Why AIOps?
Application performance

Learn how to optimize your applications
to drive business value

Table of contents

Why application performance?

The modern application era
In the modern application era, where upwards of two-thirds of all gross domestic product (GDP) is now digital,[1] application performance is singularly the highest priority for CIOs because the application *is* the business. Delivering applications is the primary reason IT exists. The CIO, who heads IT, has no choice but to deliver application performance to assure the business is never constrained by IT. In fact, the CIO is considered a failure when not delivering application performance; ironically, it's very reasonable for CIOs to exceed their planned budgets. In short, failing to assure application performance damages the business. What's challenging is the applications that congest first and longest are the applications with the greatest surges in demand—often the most valuable apps.

The modern application hosting service
Application development investment vastly exceeds the cost of application hosting. With this relative investment, companies knowingly and willingly overprovision their infrastructure and cloud environments to mitigate application performance risk. Infrastructure resources, in both data centers and cloud, are fast becoming disposable commodities and, therefore, a less-reliable return on investment (ROI) justification. Additionally, poor application performance creates mistrust between line of business (LOB) application owners and the ITOps and CloudOps teams that deliver the application hosting service. Think of the LOB investing millions of dollars with hundreds of people developing new user experiences, putting their reputation on the line—only to have Ops teams deliver end users the slow "loading wheel" experience. All that investment in enhancing the end-user experience is lost.

The growing pressure point for application performance
Application starvation occurs when the infrastructure—on premises or in the cloud—can't service the demand of the application and its end users. Infrastructure resource starvation is the most frequent cause of application performance degradation. By contrast, application code architecture, monitored with application performance management (APM) tools, is rarely less than 10% of the source of application performance degradation in production environments.

Moreover, given application development focus on shipping quality code by using enhanced processes, such as continuous integration continuous delivery (CICD), quality assurance (QA), preproduction and staging, code quality impacting performance is becoming increasingly rare.

Effective infrastructure resource sharing requires prioritization
In the on-premises data center, applications starve when the demand for shared infrastructure resources remains nonprioritized. Given this limitation, application workloads are typically overprovisioned and placed without sufficient understanding and context for the available resources. When sharing resources, all applications take from a common resource pool regardless of utilization. Overprovisioned and mis-sized resource allocation results in consistent congestion, leading to service-level agreement (SLA) violations, inefficient manual troubleshooting, perpetual resource adjustments and, as previously noted, unleveraged investment in application development.

Today's reactive and single-resource monitoring tools don't understand the relationship between applications and infrastructure, and, therefore, rely on manual interpretation and intervention to resolve resource congestion. The more time it takes to render a resolution, the more likely the decision on what to do will be obsolete.

An alternative is to continually use dynamic application demand to prioritize the allocation of the full stack of shared infrastructure resources. As an application's demand curve ascends, its relative priority to obtain and preserve resources increases. As that demand curve descends, its relative priority to obtain and retain resources decreases. This continuous reprioritization of the full stack of shared resources minimizes starvation, enables fluid resource sharing across all applications and assures application performance.

Effective public cloud resourcing requires knowledge of the application

In the public cloud, applications starve when instances—resources—remain insufficient to service the application's demand. To address this risk, cloud administrators often overprovision instances. In both cases, incorrect instance provisioning results from the cloud administrator's limited knowledge of the application demand's resource requirements. Further, developers who focus on building the application are typically disinterested in profiling and forecasting its resource requirements. As a result, resource guesstimates are made while the cloud provider places the risk, burden and consequences of that guesstimate on the customer.

Compounding the complexity, the right instance decision is increasingly challenging as a single Amazon Elastic Compute Cloud (Amazon EC2) instance provisioning has millions of configuration options when deciding the resource type, underlying hardware, sizing, geographies, pricing, reservations, savings plans and so on.

