AIOps means using data analytics to detect relevant incidents or patterns within application behavior, analyze them and, when feasible, take steps to remediate problems automatically.

In the event that AIOps cannot resolve issues automatically, it can at least perform root-cause analysis, providing IT engineers the insight they need to identify and address the source of a problem quickly. This visibility is especially critical in today's complex application environments, where a surface-level performance issue such as slow application performance could be caused by a variety of underlying problems, which IT teams may struggle to identify when relying on manual investigation alone.

AIOps can also be used to predict future application behavior, and suggest remediation steps before problems have even fully materialized.

Key components of AIOps

The AI and machine learning that power AIOps are founded upon several

[2] Gartner Glossary, AIOps (artificial Intelligence For It Operations), https://www.gartner.com/en/information-technology/glossary/aiops-artificial-intelligence-operations

key components, all of which must be in place to make AIOps-powered APM tools work properly:

- **Data**
  Data is at the core of AIOps. A data lake data warehouse is required to store data. Typically, data for AIOps is organized using a time-series structure, which allows APM tools to track patterns over time and identify the root cause of performance problems.

- **Analytics**
  Machine learning algorithms allow AIOps tools to make sense of the data that they parse

- **Support for disparate data types**
  Gaining holistic insight into application performance requires the ability to work with as many types of data sources as possible. Toward that end, modern APM tools must be able to collect and analyze not just traditional system logs but also application tracing data, ephemeral log data produced by containers and serverless functions, data generated by business systems and other data sources that were not conventionally used for APM. They must also be able to integrate with any and all types of software tools that organizations use, even if those tools are designed by different vendors and structure data in different ways.

- **Flexibility**
  Because the data and performance challenges that AIOps tools must contend with range widely in type and scope, successful AIOps platforms must be flexible and adaptable, able to support a broad range of scenarios and adjust their operations as needs change.

- **Self-learning**
  The best AIOps tools are those that leverage past experiences to hone their predictive and remediative abilities over time so that they can continuously improve their ability to detect, analyze and solve performance issues.

- **Intelligent remediation**
  As noted above, the ability to remediate performance problems automatically is one of the key features that set modern, AIOps-powered APM platforms apart from their predecessors. Thus, to deliver fully

on its promise, AIOps must be able not only to detect and interpret performance problems, but also to fix them automatically. Automatic remediation may not be possible in all circumstances, but it should be a primary goal. And in cases where tools cannot automate remediation completely, APM tools should offer semi-automation features for reducing the manual effort required from engineers. Over time, the tools can learn from manual remediation workflows so that those workflows can become automated.

In theory, IT teams could build each of these components from scratch in order to integrate AIOps into their APM workflows. But because doing so would require tremendous effort (not to mention create a large ongoing maintenance burden), a better approach is to adopt AIOps platforms that have APM built in, and provide the requisite data management, machine learning and intelligent remediation features out-of-the-box.

*For full details on the what, why and how of AIOps, please refer to Broadcom's **Definitive Guide to AIOps**, which is available as a free download.*

CHAPTER 3

Understanding User Experience and Behavior

As explained in the introduction to this guide, ever-increasing expectations regarding the end-user experience is one reason why APM is undergoing a transformation, and why AIOps has become so central to the next generation of APM solutions.

Changing user experience demands

To be sure, providing a positive end-user experience has always been important for businesses. Yet never before have expectations been as high as they are today. Consider the following data points, which drive home how much users expect of their digital experiences, and how much businesses have to lose by failing to live up to them: gates and analyzes volumes of data from disparate sources in order to predict and react to performance problems.

- More than half of visitors to websites and Web apps who use mobile devices will abandon the site if it fails to load fully within three seconds.
- Nearly 90 percent of mobile users will stop using an app if they experience performance problems.
- 86 percent of software buyers indicate that they are willing to pay more for software that delivers a great user experience.
- According to Gartner's 2019 Market Guide for Digital Experience Monitoring, by 2023, 60 percent of digital business initiatives will require I&O to report on users' digital experience, up from less than 15% today.[3]

These and similar statistics drive home how high user expectations have become, and how important meeting them is to business' ability to retain customers and stay ahead of their competitors.

[3] Gartner Market Guide for Digital Experience Monitoring, September 2019

It’s worth noting, too, that it’s not only in the context of public-facing applications that excellence in user experience is crucial. Internal line-of-business applications must deliver fast and reliable performance in order to keep employees efficient and avoid disruptions to business operations. Users of line-of-business apps might not be able to abandon the apps or switch to a competitor’s offering – instead, they have to use whichever apps their employer requires – but their productivity will nonetheless be undercut by poorly-performing apps.

