AKAMAS

Autonomous Optimization for Java Applications

Java™

Java performance tuning is critical

Java applications are the core of many online services. Their availability, performance and resilience is critical to ensure the required levels of service and user experience.

However, even after weeks and months of tuning efforts by Java performance experts, Java applications may still show **unacceptable throughput and response time** and **huge resource utilization**.

This translates into **reduced operational efficiency**, **low business agility**, **unnecessary costs** in terms of software licenses, infrastructure and cloud, and **missed competitive advantages**.

55% of Java Developers mention "long application response time" as the most common issue
Source: JRebel

2M Time spent to optimize a single mission-critical Java microservice out of 200 in production
Source: Leading booking service

40% of Java Developers mention "large memory requirements" as top Java challenge
Source: Jakarta EE

JVM performance tuning challenges

The complexity, resource greediness and unpredictable bottlenecks of the JVM technology make **Java performance tuning** a daunting task, even for the best Java performance experts.

Any **manual, trial-and-error approach** is bound to fail to identify the best JVM configuration that ensure maximum performance and resilience at lowest cost for any given application.

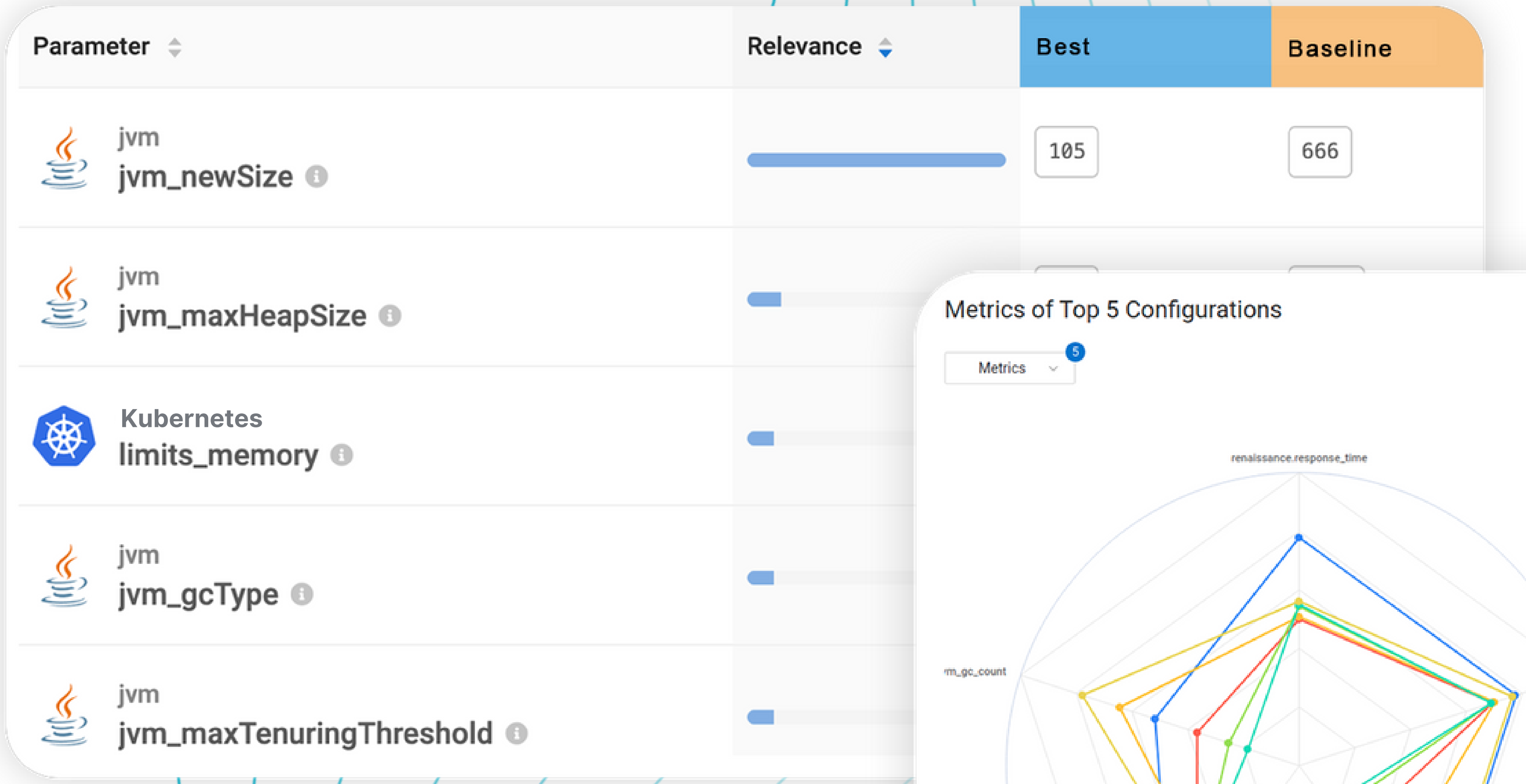
The adoption of **DevOps practices** and **microservice** paradigms further exacerbates this situation, as increasing the problem complexity and shortening the allowed tuning time.

