Tools to Monitor and Visualize Microservices Architecture

Previous articles in *ProgrammableWeb's* microservices series look at what microservices are and explain differences between monolithic and microservices architectures.

Once a distributed application is built and deployed, it is crucial to monitor and visualize it to make sure the software is reliable, available, and performs as expected. That isn't necessarily easy.

The heterogeneous and distributed nature of applications driven by a microservices architecture make monitoring, visualization, and analysis a difficult prospect. Traditional application monitoring and performance management (APM) solutions are not suited for today's complex distributed applications.

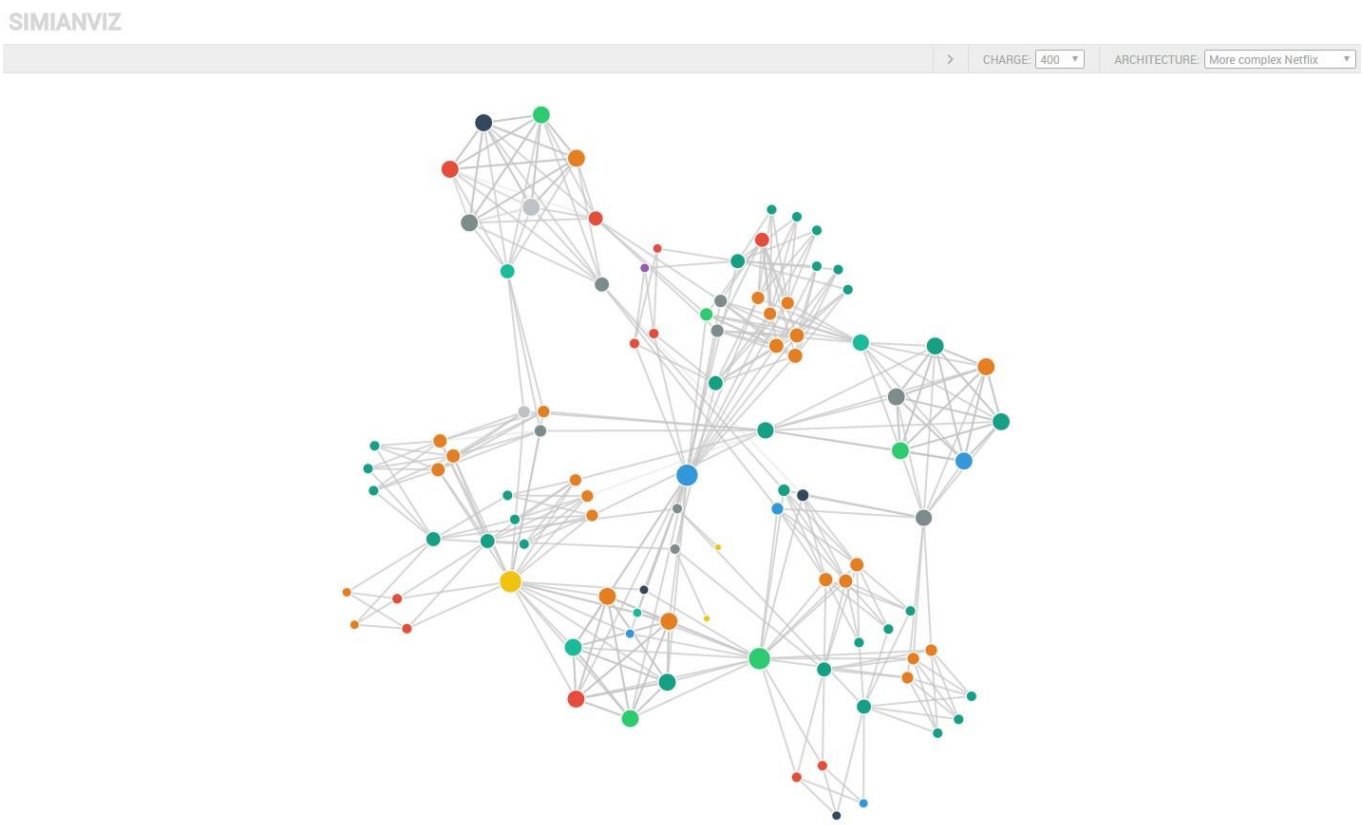
Fortunately, several new APM solutions have been launched within the past few years to address these issues. These APM solutions take advantage of advanced technologies such as artificial intelligence (AI), machine learning, and graph analysis to monitor, visualize, and analyze microservices architectures. Many of these modern APM solutions also include distributed tracing and topology visualization capabilities necessary for effectively managing microservices architectures.