Digital experience monitoring

Indeed, guaranteeing a positive user experience has become so critical that an entire discipline has emerged around it. Digital experience monitoring refers to using a variety of tools and monitoring methods to optimize the way that humans experience software, as well as to make machine-to-machine interactions as efficient as possible.

Digital experience monitoring involves more than just APM, and a full discussion of the topic is beyond the scope of this guide. But having an intelligent APM solution in place is one essential element of digital experience monitoring, because APM provides the following insights:

- An omnichannel, end-to-end view of software environments, which enables businesses to track the user experience across multiple applications and platforms;
- Insights into the performance of both Web and mobile apps;
- Business KPIs and metrics such as user retention rates, drop-off frequency and revenue generation; and
- Performance and crashes by software systems, and the way they correlate with user usage statistics.

In short, it is not possible to deliver an excellent user experience without the help of an APM solution that can monitor any type of application or environment at any level of scale necessary, while also helping to prevent the performance disruptions that will drive users away from applications.

CHAPTER 4

Making sense of modern software interactions

A second challenge that modern APM tools must meet is the enormous complexity that characterizes modern software environments (and the infrastructure that hosts them). This complexity exists on multiple levels and dimensions. The following chapters discuss each of them, starting with the topic of the way modern applications (and their constituent parts) interact.

How modern apps interact

Historically, interactions within software systems were relatively straightforward and easy to trace. An application that needed to access persistent storage would have one storage location to manage and one procedure for sending and receiving data from it, for example. Likewise, a server delivering data to a user over the network would follow one protocol, and data could be easily traced as it crossed the network. And because applications ran as monoliths, there were few intra-application interactions or exchanges to monitor.

In contrast, interactions within and between today's applications are tremendously complex. Modern apps are often run as a set of microservices, each of which send and receive data from each other constantly, often using multiple APIs and overlay networks. Applications are distributed across multiple servers, and it is common to deploy multiple instances of the same application in order to improve reliability. As a result, it is not always possible to know which server is hosting which application at any given moment, or which application instance will respond to a given request. Network endpoints and IP addresses are configured dynamically and change constantly, requiring applications to automatically discover each other and keep track of ever-changing network topologies.

Amidst this complexity, detecting and understanding performance problems can be very difficult. An issue that manifests itself as a performance degradation with one part of an application (such as the frontend) may have its root cause in a different part (such as a storage microservice that interacts with the frontend), or in the APIs or protocols that define how services interact. (In some cases, performance problems may not lie in an application itself, but rather in a buggy or poorly implemented API that is external to the application.)

Making sense of application interactions

In order to contend with this complexity, APM tools require a range of functionality that was not traditionally part of APM. Key characteristics include:

- Support for interfacing efficiently with the observability systems that are built into many modern applications - Increasingly, applications offer native functionality for generating monitoring data through tracing frameworks (such as OpenTracing and Zipkin) and metrics generation tools (like Prometheus). APM tools must be able to take full advantage of these interfaces, even if the frameworks used to implement them vary from one application to another. Equally important is the ability of APM tools to fill in the gaps by supplementing these types of native application performance data with other data sources.

- The ability to trace transactions across distributed environments - Application tracing that works only for a single process is not enough when applications run as complex webs of microservices, each with its own process (or possibly multiple processes, if there are multiple instances of the same microservice).

- Multiple analytics approaches for making sense of application transactions and achieving observability - Because there is no one standard way to architect a microservices application or a single type of API that defines interactions between applications, APM tools must offer the flexibility to analyze interactions in a variety of ways.

- The ability to map applications to distributed, scale-out infrastructure - When applications are hosted using platforms like Kubernetes, the application itself typically exposes no data that indicates which specific servers are hosting the application or how its networking infrastructure is structured, because Kubernetes configures those mappings automatically without informing the application. APM tools must therefore be able to dig deeper on their own and understand the relationship between software and the infrastructure that hosts it – even if there are abstraction layers like Kubernetes in between.

- The intelligence to trace surface-level problems to their root causes - In today's complex application architectures, the "symptoms" of a performance issue may or may not be an obvious reflection of the underlying cause; an application that appears to be running out of memory might indeed simply not have enough RAM available on the server that is hosting it, for example; but the problem could also lie in resource quota configurations within a container orchestration framework that prevent the application from accessing all of the RAM available. APM tools must be able to perform the deep and broad analysis required to trace these issues to their root causes.