JVM 800+ tunable flags impact performance and resource usage depending on the application and workload - and JVM version

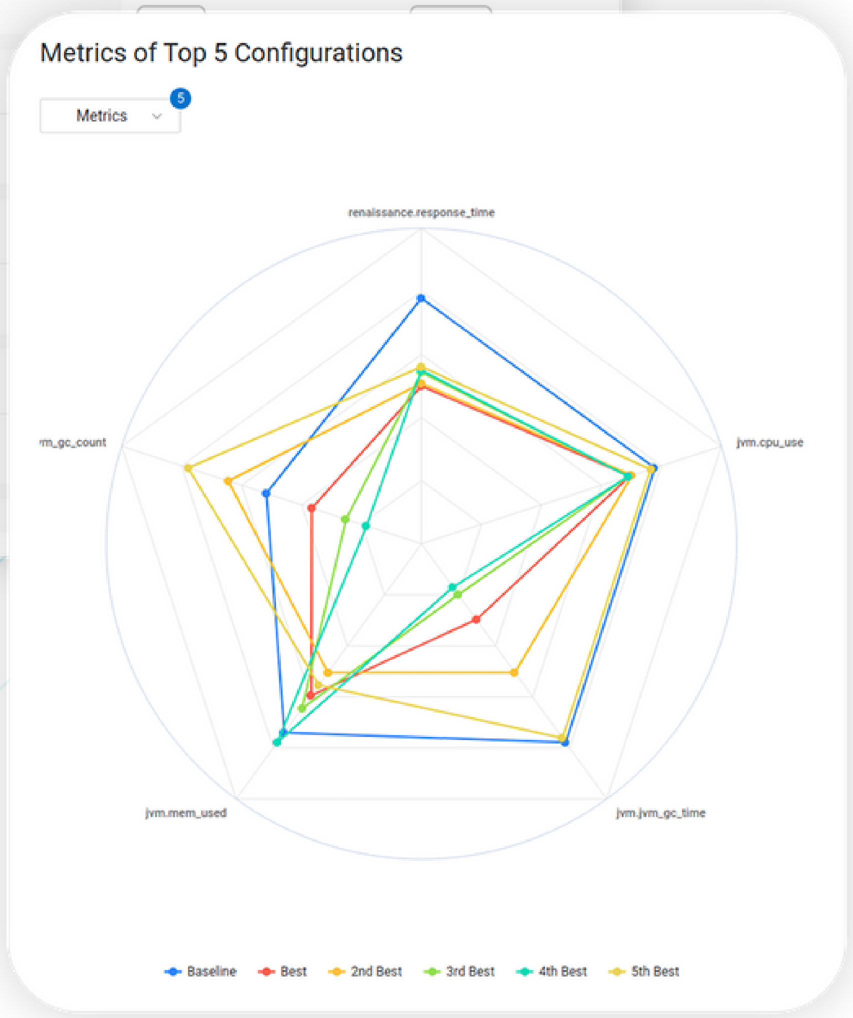
JVM default values are rarely aligned to your performance or efficiency goals and need to be tuned for each given application and workload

industry/vendor best practices only provide general guidance and may actually cause service slowdowns and user experience degradation

Autonomous Performance Optimization



Parameter	Relevance	Best	Baseline
jvm jvm_newSize		105	666
jvm jvm_maxHeapSize			
Kubernetes limits_memory			
jvm jvm_gcType			
jvm jvm_maxTenuringThreshold			



Akamas **ML-powered optimization** automatically identifies the best configurations with respect to **custom goals & constraints** defined in terms of performance, resilience, cost and SLOs for each specific application.

Akamas **full-stack optimization** can be applied to **any Java applications**, whether monolithic or microservices, and to **any IT components**, including containers, databases, middleware and cloud instances.

AKAMAS

Akamas benefits for Java applications

Akamas autonomous optimization guarantee the **best levels of performance and resilience** at all times, release after release of your Java applications, while also achieving the **best cost efficiency**, by avoiding infrastructure and cloud over-provisioning and associated cost.
With Akamas, **developers**, **performance engineering & SREs** are freed from any manual tuning tasks and receive useful insights on how to best tame the complexity of modern IT stacks and contenterized microservice applications, while also aligning with continuous delivery pipelines.
This results into **higher operational efficiency**, **business agility** and **competitive advantage**.

+24% increase in transactions/seconds with the same infrastructure resources

-50% decrease in memory footprint with the same application performance

-80% savings in engineering time spent for manual tuning tasks

ΛKΛMΛS

AKAMAS

info@akamas.io