IBM Instana

Achieving application health in the microservices age

Contents

01

Introduction

Application health maintenance is most commonly associated with application repair. Degraded health of any container, service or application immediately shifts your focus to repair.

Application repair, commonly known as debugging, has been a crucial process since the earliest days of software development. When programmers have dealt with code that didn't function as intended, it prompted the need for debugging tools such as code profilers. Even today, the process of debugging remains largely the same—it involves finding application functionality that does not perform as expected or meet user needs, and correcting it through careful analysis and remediation.

Every organization that creates software uses debugging to remediate issues. The typical methods are manual and measured as mean time to repair (MTTR). MTTR includes the time to detect (MTTD), notify (MTTN), initiate and validate code repairs. It can be slow and tedious.

Now, new methods of application repair are coming to market using artificial intelligence and machine learning (AI/ML) plus AIOps. This replaces manual methods with automated machine activities to resolve issues faster.

Excitingly, innovative solutions are emerging for addressing application issues, leveraging cutting-edge technologies such as artificial intelligence and machine learning, as well as AIOps. These novel approaches can automate the identification and resolution of operational and functional application issues faster and more accurately than manual processes. Adopting an AI/ML-driven solution is a promising strategy for improving application performance and reliability.

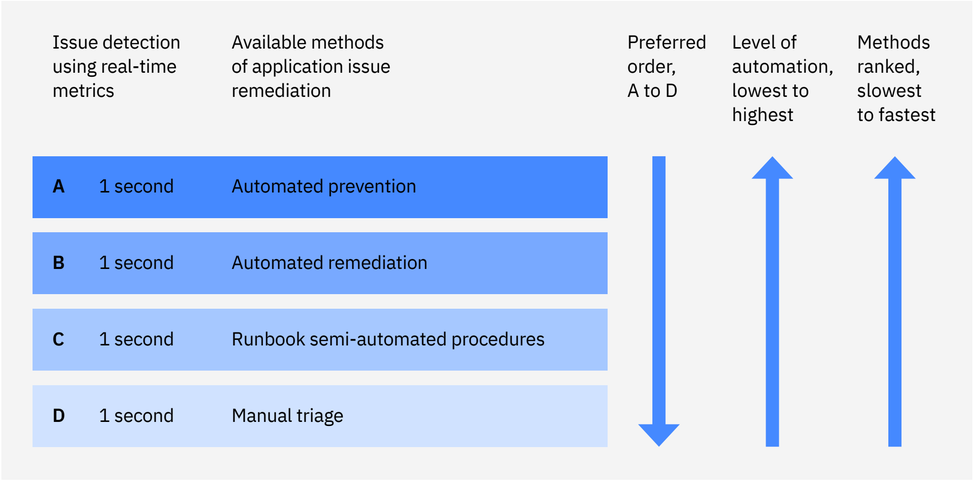
02

Application issue categories: Operational and functional

Operational issues occur when the application components are working properly but infrastructure issues negatively impact application performance. These issues could be from a lack of resources, such as CPU, memory, storage or network bandwidth. But most frequently, they occur during application scaling. These are the type of issues addressed by system reliability engineers (SREs).

Functional issues are typically application code anomalies that affect one or more application components. They can occur in one component or cascade among multiple components throughout the application transaction path. Functional issues almost always require manual triage to resolve code issues. These problems are usually addressed by DevOps teams and developers.

The application issue remediation spectrum



The application issue remediation spectrum includes a range of options for repairing both operational and functional application issues. These options include automatic, semi-automatic or manual repair methods, all simplified and automated with AI and machine learning.

The key to successful and timely issue resolution is precise, real-time metrics and traces, which enable rapid identification and remediation. In the microservices age, slow metrics and trace aggregation can negatively impact Cloud DevOps and SRE initiatives, causing delays or disruptions that impact user experience.

To prevent this, you need an observability platform that measures and aggregates precision metrics and traces with context in real-time. This platform must be able to support both precise, automated operational and functional application issue remediation to ensure application health.