AIOps is the only way to meet these challenges efficiently. While it may theoretically be possible to understand complex application architectures and software-to-hardware mappings through manual analysis, doing so is just not feasible at any kind of scale.

CHAPTER 5

Observability for Distributed Cloud Applications

A second key contributing factor to the complexity of modern applications is the fact that they are often hosted on highly distributed architectures that span multiple data centers and comprise a mix of different types of services.

The rise of distributed applications

Distributed application architectures are not new; the practice of spreading workloads across servers stretches back decades. Yet over the past ten years or so, several new trends have converged to lead to the total redefinition of what it means to run (and, by extension, manage the performance of) a distributed app:

- **Hybrid cloud and multi-cloud architectures**
  Today, the same application might be hosted on infrastructure that spans several on-premises data centers and/or several clouds. Or, one part of an application (such as the frontend) might be hosted in one cloud, while another (like a database) runs in a different cloud.

- **Containers**
  Container frameworks like Docker, along with orchestration platforms like Kubernetes that make containers practical to deploy at scale, have significantly increased the scale of distributed applications. A single Kubernetes cluster might include dozens of individual servers and hundreds of container instances, with individual containers constantly moving between one host server and another.

- **Microservices**
  When applications are developed as a set of discrete microservices, they introduce a new kind of distributed architecture. Even if a micro-services application is hosted on a single server, the microservices themselves comprise a distributed environment.

These architectural changes have added a great deal of flexibility, reliability and cost-optimization to applications, but they have also multiplied the level of complexity surrounding them. Compared to the server clusters of old, which typically included a half-dozen or so physical or virtual machines that hosted a handful of monolithic applications, today's distributed applications run on a much larger scale, include more layers of abstraction (physical servers, virtual servers, containers and orchestration platforms) and involve many more individual application parts.

APM for distributed applications

As a result, modern APM tools require a much greater ability to handle the complexity that comes with distributed application architectures. Monitoring servers or applications on an individual basis is no longer sufficient; APM platforms must instead be able to aggregate and analyze data from an entire server cluster or multi-cloud infrastructure in order to identify and understand performance trends within the applications that span that infrastructure.

Likewise, APM tools must be able to distinguish normal changes in infrastructure layout from performance problems. For example, the number of servers within a Kubernetes cluster can fluctuate naturally without posing a problem, and an APM tool that generates an alert every time a server goes offline will generate unnecessary alarms. Yet if the number of servers in the cluster is at risk of no longer being able to support the workload at hand, performance problems could occur. An effective APM tool in this scenario must, therefore, be able to use data analytics and machine learning to assess which infrastructure size is appropriate for a given workload; then, intelligently determine when to generate alerts or take other actions in response to infrastructure changes.

A third challenge for APM tools in distributed environments is that there is often no such thing as "normal." Depending on application demand, the number of application instances running at a given time, or the resources allocated to them, can fluctuate constantly as load balancers and auto-scalers continuously readjust configurations. For this reason, the traditional APM practice of defining a known-good baseline, then labeling deviations from it as problems, doesn't work. APM tools must instead be able to understand constantly changing environments and distinguish normal fluctuations from those that may signal a looming performance problem.

Using AIOps to monitor distributed environments

AIOps is the only practical approach to handling the complexity and scale involved in monitoring distributed environments. With AIOps, APM platforms can collect data from the entire span of distributed environments, then use machine learning to interpret the complex patterns within those environments. They can establish dynamic baselines to measure what constitutes a true anomaly within ever-changing environments, and they can predict how changes in the architecture of a distributed environment (such as the removal of some servers, or the migration of a workload from one cloud to another) will impact application performance and user experience.

CHAPTER 6

Using Modern AIOps to Gain Continuous Insights across DevOps

The previous several chapters of this guide discussed changes in user experience expectations and software design and deployment patterns, which have made traditional APM tools and strategies obsolete.

In this chapter, we'll shift the discussion toward a look at the new opportunities that AIOps opens up for the APM ecosystem. Specifically, the chapter discusses one key use case for AIOps-powered APM: enabling a more efficient and productive DevOps practice.

Visibility as the key to DevOps

DevOps is an approach to software delivery that emphasizes collaboration between developers and IT operations teams. There is no specific type of tool or process that is required to achieve DevOps; DevOps can be implemented in many different ways.

But whichever approach an organization takes, end-to-end visibility into the software delivery pipeline is critical for successful DevOps. The reason why is that developers and IT operations engineers cannot collaborate effectively with each other if they lack visibility into what the other group is doing.

