✕ StormForge

The Next Frontier for AIOps: Application Optimization

Managing Application Performance and Costs Inside Kubernetes Containers

Artificial Intelligence for IT Operations (AIOps) holds great promise for enterprise IT teams. Intelligently automating formerly manual IT tasks, such as resource provisioning, and monitoring and managing network and application performance, are game-changers for these teams.

Market data on enterprise adoption of AIOps shows that its promise is turning into reality. For example, in a recent report from Forrester, 51% of the respondents said that their organizations are already using AI- or machine learningbased systems, and another 21% of them said they have deployments planned within the coming year. Spending seems to bear this out. According to a 2020 forecast from Research and Markets, the market for AIOps platforms will exceed $9.2 billion by 2025.

Current Focus – Solving Network and Infrastructure Problems

Judging by analyst reports, news coverage and other sources, virtually all of AIOps' market traction has been gained at the network or IT infrastructure level. Surveys, reports, and articles center on AIOps use cases that include:

- Event noise reduction
- Root cause analysis
- Change management
- Resource utilization analysis
- Storage management

These use cases are all centered on tackling stubborn old network and infrastructure problems and challenges that still need to be solved and managed. But AIOps, with its machine learning (ML)-driven automation, is capable of much more.

The Next Phase of AIOps Growth — Moving Up the Stack

To date, there's been little discussion of AIOps use cases at the application level. The network-level use cases are obviously important because applications depend on networks. But applications are the tools with which users conduct business and generate revenue and other forms

of business value, and competitive advantages for their organizations. Being userfacing, applications create the user experience that IT teams must manage. So, managing applications in deployment consumes a lot of IT teams' time and budget dollars. Despite this reality, very little attention has been paid to how AIOps can improve operations at the application level.

That's about to change, however, because automated optimization of application performance is where AIOps is going next. In fact, it has already arrived. More specifically, machine learning (ML), one of the foundational technologies for AIOps, is already being used to enhance the performance of containerized applications. It's also being used to give IT, NetOps, and DevOps greater control over the cost of microservices-based apps, especially those running in containers in Kubernetes environments.

K8s deployment issues crop here
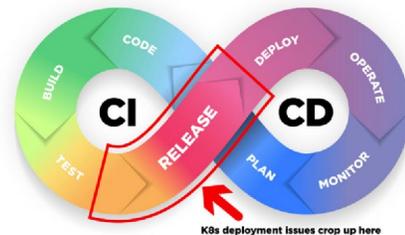
New Use Cases Enabled with Existing Tools and Processes

How will these new, application-level use cases arise within already well-defined enterprise IT and DevOps processes? One likely place is within the Continuous Integration and Continuous Delivery (CI/CD) framework that many enterprise DevOps, NetOps, and IT teams are already using.

The CI/CD pipeline is a well-known process used by developers, testers, and deployment teams. It is designed to enable teams to create, implement, and deliver applications and code changes faster, more reliably, and with higher quality.

The continuous integration part of the framework involves developers making small changes rapidly, and frequently checking their work into code repositories. Once there, the changes are integrated and validated. The continuous delivery side of the model features automated delivery of applications and code changes from development and testing phases into production environments.

A challenge that many teams encounter when using the CI/CD pipeline for Kubernetes applications are miscues and failures that arise between the testing phase and release phases, and after releases. This gap in the CI/CD framework is an ideal location for automated optimization of application performance.

How AIOps Bridges Gaps and Smooths Rough Spots in CI/CD

Development teams commonly test code prior to release, but the traditional testing approach focuses on evaluating whether individual snippets or functions work well. Once those snippets are shown to work well, they are put into containers, which are in turn individually tested.

The complexity arises when deployed applications encounter real-world scenarios that testing didn't anticipate. Or, in container environments, the application may stumble when the orchestration engine shifts the way containers are organized, or as the availability of system resources like memory change. While each individual piece of the system works, when they operate together, dynamically, over time, issues often arise that are outside of typical test parameters.

Teams that try to manually test, controlling for numerous parameters, typically find the process is so difficult that they skip it and risk impacting applications in production. AIOps tools allow application development and operations teams to thoroughly and efficiently test whether their applications are ready to meet the demands of real-world, containerized use.

This new ability has numerous real-world uses, such as adding performance boosts and stability to applications running in multi-tenant clusters, where resources are shared. Rather than running many small clusters to support individual applications, large enterprises and mature IT organizations typically run a small number large clusters supporting disparate applications. These may include the company's websites, mobile applications and customer-facing applications, as well as internal applications like company databases. Kubernetes may allocate resources in a manner that works well for their use initially, but over time, as need for resources such as memory shifts, applications no longer coexist harmoniously. AIOps can solve the "noisy neighbor" problem by taking into account the ongoing needs of every application running on a cluster to make sure they're not coming into conflict with one another.

AIOps can also be used to inform architecture planning on clusters, allowing teams to proactively distribute their software in a manner before the needs of one begin impacting others.

Large organizations that may not rely on public cloud providers – and therefore can't readily add more nodes to their environment – use AIOps tools to better manage resources in their own data centers. As demands on an application grow, but the capacity of available systems does not, the operations team must scale down the resources each application requires, but without crashing the applications. With AIOps, they can understand the tradeoffs required to give each application the minimum resources required without allowing performance to drop below a predetermined level.

Conclusion

Although AIOps usage among enterprise IT teams is still in its early phases, market watchers are expecting very rapid growth over the next few years. It's a good bet that much of that adoption will be tackling use cases at the network and infrastructure level. Of course, when lots of companies make essentially the same moves, differentiation and competitive advantages become harder to win and hold.

This isn't an argument against using AIOps to solve complicated and stubborn network problems. Far from it; companies should do that to keep their operations up to par. By taking AIOps to the application level, however, enterprises will not only keep pace with their competitors, they'll have the chance to take the lead with innovative, efficient, and cost-effective IT operations. By pointing AIOps at applications running inside Kubernetes containers, enterprises will truly stand out.

Contact us to request a free trial of StormForge at **stormforge.io**