The butterfly effect

In 1972, Edward Lorenz stood before a group of his peers at the Sheraton Park Hotel and changed the course of science with a single question:

“Does the flap of a butterfly’s wings in Brazil set off a tornado in Texas?”

While we’ll likely never know the answer to this proverbial question, the value comes from the larger point it evokes—that seemingly insignificant events can lead to significant changes in the larger system.

The idea came to be known as the “butterfly effect”—and in the fast-paced and continuously changing world of software applications, we see it in action every day.

Almost anything that touches an application either improves or degrades performance. Determining whether it’s attributed to the infrastructure, a coding change, an API, the network, application architecture or a connected service is the challenge.
Finding a Needle in a Stack of Needles

While it’s always been difficult to see what’s going on inside an application and identify the root cause of a performance issue, it’s become exceedingly more challenging today.

Modern applications run on virtual machines and containers and branch outside of application servers to connect to any number of microservices to complete a transaction. And that branching often occurs outside your premises, in the cloud. This means modern application architectures look less like the traditional three-tier model, and more like a sprawling biological ecosystem.

When a problem occurs, it can ripple across the interconnected components, generating an effect far from where the initial change took place. This complexity is compounded by the fact that change is occurring more often. In fact, 33 percent of DevOps teams report that they release one or more times daily.¹

Additionally, when app components are highly distributed, communication becomes network-centric with independent microservices communicating asynchronously via APIs. Thus, there’s a rampant increase in messaging across a variety of systems and dependencies—all potentially meaning more complex conditions, alerts and system checks which puts more strain on IT staff.

Identifying the source of a problem is often akin to finding a needle not in a static haystack, but in a swirling mass of other needles.

Coping With the Data Deluge

It’s not that teams lack reliable monitoring tools; in fact, far from it. There are plenty of tools available to monitor one piece of the stack at a time. But when distinct teams are managing performance data in silos, what’s missing is context. For example, consider a case where a network alarm indicates an increase in traffic latency. Modern tools can identify when and where this problem occurs, but because the data is managed in silos, application support teams have no indication which business services are impacted.

Another problem is the sheer volume of data. When you set up alerts, run books and monitoring for each microservice, container, virtualized resource and network you’re dealing with, there’s an exponential increase in the amount of data that needs to be collected and analyzed. Your IT operations teams can quickly become buried under a deluge of alerts. Of course, not every alert warrants a reaction, but separating the events from the non-events becomes an extremely difficult task.

This problem is made worse by traditional approaches to baselining. In the past, when application performance and fault conditions were more predictable, teams could reliably forecast acceptable performance measurements, set those manually and then throw an alert when those thresholds were passed. But this approach doesn’t work anymore. Systems act in unpredictable ways and what used to be considered bad—like high CPU utilization—might be expected. And even if predictions are accurate, setting and maintaining so many isn’t sustainable.

The simple fact is, traditional approaches to monitoring cannot scale to manage the increased volume, variety and velocity of data produced by today’s dynamic application architectures. The price is unpredictable end-user experiences, escalating costs and lost resources and time.
So how do you take this chaotic data environment and convert it into something that makes sense and is actionable?

The answer—you need to embrace a new application performance management (APM) model. One that enables the analysis of multiple data sources, provides application-to-infrastructure correlation and infuses monitoring with analytics-driven insights.

Only then can you gain complete visibility across modern application environments. And when you do, you'll know exactly how a seemingly insignificant change within a single microservice unleashes, like Lorenz's butterfly, a ripple effect across the broader ecosystem. What's more, you'll have the data you'll need to act before this change impacts the end-user experience. And, you'll have the deep insights required to make the right decisions to continuously improve application performance and business outcomes.

It starts with a big picture view of performance.
We’ve all heard the adage, “what doesn’t get measured, doesn’t get managed.” So, we capture comprehensive application, infrastructure, network, container and cloud activity.

Diverse and unstructured, this data is often collected and analyzed in piecemeal fashion. But full insight into a digital experience requires that you correlate this data, layer it and create a multi-dimensional and cross-domain view. We call this a unified data model and with it, you can automatically and contextually surface information to deliver richer performance insights to every team.

For example, let’s revisit the network latency issue referenced on page 4. By using a unified data model to layer in application topology and metrics, application teams can see if the network issue carries any significance to the immediate business. So, rather than wasting time chasing alerts and setting up war rooms, application and IT operations teams can work collectively to resolve the issues most detrimental to the end-user experience.

But the value of this model goes beyond swift root-cause analysis. It can be leveraged to align teams around shared business goals. For example, application development teams can analyze their code modifications in the context of how performance improvements correlate to better business outcomes. Whereas cloud architects can compare performance outcomes across different microservice deployment patterns to determine where the business is getting the most value from its investments.
Modern application architectures with their distributed systems generate an exorbitant amount of data. Pulling it all into a coherent view is one challenge. Deriving insights to know how, when and where to act is another. But with the right APM solutions, it’s possible to continuously observe and learn application behavior with minimal intervention.

On page 3, we discussed how releases are occurring with greater frequency. This means any number of modifications are being introduced into the application ecosystem in a near-continuous fashion. And any one, or a combination of these changes could wreak havoc on application performance. But the question remains—what components caused the issue?

By correlating application performance with underlying system and network metrics, you can automatically detect changes, extract only the data that is relevant to that change and correlate it to the application or service being monitored to measure impact. Additionally, modern APM solutions have capabilities that can track back in time to help teams identify all the possible causes of a performance problem—the “patient zero”—be that a new code build or configuration changes.