Typically, developers work only with the part of the software delivery pipeline that involves code production and integration, whereas IT engineers work with software after it has been deployed. By default, then, developers do not know about relevant performance trends or problems in production applications that they should take into consideration when planning their next coding sprints. At the same time, IT operations teams have no default means of knowing about performance-related considerations within code written by developers.

Using AIOps to achieve continuous visibility

With the help of AIOps and next-generation APM tools, however, both groups – developers and IT operations teams – can gain full visibility into the entire software delivery pipeline and, in turn, collaborate effectively in the spirit of DevOps.

The ability to collect data about the status of all stages of the delivery pipeline is perhaps the most obvious realm of functionality for achieving continuous visibility across the pipeline. But that is a traditional function of APM tools, and it represents only the beginning of what an AIOps-powered APM platform can deliver when it comes to facilitating DevOps.

At a deeper level, AIOps enables the following:

- The ability to analyze the state and health of the delivery pipeline and identify problems within the pipeline itself that may undercut the ability of developers and IT engineers to collaborate, and may ultimately harm application performance - For example, an AIOps tool could identify slowdowns in the continuous integration process (which is one key part of the broader software delivery pipeline) in order to help the DevOps team address them.

- Comparisons of application performance on a build-by-build basis - In other words, AIOps-powered APM tools can assess performance trends and patterns within one application build and compare them to other builds in order to identify which software delivery practices are yielding the best-performing builds. These insights can then help both developers and IT engineers to optimize the software delivery pipeline for high-performing applications.

- The ability of developers and IT engineers to interpret pipeline-related data in ways that are most meaningful to them - It is one thing to collect data about the software delivery pipeline; it is another to ensure that that data leads to insights that are actionable for both developers and IT engineers (especially when what is actionable for one of these groups is not for the other). AIOps can deliver analytics results that are tailored to each group's needs and abilities, thereby enabling deeper collaboration around optimization of software delivery processes.

To put all of the above another way, AIOps is the key to making sense of all of the data that emerges from the various processes within a software delivery pipeline – code integration, software testing, software builds, build deployments and beyond. Without AIOps, this data would have little meaning, or would require significant manual effort to interpret. With AIOps, disparate data can be correlated and analyzed in ways that lead to insights for all stakeholders in the DevOps process.

CHAPTER 7

The Future of Application Performance Management

Now that we have discussed the challenges that APM platforms must solve in order to remain relevant today, and we have examined how AIOps-powered APM solutions promote DevOps, let us take a look at the complete functionality that a next-generation APM platform must offer in order to meet the needs of software delivery teams today.

This chapter summarizes the future of APM tools by discussing each of the key features that will distinguish market-leading tools from their legacy predecessors.

Microservice-aware

For one, APM platforms that can meet the needs of modern applications, as well as adapt to those that will arrive in the future, must be able to monitor and interpret application behavior meaningfully within a micros-ervices-based environment.

As noted above, microservices architectures have become increasingly common over the past decade as software engineers have sought application design patterns that provide greater agility and scalability, and that also make it easier to roll out new features quickly. Yet because microservices introduce more moving parts to the application, as well as more layers of communication, APM tools that understand applications (and the data they produce) from a monolithic perspective can't keep up.

The future of APM, then, lies in part in APM platforms that can aggregate and interpret data from dozens of microservices at once – even if those microservices are hosted on multiple types of infrastructure (some as containers, for example, and others as serverless functions). And they must be able to correlate this data with data from the APIs and overlay networks that facilitate communication between microservices. It is only through this holistic, nuanced approach that APM tools can understand the full complexity of application performance in a microservices-based world.

Dynamic analytics

A second key feature of next-generation APM platforms is the ability to discern genuine performance problems in environments where flux is a constant and establishing a baseline for normal behavior is not possible.

Doing this requires constantly reevaluating application performance trends and comparing them against other sources of insight – such as historical performance information and data about application load – in order to make intelligent assessments about which changes or patterns indicate true performance issues, and which are part of the normal operations of highly dynamic environments.

Unlimited scalability

No matter how you measure them, application deployments are much larger today than they were just a few years ago. They involve more application instances, more services, more servers and (in many cases) more clouds. And their size is likely to only grow in the future.

For this reason, next-generation APM platforms must be able to scale without limit. This means not only scaling the monitoring and analytics functionality of the APM tools themselves, but also ensuring that they provide the insights and automation features required to empower IT teams to remain efficient and agile no matter how large their application deployments become.