Additionally, dynamic changes in application demand require continuous re-evaluation of this guesswork used to select cloud instances. Finally, managing the elasticity of public cloud application instances requires real-time, continuous calibration—especially with the adoption of short-lived, container-based applications. As a result, most public cloud instance allocations are overprovisioned, and managed manually and reactively as a hedge to performance risks and with a complete lack of understanding of application resource demand. In addition, monitoring-based cloud provider and cloud management platform tools lack the understanding of the application's demand to assure application performance in the cloud. They're restricted to cost visibility, historical billing and departmental cost allocation—nothing to do with assuring the performance of applications running in a public cloud.

Why IBM Turbonomic Application Resource Management?

Turbonomic, an IBM Company, provides the underpinning of the modern application hosting service by uniquely using an understanding of application demand to provide continuous application resourcing actions and performance analytics to help assure application performance in real time, over time, 24x7x365.

IBM Turbonomic Application Resource Management for uses a common data model so customers can confidently resource their applications today, as well as the modern applications of the future, regardless if they run on premises, in public clouds or on the edge.

Application resource management (ARM) delivers performance and savings

IBM Turbonomic's ARM, using AI-driven analytics, helps assure application performance by continuously matching real-time application demand requirements with resource types and sizes. Resourcing actions include starting and stopping, initial and continuous placement, scaling, and resizing.

By contrast, hundreds of tools—and even spreadsheets—purport to save customers money, but their "recommendations" are often based on simple threshold alerts that can cause performance problems. A recent survey reported that 39% of financial operations (FinOps) professionals marked "getting engineers to take action" as their top challenge.[2] The reason? lack of trust in the existing tools used to generate these "actions."

The new requirements for assuring the performance of modern applications

The IBM Turbonomic AIOps platform uses a comprehensive common data model that ingests, normalizes and manages all shared resources on which application performance depends. Most importantly, it builds the supply chain relationship topology—"stitching"—between each resourcing dependency, from application to infrastructure.

By contrast, fragmented and manual tools can't assure performance because they monitor resources in isolation and have limited—if any—application perspective. Application performance requires understanding and managing all resource dependencies in precisely the right amounts, order and time frame.

Conclusion

In the modern application era, the application is the business. As applications become more complex, with more dependencies, and environments more diverse and distributed, the risk to application performance and user experience is growing exponentially. The only way to address these challenges is to implement an application-focused approach that continuously evaluates the applications' demand, the supply of the available resources and generates actionable recommendations that both operations and application owners can trust. Over time, as the confidence builds, the next stage is to automate these actions. The result will be highly performant applications, massive IT spend reduction and the opportunity to unleash business innovation. That's the magic of AIOps.

Upwards of two-thirds of global GDP is now digital.[1]

- Delivering applications is the primary reason IT exists.
- CIOs must assure the business advancement is never constrained by IT.
- Application development is 3 times the cost of application hosting.
- Companies overprovision to mitigate the risk of application performance degradation.
- Infrastructure resource starvation is the most frequent source of application performance degradation.
- Applications starve when demand for shared infrastructure resources isn't prioritized.
- Today's resource-focused monitoring tools, without understanding application demand, can only react to negative situations.
- Continuous reprioritization of the full stack of shared resources, based on application demand, is the only way to assure application performance.

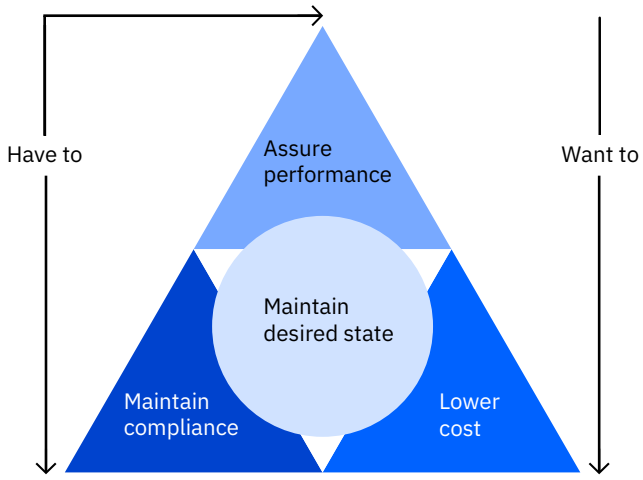

Figure 1. Applications are in desired state where performance is assured while maintaining compliance at the lowest cost.