Distributed Tracing and Topology Visualization

Among the current open source distributed tracing systems are Zipkin, HTrace, X-Trace, and Trace. There is also the OpenTracing Project, which aims to provide vendor-neutral APIs so that distributed tracing and context propagation can be implemented on all popular platforms.

Adrian Cockcroft, a technology fellow at Battery Ventures and former chief architect at Netflix, described distributed tracing in a recent presentationsaying, "Distributed tracing systems collect end-to-end latency graphs (traces) in near real-time. You can compare traces to understand why certain requests take longer than others."

Not all of the open source distributed tracing systems available include topology visualization capabilities, which can be an important feature. Topology visualization maps or diagrams the layout of applications in a microservices architecture and in other distributed applications. Doing so is critical when you need to discover performance issues and other problems.



Screenshot of SimianViz demo, more complex Netflix visualization: View live demonstration.

Adrian Cockcroft recently released a new open source tool, SimianViz(formerly Spigo), that generates large-scale simulations of complex microservices. Companies can use these simulations for visualizing topologies and for stress testing microservices monitoring solutions without having to set up large test configurations.

Major technology companies, such as Netflix and LinkedIn, have built their own distributed tracing and performance monitoring solutions. For Netflix, the need for its several distributed tracing tools was driven by its need for scalability, as most commercial tools are unable to scale at the level Netflix requires. Netflix also uses a variety of visualization tools including on-demand CPU flame graphs for analyzing and optimizing Java and Node.js application performance.

LinkedIn has a real-time distributed tracing system that uses Apache Samza results to build real-time call graphs. The call graphs are used for performance optimization and root cause analysis of the LinkedIn distributed architecture.

Most companies don't have the extensive resources of companies like LinkedIn and Netflix, so building a custom distributed tracing and performance monitoring solution from the ground up

may not be possible. Fortunately, there are a number of tools that developers in any size company can use for monitoring and visualizing distributed applications.

Monitoring and Visualization of Microservices Architectures

There's a lot going on in any system built on a microservices architectures. A microservices architecture typically consists of dozens, sometimes hundreds, of fine-grained services; every user transaction goes through many of those services. In addition, transactions are often asynchronous, involving multiple concurrent service requests. Traditional APM products are typically unable to monitor distributed applications that process multiple concurrent service requests.

Their inherent complexity and high scalability requirements have led to the creation of application monitoring and visualization tools that use machine learning, graph analysis, distributed tracing, topology visualization, and other cutting-edge technologies.

Here are just a few examples of these solutions, so you can get an idea of the tools available to help you understand what's going on in your software.