By adopting real-time monitoring and automated remediation, you can prevent issues before they occur to deliver a more reliable, consistent user experience.

04

New and emerging remediation methods

Automated incident prevention—mean time to prevention (MTTP)

Automated incident prevention is used for remediating application resource issues in the underlying application infrastructure. These issues require both an observability platform and an application resource management (ARM) solution.

Mean time to prevention (MTTP) requires the fastest and most precise observability metrics to help achieve real-time remediation with automated ARM. ARM uses AI/ML to evaluate app performance built on baselines you define.

If a performance degradation is detected by the observability platform, the ARM platform is notified. If a threshold is exceeded, ARM automatically implements procedures to fix the problem. Conversely, if the ARM platform detects that application resources are over allocated, it will reduce the allocated resource amount to help lower the cost of cloud-based resources.

The fastest metrics gathering rate is currently one second, allowing ARM engage and potentially provide an MTTP of two seconds or less. Why is this important? Because rapid response to maintain application health has never been more critical than for cloud-based microservices applications. Two seconds versus 10–12 seconds can be the difference between users not noticing a problem and users becoming frustrated.

Intelligent automation is the new imperative. Detecting issues faster is the only way to effectively handle the large number of issues that can occur in such a highly distributed microservices environment. Fully automated MTTP can drive mean time to repair (MTTR) times to near zero. Lowering costs by reducing human resources is why automated remediation continues to become more popular. But it’s only successful if you choose the fastest observability platform plus AI/ML.

Runbook procedures
Runbooks are step-by-step compilations of procedures and operations that are carried out either automatically, semiautomatically or manually to resolve issues. Typically, a runbook contains procedures to begin, stop, supervise and debug a system or application. It may also describe procedures for handling special requests and contingencies. An effective runbook allows operators to manage and troubleshoot a system. Runbooks are also used to help with onboarding new or less experienced DevOps and SRE team members to help them become familiar with existing recovery and resiliency policies and procedures.

With runbook automation, the processes can be performed in a predetermined manner. In addition to automating specific processes, such as resource management and optimization, the runbook results can be presented back to the user for further action. Multiple runbooks can also be linked together using machine learning to provide interactive troubleshooting and guided or automated procedures.

Runbook automation is the process of defining, building, orchestrating, managing and reporting workflows that support system and network operational processes.

A runbook workflow can interact with all types of infrastructure elements, such as applications, databases, containers, endpoints and hardware using a variety of communication methods such as command-line interfaces (CLI), HTTP REST and SOAP APIs, SSH sessions, scripts, utilities and code libraries.

The automated resource management (ARM) capabilities described in the previous section are a specific runbook use case for optimizing distributed system resources.

Automated ML-driven code completion tools

ML-powered code completion tools accelerate application implementation by providing automatic code recommendations based on your input in your integrated development environment (IDE). The tools can generate entire functions and logical code blocks; no more searching for and customizing sample code snippets.

There are two notable recent commercial entrants: GitHub Copilot and Amazon CodeWhisperer. There are also a range of open-source offerings such as Asm-Dude, Atom, Captain Stack, GPT-Code-Clippy, Kite, Second Mate and YouCompleteMe.

Amazon CodeWhisperer is an AI pair programming tool that can autocomplete entire functions using only a comment or a few keystrokes. CodeWhisperer is based on billions of lines of publicly available open-source code, its own codebase and public forum documentation and code. Software engineers choose different code suggestions and autocomplete comments, then accept functions.

GitHub Copilot, from Microsoft, uses AI to assist users by autocompleting code. Like CodeWhisperer, it’s trained on billions of lines of code and turns natural language prompts into coding suggestions across dozens of programming languages. Code completion tools can be applied for both testing and development initiatives to help accelerate the code implementation process. Code completion automation provides code suggestions, but the final decisions are up to the practitioner, to help developers maintain proper code function and security.