Data-driven insight also supports another key requirement—noise elimination. Alert storms are a byproduct of a highly modular and distributed application architecture. But it doesn’t have to be this way. By replacing binary, pass/fail performance baselining with “how wrong for how long” algorithms, you can automatically distinguish small nuisance alerts from irregular trends and anomalies worthy of action. This means support teams only get alerted when there are business-impacting issues.
Once you’ve isolated business-impacting issues, the next step is to answer the question “why is the experience poor?” For many, that means assembling a team of cross-functional experts and hunting for a root cause in a sea of metrics, visualizations and performance indicators. The process is time-consuming and resource-intensive. But what if you could radically simplify the triage and diagnosis process? With the right APM solution, you can.

Instead of parsing through caches of data, modern APM automatically gathers key pieces of information that are tied to the problem or anomaly, including metrics relaying when the problem first and last appeared and a map of affected components. All this data is correlated with the application components associated with the degraded experience and presented in a single view.

This means the time-consuming steps associated with war-room scenarios are eliminated. Most importantly, teams can gain a common and shared understanding of why the customer experience is poor to speed problem resolution and drive application performance improvements.
We just covered how analytical insights can help teams triage problems and optimize performance in production, but it’s equally valuable during design and development—where teams need to check the performance of components as they are added to the larger code base.

Take for example a development build process using Jenkins®. Developers want to assess how their coding changes could impact application performance, but want to do this in context of their immediate work and without having to access unfamiliar tools. Modern APM solutions allow this, immediately presenting performance data during the actual build process—straight from their Jenkins interface. This way they can see the impact of every coding change on performance before it’s released in production where it’s costlier to fix.

And when changes are applied—be it new code, infrastructure configuration changes or new container deployments—it’s logged. So, if there are any ensuing performance issues, you can easily correlate them with changes at any point in time to pinpoint the source of the problem faster.

In addition to monitoring during build and integration processes, analytics can be embedded within continuous delivery, testing and release processes to identify opportunities to increase software quality. For example, an integration between APM tools and CA BlazeMeter®—a solution to run massive load tests in the cloud—enables testing teams to pinpoint what conditions slow down app performance at scale.

By shifting analytics insights left into development, you can accelerate the delivery of high-quality applications.
Microservice architectures and container-based apps are the new norm. In fact, **83 percent of organizations are actively using containers today.**¹

Containerized applications present a new set of complex challenges. The increase in components and dependencies renders traditional topology mapping obsolete, while their dynamic nature—where they can be replaced in seconds—makes tracking overall service performance highly problematic.

To do so, you now need to understand the dependencies and correlate performance between the application, containers and host, as well as the health of individual component whenever necessary.

With the right APM tool, you can automatically detect and discover containers, dependencies and critical communication flows without needing to instrument individual containers. In these dynamic environments, understanding overall service performance is paramount regardless of their dynamic state. To do this, modern APM solutions can correlate app performance within and across microservices, containers and host infrastructure to quickly pinpoint problematic conditions. Furthermore, a unified data model allows previously dense and indecipherable topologies to become clear and actionable views into containerized application performance—from any number of perspectives.

---

CA Application Performance Management (CA APM)

The solution you need to deliver actionable insights across increasingly complex and chaotic application environments.

With CA APM you can:

- **Manage what comes first—customer experience**: See the health of apps across channels, services and locations. Align teams to business outcomes with sharable experience dashboards.
- **Find and fix problems faster**: Assisted triage delivers intuitive root-cause analysis workflows based on analytics.
- **Eliminate alert fatigue**: Use industry-proven differential analysis to quickly recognize, understand and differentiate more critical problems and anomalies from nuisance alerts and alarms.
- **Simplify complex topologies** into role-relevant views specific to the task at hand with a unified, attribute-driven data model and team center perspectives.
- **Gain fast and accurate feedback**: Understand the impact of changes on application performance at any point in time. Quickly locate where problems originate—in code, releases or configurations.
- **Shift left for higher quality**: Seamlessly integrate across the DevOps toolchain—including continuous integration, delivery and testing—to increase quality and help prevent costly defects from reaching production.

- **Gain richer performance insights**: Automatically collect, correlate and layer application performance information with associated infrastructure topology and metrics.
- **Achieve fast value with low overhead**: Use a zero-configuration agent to automatically collect deep transaction traces whenever a problem arises.
- **Monitor performance from mobile to mainframe**: Gain end-to-end insight into app performance from APIs, partner-systems and mobile apps to back-end processes hosted on mainframe systems.
- **Manage whatever comes next**: Manage modern distributed application environments, microservices and containers with automatic container discovery and flow maps and massively scalable metric collection and analytics.

The solution is available on-premises or through **CA Digital Experience Insights**, our SaaS-based digital operations monitoring and analytics platform.
To learn more, visit ca.com/apm

CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate—across mobile, private and public cloud, distributed and mainframe environments. Learn more at ca.com.

Copyright © 2017 CA, Inc. All rights reserved. All marks used herein may belong to their respective companies. Jenkins® is a registered trademark of Software in the Public Interest, Inc. This document is provided for informational purposes only. Any functionality descriptions may be unique to the customers depicted herein and actual product performance may vary. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this document “as is” without warranty of any kind, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, business interruption, goodwill or lost data, even if CA is expressly advised in advance of the possibility of such damages.