AppDynamics

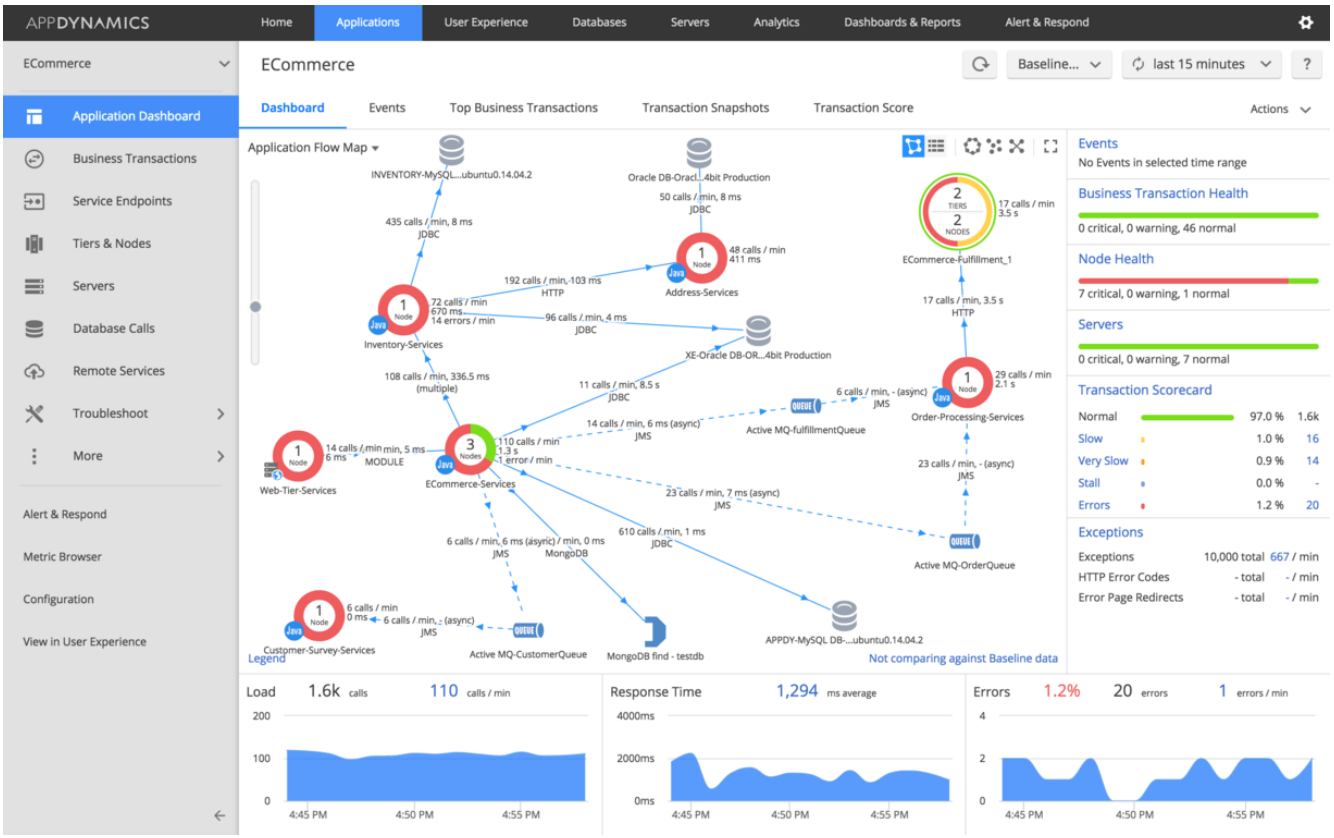


Image Credit: AppDynamics

While AppDynamics has been around for quite some time, the company launched its machine-learning powered APM product in June 2015, to monitor, manage, and analyze complex architectures such as microservices. AppDynamics shows application performance in real time and automatically discovers application topology and interdependencies. Its APM tool also includes distributed tracing, topology visualization, and dynamic tagging.

Developers can use AppDynamics to determine the health of a distributed application, understand transaction paths, determine the root cause of service failures, and gain other key insights into a microservices architecture. AppDynamics APIs can help extend and customize the platform's features.