Overall, the ML-driven code completion tools help you quickly add new features or improve functionality, usually with better reliability and security. ML-based code completion tools are evolving quickly and will continue to improve.

Manual repair—mean time to repair (MTTR)
Manual code repair has been the gold standard method for application repair and resiliency since the inception of programming languages. It has led to the most common application remediation term—mean time to repair (MTTR).

MTTR is a maintenance metric that measures the average time required to troubleshoot and repair a failed application.

Application MTTR and code triage methods are so commonplace that they've been codified as application lifecycle processes by the International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). The ISO and IEC 14764 divides application maintenance into four categories.

Corrective maintenance
Reactive modification of an application product performed after delivery to correct discovered problems; can be automated with automatic bug fixing

Perfective maintenance
Modification of an application product after delivery to improve performance or maintainability

Adaptive maintenance
Modification of an application product performed after delivery to keep a software product usable in a changed or changing environment

Preventive maintenance
Modification of an application product after delivery to detect and correct latent faults in the application product before they become effective faults

05

Application maintenance tips and tricks

01

Build a solid application issue remediation plan that addresses known issues before they can impact users. Application maintenance and reliability are more complicated in today’s containerized microservices era than ever before, so create a comprehensive plan.

02

Use the method or methods that best suit your needs. You’re no longer constrained by a one-size-fits-all manual repair strategy; now have access to a broader range of automated options that can help you make repairs faster or even prevent issues.

03

Verify your application repairs and maintenance in preproduction before you release new or updated software into production. This helps avoid unwelcome surprises.

04

Use application remediation automation wherever you can to help ensure peak health for your applications and infrastructure. Automated issue response, code completion and other techniques will only be invoked when you choose to do so.

06

The IBM Instana application health advantage

The IBM Instana® platform provides critical observability functionality to help ensure application health and resilience. IBM Instana delivers one-second metrics and retains them for up to 24 hours. These high precision capabilities can instantly drive industry-leading, unparalleled operational and functional issue remediation. Key IBM Instana attributes include:

AI driven: AI is an integral part of the IBM Instana observability platform providing advanced capabilities such as smart alerts, trace context, unbounded analytics and continuous automated discovery. It helps simplify MTTR and engage AIOps to automatically or semiautomatically remediate operational and functional app issues.

Precise one-second metrics: These metrics provide an industry-leading mean time to detection of one second and a mean time to notification of three seconds. IBM Instana retains these metrics at that granularity for up to 24 hours. This helps ensure measurements that deliver a mean time to prevention (MTTP) to prevent delays in automated ARM and runbook procedures.

