turbonomic
an IBM Company

WHITEPAPER

Why AIOps? Application Performance

www.turbonomic.com

Upwards of two-thirds of global GDP is now digital

- *Delivering applications is the primary reason IT exists*
- *CIO must assure the business advancement is never constrained by IT*

- *Application Development is **3x** the cost of Application Hosting*
- *Companies overprovision to mitigate the risk of application performance degradation*

WHY APPLICATION PERFORMANCE?

The Modern Application Era

In the Modern Application Era, where upwards of two-thirds of all GDP is now digital, 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 is very reasonable for a CIO to exceed their planned budget. 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 (oftentimes the most valuable apps).

The Modern Application Hosting Service

Application development investment vastly exceeds the cost of application-hosting (>3:1). With this relative investment, companies knowingly and willingly over-provision 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 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 end user experience is lost.

The Growing Pressure Point For Application Performance

Application starvation occurs when the infrastructure (on premises or cloud) cannot 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 APM tools, is rarely (<10%) the source of application performance degradation in production environments. Moreover, given application development focus on shipping quality code leveraging enhanced processes—CI/CD, QA, Pre-Prod and staging—code quality impacting performance is becoming more and more rare.

- *Infrastructure resource starvation is the most frequent source of application performance degradation*

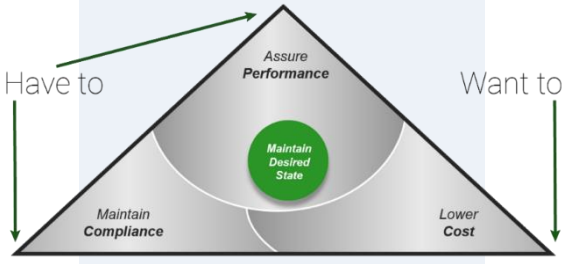
Effective Infrastructure Resource Sharing Requires Prioritization

In the on-premises data center, applications starve when the demand for shared infrastructure resources remains non-prioritized. Given this limitation, application workloads are typically over-provisioned 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. Over-provisioned and mis-sized resource allocation results in consistent congestion leading to SLA violations, inefficient manual troubleshooting, perpetual resource adjustments, and as noted above, unleveraged investment in application development.

Today's reactive and single resource monitoring tools do not 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 leverage 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.

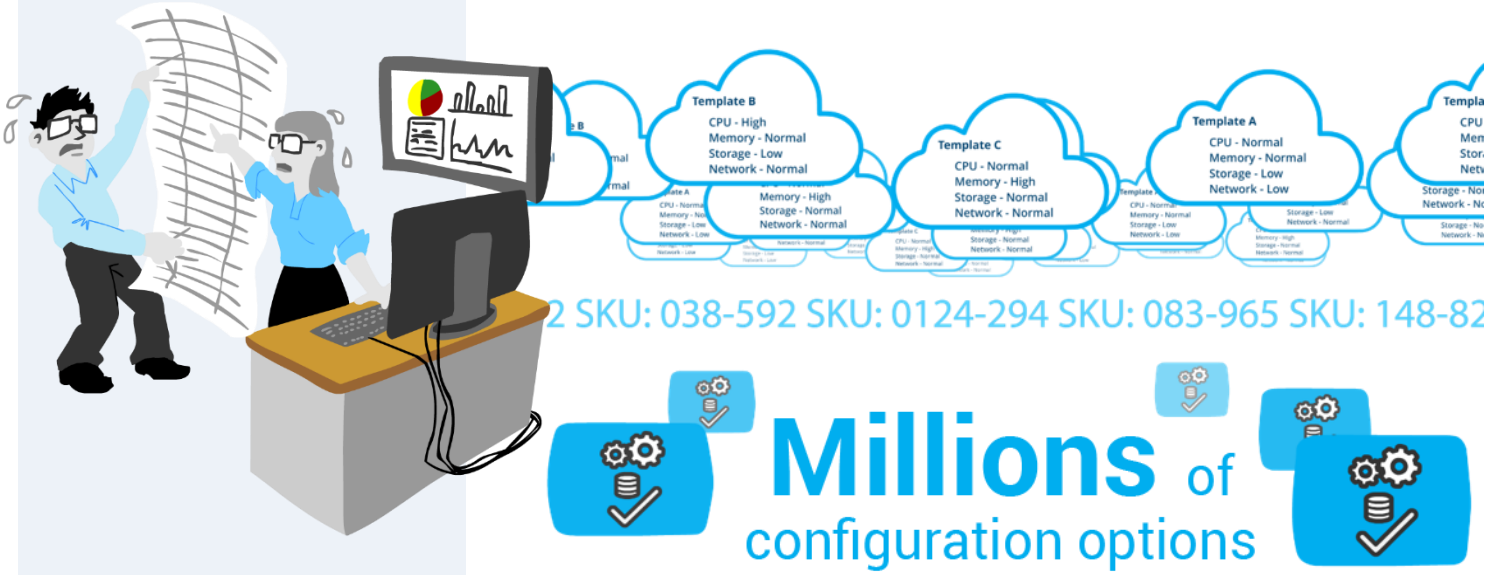
- *Applications starve when demand for shared infrastructure resources is not prioritized*
- *Today's resource-focused monitoring tools, without understanding application demand, can only react to negative situations*
- *Continuous reprioritization of the full stack of shared resources, based on application demand, is the only way to assure application performance*