Instana

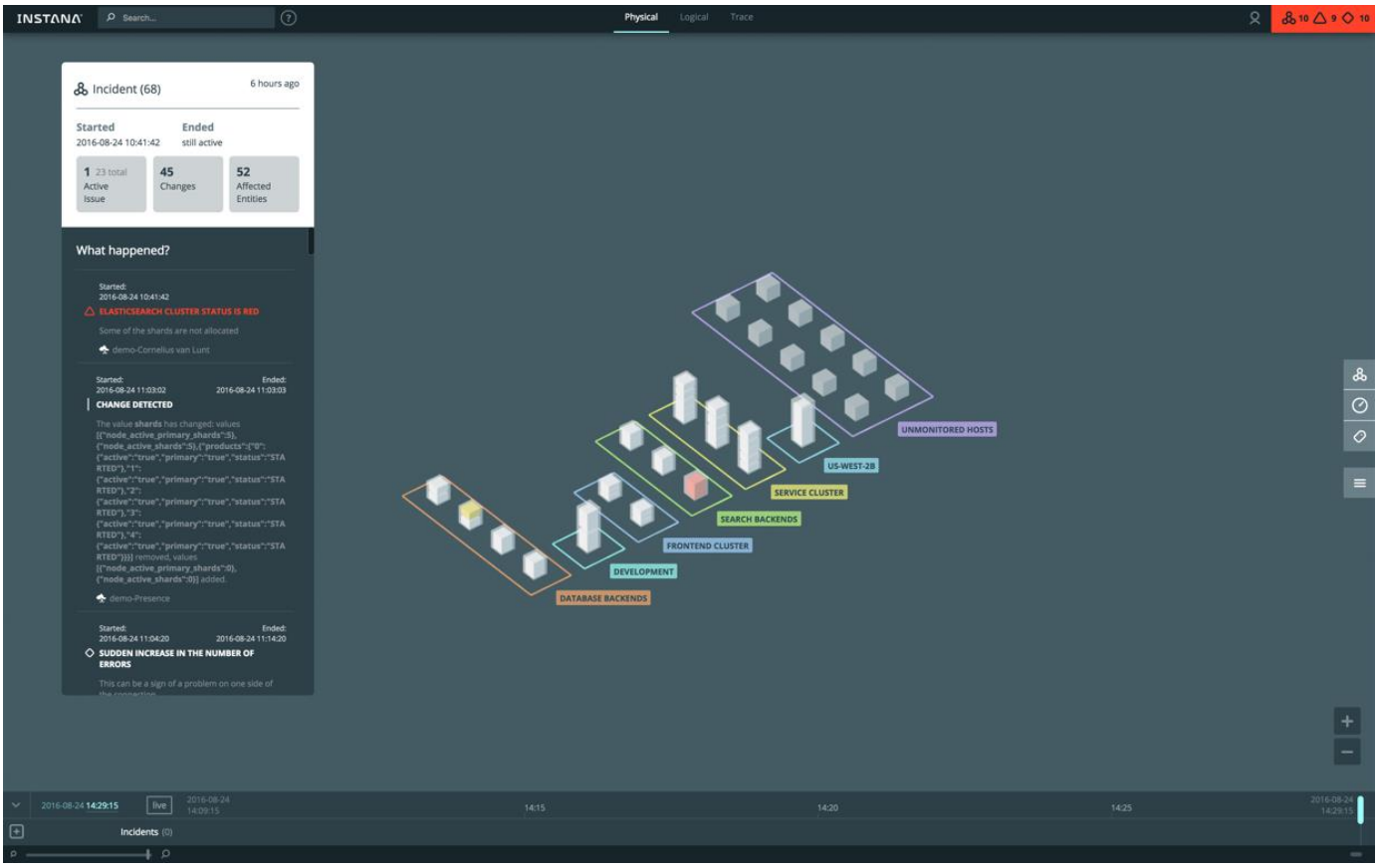


Image Credit: Instana

Instana, a monitoring and management platform for Web-scale applications, was founded in April 2015. One of its key features is Stan, an intelligent virtual robot assistant.

Stan helps Instana users (that is, developers and DevOps) monitor and optimize complex applications and architectures via immediate notifications. Stan has extensive DevOps knowledge built in, and continuously learns about and understands cutting-edge application components and architectures. The robot assistant relies on several technologies including a dynamic dependency graph, automatic discovery and sensoring, and health prediction for components and systems. Instana also includes a real-time knowledge engine that automatically discovers application topology and interdependencies.

Instana uses machine learning, mathematical algorithms, and a proprietary knowledge system to provide dynamic graphs and visualizations. With Instana, the company promises, developers can measure the health of a distributed application (latency, error rate, etc.), understand

service relationships and interdependencies, investigate specific incidents and service failures (real time and historical), and gain a better understanding of the overall application.

