OVUM OPINION

Java relevance continues as Java 7 launches

Reference Code: OI00126-069

Publication Date: July 2011

Author: Michael Azoff

OVUM VIEW

Summary

It is trendy to dish the dirt on Java and herald its demise, but as father of Java James Gosling recently said, it is not Java that matters so much because the relevance today is in the platform. Unlike in the early days, the Java platform is now host to multiple languages, which run on the Java Virtual Machine (JVM). As this shift from language to platform matures, the support that non-Java JVM languages receive will increase. Oracle launches Java 7, the first major release of Java in five years, on July 28, 2011. The outlook for enterprise server-side Java remains as strong as before, but the situation on the client side, including web and mobile, is less clear. The options are currently too fragmented – an issue the Java community needs to put right.

Java is a safe choice

The Tiobe index of programming language popularity shows that Java remains the top language and has been on an upward trend since January 2010. C continues to be the second-most popular language, and the gap between Java and C and the rest is widening. The continued importance of C can be understood from the explosion in demand for embedded software, where C predominates, but Java continues to hold sway at the top.

Java has certainly gone through some changes: the criticism that Java Enterprise Edition (EE) is overly complex created a movement to simplify server-side Java, and gave rise to Plain Old Java Objects (POJO) techniques and application frameworks such as JBoss and Spring. It is true that Java EE remains complex to develop with, but efforts are being made to reduce that complexity. The richness of the Java libraries and the possibility of mixing languages within a JVM application

to bring out the advantages of individual languages make investing in server-side Java a safe choice.

For complex enterprise applications the choice is still between Java and Microsoft .Net, though on the client side and particularly for web and mobile applications, the interfaces are increasingly fragmented between various technologies. Java has had a mixed client-side history; for example, Java ME was a success on mobile feature phones, but for smartphones Java is lagging, and the market has shifted strikingly to the new generation of mobile smart devices. There are numerous Java web client solutions to choose from, and this in itself is a problem. The need for frameworks such as Struts highlights the lack of a compelling single client-side choice. With mobile becoming the center of attention, there appears to be a missed opportunity for Java – for example, no JavaFX for smartphones. Rather, it is HTML5 that is gaining traction.

However, as smart mobile devices become more powerful, boasting quad core system-on-chip processors and huge storage capacities, one can expect a re-evaluation of client-side Java options around 2014.

Java 7 and 8

The Oracle Java team describes Java 7 as evolutionary, whereas it describes the next release, Java 8, as revolutionary. The Java I/O file system interface is notoriously clumsy (apparently put together in haste), and Java 7 corrects that with JSR 203, as well as offering improved asynchronous I/O.

A significant feature is InvokeDynamic, a new linking mode – the first new instruction in the JVM since Java launched. This is designed to help the creators of dynamically typed non-Java languages run these optimally in the JVM. These languages include Groovy, JRuby, Jython, and Rhino (JavaScript). There are over 200 JVM languages now available, and a single JVM application can call on any mix of these languages. Project Coin (JSR 334) adds many small language enhancements; for example, Strings can now be used in a Switch statement.

In contrast, Java 8 will offer major step changes; a significant example is "Closure," from Project Lambda. Closure makes libraries easier to use, particularly in multi-core programming, with the JVM making decisions on how to distribute loads across multiple cores, rather than burdening the programmer with this task. Closure encapsulates methods with data and parcels these out to individual cores. According to Mark Reinhold, chief architect of the Java platform, the result will be the ability to perform Smalltalk and MapReduce types of programming. Modules will be introduced to assist large program construction, JVM convergence will take place, and collections will no longer be bound by 32-bit lengths so big data can be more easily handled.

Reinhold also makes the point that Java evolves by being stable and adding successful features innovated in other languages. To contradict the naysayers, Java has a healthy future: it is a primary choice for enterprise server-side development and although at present the client side (with web and mobile in mind) is fragmented, there is an opportunity for Oracle to improve that in the long run.

APPENDIX

Further reading

Oracle Refreshes Sun's Java Strategy, OI00124-032

Disclaimer

The facts of this report are believed to be correct at the time of publication but cannot be guaranteed. Please note that the findings, conclusions and recommendations that Ovum delivers will be based on information gathered in good faith from both primary and secondary sources, whose accuracy we are not always in a position to guarantee. As such Ovum can accept no liability whatever for actions taken based on any information that may subsequently prove to be incorrect.