Effective Public Cloud Resourcing Requires Knowledge Of The Application

- *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*

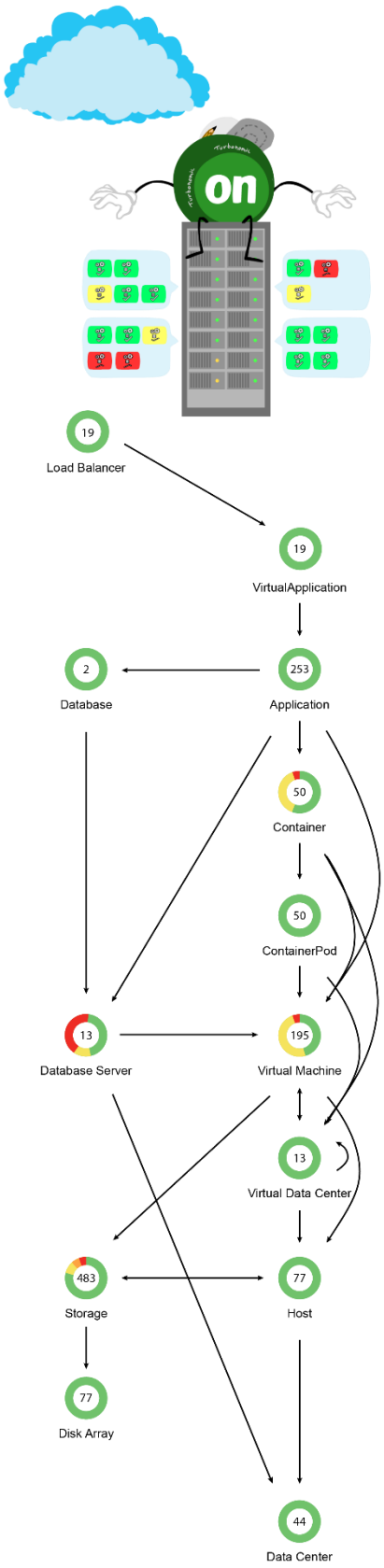
In the public cloud, applications starve when instances (resources) remain insufficient to service the application's demand. To address this risk, cloud administrators often over-provision instances. In both cases, incorrect instance provisioning results from the cloud administrator's limited knowledge 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.



- *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*

Compounding the complexity, the right instance decision is increasingly challenging as a single EC2 cloud instance provisioning has million of configuration options when deciding the resource type, underlying hardware, sizing, geographies, pricing, reservations, savings plans, etc.

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 over-provisioned, 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 are restricted to cost visibility, historical billing, and departmental cost allocation—nothing to do with assuring the performance of applications running in a public cloud.



Turbonomic automatically builds the supply chain relationship topology

Why Turbonomic?

Turbonomic provides the underpinning of the Modern Application Hosting Service by uniquely leveraging an understanding of application demand to provide continuous application resourcing actions and performance analytics to assure application performance in real-time, over time, 24/7/365.

Turbonomic AIOps platform leverage a common data model so customers can confidently resource their applications today as well as the modern applications of the future regardless if the run, on premises, on public clouds or on the edge.

Application Resource Management Delivers Performance and Savings

Turbonomic's Application Resource Management (ARM), **leveraging AI-driven analytics**, assures 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 re-sizing.**

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 FinOps professionals marked "Getting Engineers to take action" as their top challenge**. The reason? lack of trust in the existing tools used to generate these "actions."

The New Requirements For Assuring The Performance of Modern Applications

Our 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 cannot assure performance because **they monitor resources in isolation and have limited (if any) application perspectiv**e. 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!

About Turbonomic, an IBM Company

Turbonomic, an IBM Company, provides Application Resource Management (ARM) software used by customers to assure application performance* and governance by dynamically resourcing applications across hybrid and multicloud environments. 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, please visit www.turbonomic.com

*www.turbonomic.com/resources/case-studies