Netsil

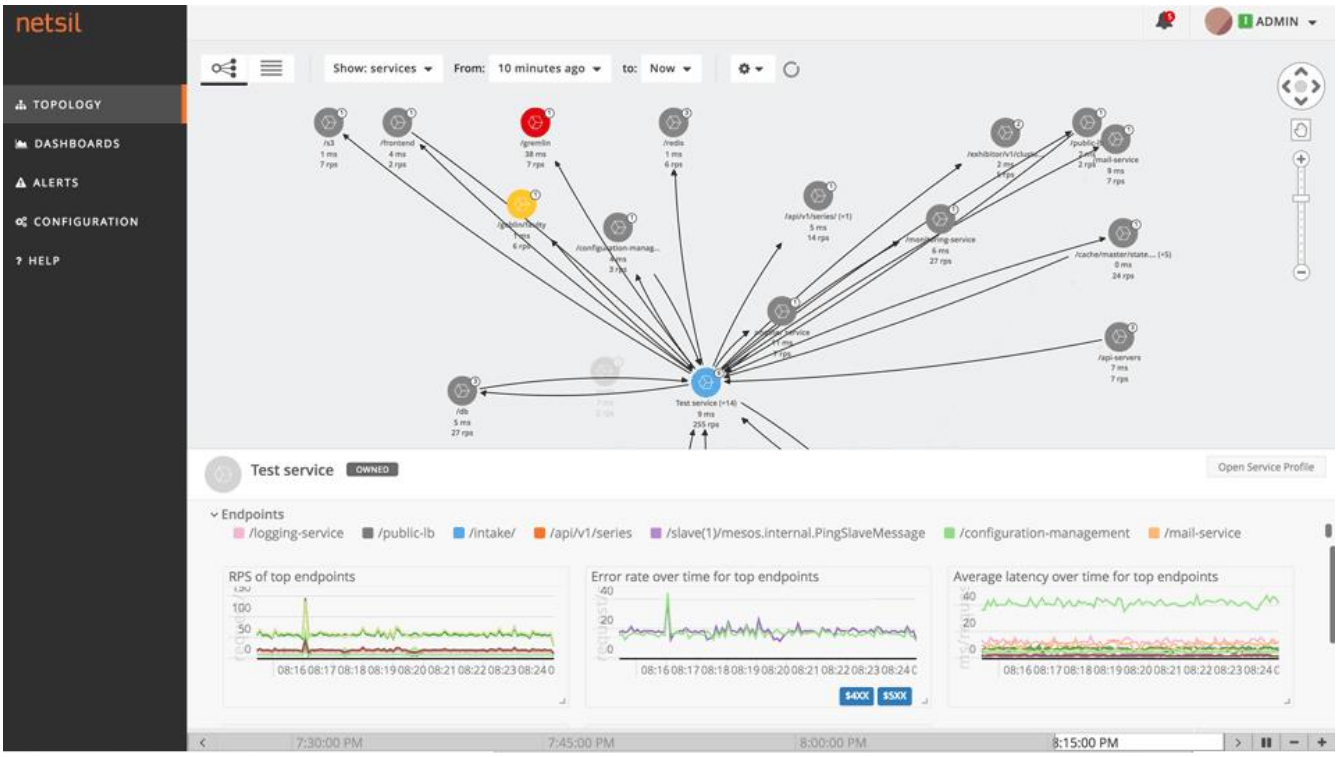


Image Credit: Netsil

Founded in 2016, Netsil's distributed application monitoring and analytics platform automatically discovers complete application topologies, continuously monitors distributed applications, performs distributed tracing, and analyzes application metrics (historical to present).

Microservices-based applications consist of several services, often built using different languages and frameworks. While the services of a distributed application may use numerous languages and frameworks, the protocols of these services are generally the same (REST, HTTP, RPC, pub/sub, etc.). Some APM tools like Netsil can be integrated with these common protocols monitoring services regardless of language or framework.

Netsil monitors and captures distributed application service interaction data to create visualizations that help developers discover and manage incidents, measure an application's overall health, and understand an application's components and dependencies.