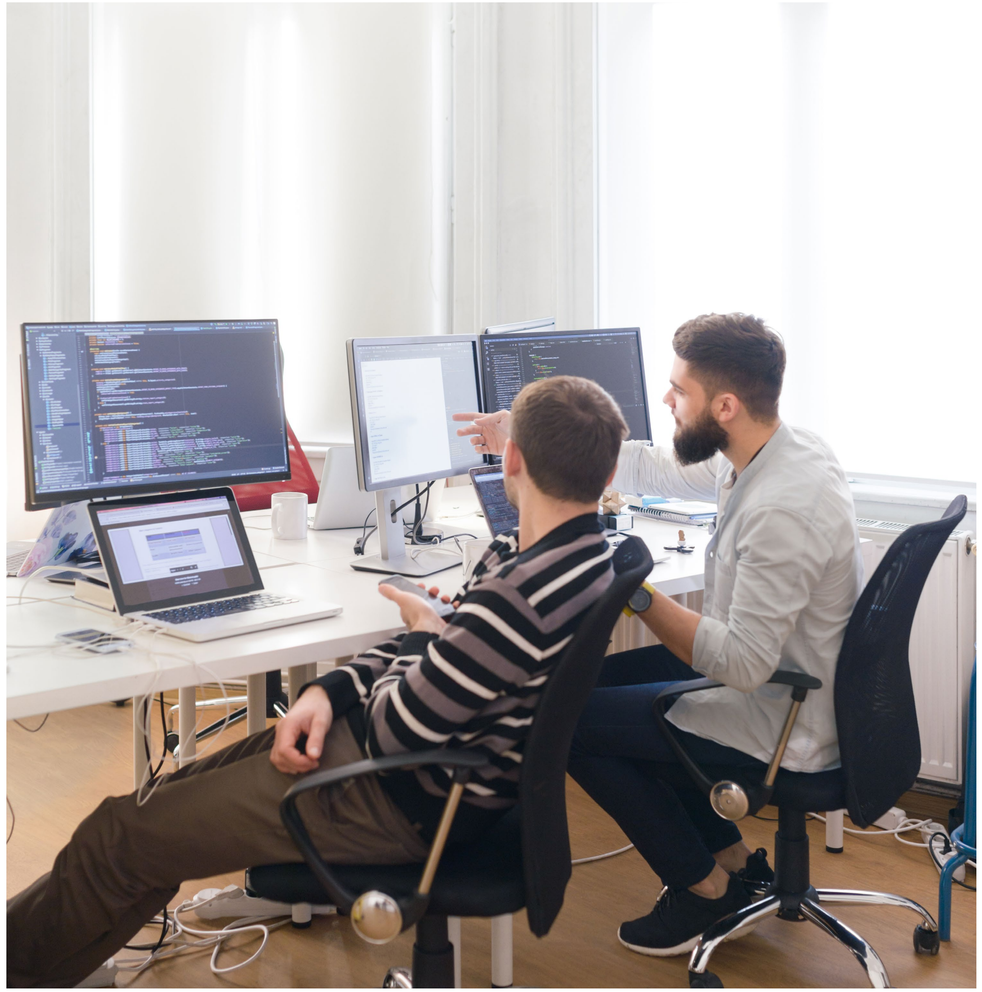
Full end-to-end transaction traces: IBM Instana maps every end-to-end trace without sampling for every transaction. This capability means no gaps caused by sampling. This operational speed shows upstream and downstream dependencies in real time to help quickly pinpoint the root cause of issues.

Automated discovery: IBM Instana automatically discovers every application and infrastructure element the moment it’s installed. The platform then instantly collects application components, nodes, containers, and architectural entity metrics and traces.

Automated context: IBM Instana provides a context guide driven by the continuously updated dynamic graph. It tracks the components of your infrastructure, associates them and visualizes them with their logical counterparts. It features an upstream or downstream button that lets you quickly navigate to the dependencies of an application, service, endpoint, or infrastructure or Kubernetes entity.

Architecture monitoring: By monitoring your architecture as well as your applications, IBM Instana gives you a better view into the impact of your applications on your architectural components The architecture monitoring included in IBM Instana provides the precision metrics to support automatic ARM and rapid MTTR.

IBM Instana is the first observability platform that offers precise, automated issue remediation for both operational and functional application problems in real time. The platform’s continuously updated interface helps you produce the fastest automated remediation and MTTR.



07

Is IBM Instana right for you?

IBM Instana provides an industry-leading real-time automated enterprise observability platform. Its application performance monitoring capabilities are ideal for organizations operating complex, modern, cloud-native applications. IBM Instana is ready to go to work anywhere your workloads run—in public clouds, private clouds, hybrid clouds, on mobile devices, on premises or in an IBM zSystems® environment.

IBM Instana gives you expanded control over modern hybrid applications, thanks to its precise metrics, full end-to-end traces for all transactions and AI-powered contextual dependencies discovery inside hybrid applications. For systems reliability engineers, IBM Instana helps improve the reliability and resilience of cloud-native applications by preventing issues from turning into incidents. And by providing blazing fast remediation times when incidents do occur.

See the power of IBM Instana for yourself. Sign up today for a free 14-day trial of the full version of the product. No credit card required.

IBM Instana free trial →

Explore IBM Instana →

IBM Corporation
New Orchard Road
Armonk, NY 10504

Produced in the United States of America
May 2023