ORACLE

Oracle Fusion Applications reduces OPEX and boosts performance with Oracle Java EPP

Case study

Public

ORACLE

Summary

Oracle Java Enterprise Performance Pack helps Oracle Fusion Applications improve application response times by 40% and decrease CPU utilization by 25%.

Business Context

Oracle Fusion Applications serve a massive customer base, from small to medium size enterprises to major Fortune 100 companies, and their customers have diverse and complex needs. Fusion Applications have dozens of feature updates each quarter and support over 14,000 customers across healthcare, gaming, industrial manufacturing, finance, and telecommunications.

The Oracle Fusion team needed to improve their offerings while maintaining the speed of innovation; they could , only work with existing financial and engineering resources. Moreover, due to a number of external dependencies and legacy code, maintaining compatibility with application data and other services was critical to success.

Technical Challenge

To meet critical customer needs, the Fusion team had to find a solution to meet specific demands of performance, memory improvements, cost savings, and more. They had to find a solution without requiring an entire codebase re-write or application re-factor, while at the same time keeping the applications coupled with the proper data in the cloud.

Additional IT challenges like compliance, version management, and security baselines were already complex To offer a range of options (on-premise vs cloud) while improving performance seemed too challenging to tackle with operational expenditures (OPEX) at its limit and resources already bound to other priorities.

Solution

The Oracle Java team recognized that the Java Enterprise Performance Pack (Java EPP) could help Oracle Fusion Applications. Java EPP makes the significant memory management and performance improvements brought to Java Development Kit (JDK) 17 to Java Development Kit (JDK) 8. These include modern garbage collection algorithms, compact strings, enhanced observability, and dozens of other optimizations. The Fusion team quickly realized that EPP could meet their needs and requirements: simple procedures, no impacts on compatibility, and an increase in memory and CPU savings.

> "We did the Java EPP rollout gradually and had no issues. Over the period of the last year, we have monitored the performance, and the testing was validated – memory and processor gains have checked out and have been stable."

Rahul Sarin
Oracle

Using Java EPP, the Oracle Fusion team was able to keep the design patterns and code they had previously created and realize significant gains in performance both in compute and memory usage. Fusion customers with external dependencies are unaffected. Customers did benefit from faster response times in the hosted applications and more cost-effective application

optimization on specific hardware. Most importantly, Fusion was able to deliver the same security, scalability, and reliability that Oracle customers expect.

Benefits

The Fusion team experienced improvements in application response times by almost 40% and saw CPU utilization decrease by 25%, with memory utilization improving by over 20%. Performance enhancements of Java EPP included tests showing ~40% better workload (application) performance, a reduction in memory footprint requirements, and pause times for garbage collection were reduced by up to 99%. [1]

For more information

- Documentation: Oracle Java SE Subscription Enterprise Performance Pack
- Oracle Java Enterprise Performance Pack Release notes
- Blog: Meet the Java Enterprise Performance Pack (Java EPP)

---

[1] In internal testing, performance benefits varied across type of customer, workloads, and outside dependencies, with consistency in lower memory usage, significantly less time doing garbage collection, and CPU and response times improved across most customer usage patterns.

ORACLE

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

📝 blogs.oracle.com   📘 facebook.com/oracle   🐦 twitter.com/oracle