OpsClarity

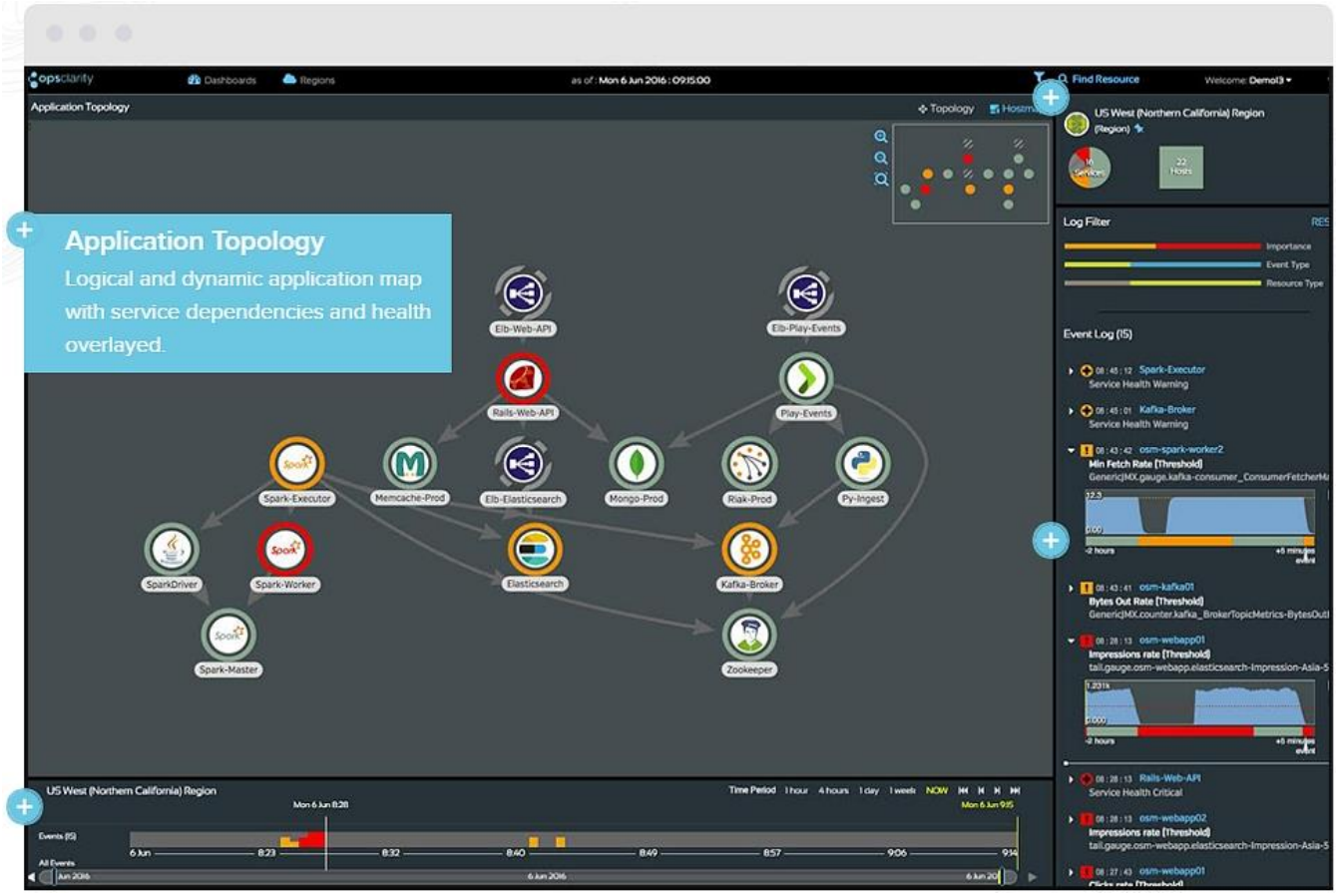


Image Credit: OpsClarity

Launched in December 2015, OpsClarity is an intelligent monitoring and analytics platform for high-velocity Web-scale applications. OpsClarity promises features that include automated topology discovery and metric collection, topology visualization, and performance monitoring.

One of its components is an Operational Knowledge Graph which understands and continuously learns operational data models, service topologies, and other application/system performance baselines. Developers can take advantage of the OpsClarity RESTful API to capture custom metrics, tag annotations for each metric, and push metrics and events. OpsClarity also provides monitoring and analysis tools that show a top-down, consolidated view as well as drill-down data visualizations.

OpsClarity uses AI and graph analysis to visualize and analyze distributed applications at scale. Infrastructure hostmaps show the health of each host or service; topology graphs help developers to understand service dependencies and infrastructure components; and the time-

line feature enables developers to go back and look at previous system statuses to learn how errors and failures occurred.

Common Capabilities

Most of the APM tools highlighted in this article include common capabilities such as automatic discovery of application topology and interdependencies, monitoring of application health, service level alerting, and replay system statuses.

Automatic discovery of application topology and interdependencies saves developers time and reduces the mean time to repair (MTTR). Developers do not have to spend hours trying to figure out service associations and mapping application components. Visualization of application topology can help developers identify and reduce bottlenecks in service dependencies.

Services within a distributed application may have latency issues, errors, and other problems that impact the overall health of the application. Tools that monitor application health and provide service level alerts help developers quickly discover and fix application problems.

Some APM solutions include historical replay capabilities that help developers investigate and determine the root cause of service failures and errors. Historical replay of system statuses can also help developers discover topology changes and gain a better understanding of the overall application.

A New Generation of APM Tools

The number of complex distributed applications continues to skyrocket, creating a need for new types of application monitoring and visualization tools to help the fretting developer, lost in a maze of debugging, figure out, “Why isn’t this working the way it should?!”

Several application monitoring and management platforms are doing their best to give developers insight into these complex, interrelated applications. These APM platforms are leveraging advanced technologies such as artificial intelligence, machine learning, and graph analysis.

Traditional application monitoring and management platforms are no longer enough. As application architectures continue to evolve and become more complex, the tools to monitor, analyze, and manage applications must evolve as well.