Providing the insights necessary to keep application hosting costs low in ever-expanding infrastructures is critical, too. Although (as chapter one noted) cost optimization is only one aspect of APM, it is poised to become an increasingly important one as more and more organizations migrate their workloads to cloud-based infrastructure, where allocating more resources than necessary to a workload leads to unnecessary costs, while allocating too few may starve applications of resources and cause performance issues.

Proactive automation

To deliver value in the complex, fast-moving environments of the future, APM tools must do much more than simply collect data, identify anomalies that may indicate performance trouble and then send alerts. Instead, the tools must be as proactive as possible by taking steps to remediate performance problems automatically, wherever feasible.

For instance, consider an application that is running out of software space (and will suffer performance degradation if it exhausts available storage). This is a simple problem that an intelligent APM could solve automatically by allocating more storage space to the application (provided the tool determines that a simple lack of storage is indeed the root cause of the looming performance problem).

In complex situations, fully automated remediation by APM tools may not be possible. Nonetheless, next-generation APM tools should provide automation features that assist engineers in resolving issues faster and with less manual effort. They can do this by making recommendations about remediation steps that engineers can follow, for instance, or by performing routine remediation tasks (such as reconfiguring firewall rules or resource allocations) while engineers focus on more complex tasks that cannot be automated.

At the same time, APM tools must be able to learn which steps engineers followed when resolving issues manually and learn how to automate those workflows over time. The first instance of a given type of performance issue may require fully manual remediation. But the next time that the same problem occurs, an APM tool should be able to learn what remediation steps were taken previously, and automate them as much as possible. This process can be repeated until the APM tool is able to implement a complete remediation path automatically.

Focus on root causes

Finally, to thrive in the future, APM tools must focus not on treating symptoms, but instead on treating the disease. In other words, they must identify the root cause of performance problems and address it, rather than taking steps that only temporarily relieve or work around a performance issue while leaving its underlying cause in place.

For example, imagine an APM tool that detects an application that is responding slowly to user requests. A traditional APM tool might determine that the application needs more memory in order to improve performance, and engineers could increase memory allocation to address the issue. But if the root cause of the lack of memory is a memory leak in the application itself, the problem would need to be addressed by modifying application code. Otherwise, not only will the problem continually recur, but memory resources will be wasted on attempts to shore up application performance by allocating more memory without addressing the root of the problem.

In short: AIOps is the future of APM

AIOps is the engine that enables all of the features described above. It's what distinguishes APM tools that merely monitor and generate alerts from those that deliver the machine learning and automation features that IT teams need to manage performance challenges within application environments that comprise large-scale, fast-changing webs of microservices.

CHAPTER 8

Broadcom's solution: DX APM

Fortunately for IT and DevOps teams intimidated by the rapidly evolving nature of the APM market, implementing APM solutions that offer the performance-management features necessary to thrive today and in the future does not require starting from scratch. Companies can instead deploy a trusted solution like Broadcom's DX Application Performance Management, which has evolved along with application architectures and deployment patterns to remain an industry-leading, future-oriented APM solution.

DX APM has a long history; in fact, it originated in the earliest years of the APM market (when it was known as Wily). But today, DX APM stands out as a leader in the APM industry.

What's more, DX APM offers full integration with AIOps from Broadcom®, which offers a microservices-based approach to APM that groups independent services, such as metrics, topology store, data exports and API data, to offer a holistic overview of application health. This architecture makes it easy to scale components as the needs of application teams grow. It also allows for in-place upgrades in a rolling fashion and easy onboarding of new application deployments within existing environments, which in turn helps to deliver a lower total cost of ownership.

And because of Broadcom's long experience in the APM market, the DX APM platform includes the professional-level support and foresight necessary to ensure that the solution can meet any APM challenge, both today and in the future.

To learn more about how Broadcom can help meet your team's Application Performance Monitoring needs in today's highly complex environments, visit us at http://www.broadcom.com/apm.

Sweetcode

Sweetcode.io is a site owned and managed by Fixate IO. Its purpose is simple: To give techies a place to share what they know and impact the market with high-value, practioner-generated technical content. Sweetcode is committed to publishing tactical content that supports the growth of seasoned developers, while also serving as a platform for those who are just starting their careers. All of the content on the site is practitioner-created, and Sweetcode invests in sourcing content from under-represented coders to give them a place to share what they know.

BROADCOM®

Broadcom Inc. (NASDAQ: AVGO) is a global technology leader that designs, develops and supplies a broad range of semiconductor and infrastructure software solutions. Broadcom's category-leading product portfolio serves critical markets including data center, networking, software, broadband, wireless, storage and industrial.

For more information, go to www.broadcom.com