- Cloud administrators have limited knowledge of an application's resource requirements.
- Application developers are focused on business logic and defer resourcing decisions to IT staff.
- Selecting the right public cloud resources is complex with millions of configuration choices.
- Most cloud instance selections are overprovisioned to minimize the risk of performance degradation.
- Rightsizing public cloud instances requires knowledge of application demand.

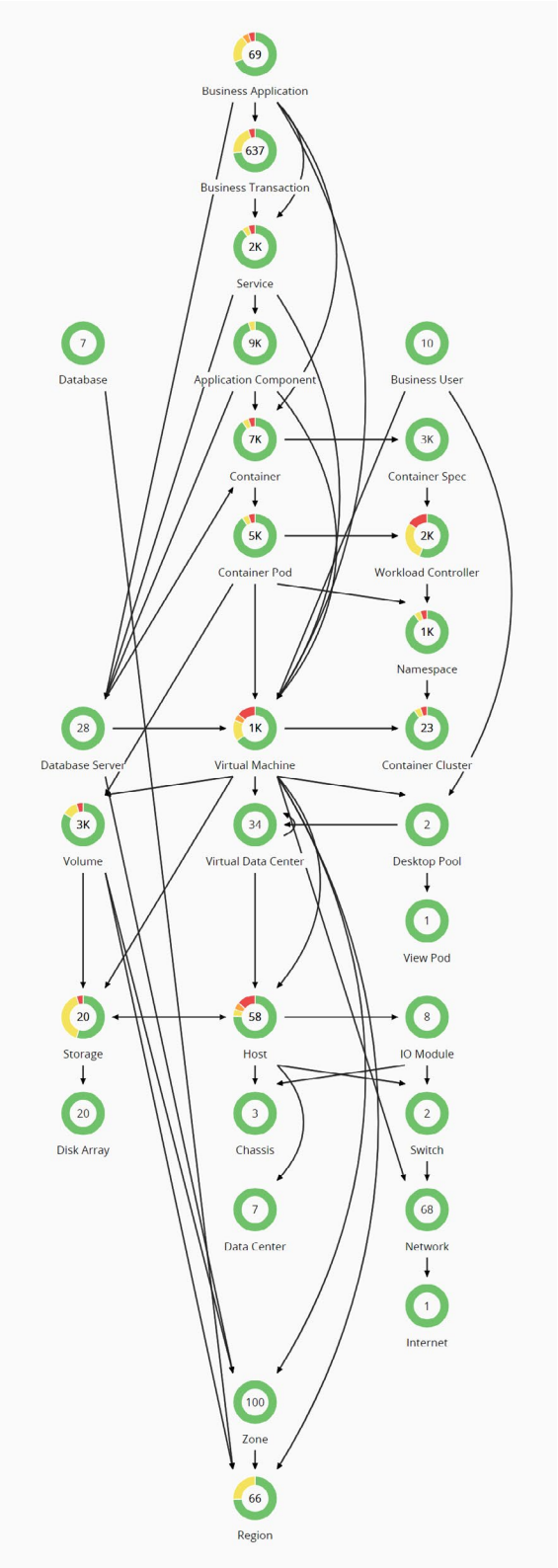

Figure 2. IBM Turbonomic automatically builds the supply chain relationship topology

About IBM Turbonomic, an IBM Company

IBM Turbonomic, an IBM Company, provides application resource management (ARM) software used by clients to help assure application performance and governance by dynamically resourcing applications across hybrid and multicloud environments. IBM Turbonomic network performance management (NPM) provides modern monitoring and analytics solutions to help assure continuous network performance at scale across multivendor networks for enterprises, carriers and managed services providers.

For further information, visit ibm.com/cloud/turbonomic.

IBM Corporation
New Orchard Road
Armonk, NY 10504

Produced in the United States of America
March 2022

[1] IDC Reveals 2021 Worldwide Digital Transformation Predictions, IDC, 29 October 2020.
[2] State of FinOps Report 2021, The FinOps Foundation, 2021.