CRACKING THE CODE FOR JAVA ON THE MAINFRAME

COBOL remains the dominant programming language on the mainframe, but Java® is making substantial inroads on COBOL's popularity.

According to the 2024 BMC Mainframe Survey, developers are writing 64 percent of new mainframe applications in Java – and they are rewriting a remarkable 55 percent of existing applications in the language as well.

Clearly, mainframe operators must treat Java as a first-class mainframe participant by leveraging appropriate tooling.

While tooling that supports Java development in the distributed world is familiar and commonplace, the mainframe requires specialized tooling so organizations can optimize their use of Java on the mainframe.

As a result, the mainframe context for Java requires extra care from the organizations implementing it.

Java tooling requirements on the mainframe

BMC is a pioneer in DevOps on the mainframe and retains its leadership role with a comprehensive suite of mainframe management, DevOps, and automation tools under its Automated Mainframe Intelligence (AMI) brand.

It's no surprise, therefore, that BMC offers tooling that supports and optimizes Java on the mainframe. In fact, BMC offers a complete Java toolset, extending the value of established mainframe tools to Java.

Optimizing Java on the mainframe requires tooling specific to the platform. Analyzing and optimizing Java performance on the mainframe requires different tools than similar tasks in distributed environments.

The performance of Java applications, for example, depends upon the infrastructure supporting those applications, including all the dependencies among various infrastructure elements that provide Java with a runtime context.

Such modernization tools must take into account the specific requirements of the mainframe, including data structures and integration with mainframe assets and other dependencies.

Troubleshooting Java on the mainframe

How the mainframe handles Java exceptions is also different from Java in other environments.

Java provides its own exception handling, of course – but developers don't always implement it properly. As a result, there is always a chance that a programming failure will impact more than the failed program itself.

The mainframe handles exceptions in its own way, as any COBOL developer will attest to. For this reason, Java on the mainframe requires its own approach to exception handling.

Addressing this need is BMC AMI DevX Abend-AID, which brings automated exception handling to Java applications on the mainframe, supporting the troubleshooting of Java-based applications.

BMC AMI DevX Abend-AID automatically detects, analyzes and diagnoses problems across multiple mainframe environments, including Java. By extending the power of BMC AMI DevX Abend-AID to Java workloads on the mainframe, developers become more productive across the full lifecycle of Java development.

Managing the performance of Java workloads

The second tool that BMC has extended to Java on the mainframe is BMC AMI Strobe, which enables operators to capture and analyze Java performance data, empowering developers to locate and eliminate resource bottlenecks for Java applications on the mainframe.

BMC AMI Strobe helps operators identify application inefficiencies that lead to excessive CPU

consumption and prolonged execution times. BMC AMI Strobe for Java® combines its powerful measurement capabilities with the BMC AMI Ops Monitor for Java Environments so that operators can measure and analyze the performance of Java workloads on the mainframe.

By leveraging BMC AMI Strobe, mainframe teams can improve the performance of their Java applications and develop more efficient and responsive applications moving forward.

The Intellyx take

The combination of BMC AMI DevX Abend-AID and BMC AMI Strobe empowers mainframe Java

development teams to develop and troubleshoot their applications while maintaining the reliability that organizations have come to expect from their mission-critical mainframe systems.

These benefits extend across the entire mainframe landscape by enhancing support for modernization and AI initiatives, building cross-platform expertise, and reducing mean time to resolution for mission-critical systems that include Java applications.

For the organizations depending upon Java on the mainframe to support their mission-critical application development and modernization efforts, BMC's industry leadership provides the support they require to drive innovation on the mainframe.