

RBG moves their Java Banking Framework to OS/390 UNIX System Services

Rechenzentrale Bayerischer Genossenschaften eG (RBG) is a German IT service provider and data processing center for cooperative banks in Bavaria. RBG runs one of Europe's largest private networks, with over 4500 routers, 5000 servers and 30,000 workstations based on OS/2. RBG's core business is data processing, software development, and training and consultancy for their customers – 450 "Volksbanken" and "Raffesbanken" in Bavaria. RBG runs up to 10 million IMS transactions per day on two IBM 9672-Z87 servers and one AMDAHL M575, totaling 2765 MIPS and 20GB real memory. This Spring, RBG demonstrated its *Java Banking Framework*, which now runs under OS/390 UNIX System Services, at CeBIT in Hannover.



Starting from scratch

About two years ago, RBG decided to verify whether Java technology would be a reliable foundation for platform-independent software development. The RBG development team tried to find a Java framework which would support application prototyping, but nothing that fit our needs was available on the market. We thus decided to build a framework "on the fly" while we developed prototypes.

At that time, with EJBs not even in sight, we designed and built an application server on our own. It soon became clear that to provide maximum reliability and scalability, the different functions of the application server had to be split onto different logical servers, each one of them running a separate Java Virtual Machine.

Another major part of the framework – today called the *Java Banking Framework (JBF)* – was the frontend design. When the first version of the JBF was finished, we asked experts from IBM to do a code and design review. After a week of long, productive sessions, the requirements for the framework were clear and final coding began.

Joining the old and the new

As designed, the JBF's server side consists of two dispatchers that provide load balancing, one or more application servers (*Remote Object Servers*), several infrastructure servers for messaging and journaling, and one or more host connectivity servers (*Remote Host Servers*).

We think that host connectivity for the JBF is unmatched in functionality and ease of use. For example, we built a generator which creates Java classes out of COBOL programs. These generated classes provide functionality to map Java data types and names to COBOL and vice versa. For each IMS transaction on the host, one Java counterpart is generated and, if necessary, slightly modified by hand. Through an RMI¹ connection to a Remote Host Server, the business logic residing on the application server can start transactions on the host, without needing to know any COBOL specifics.

The connection between a Remote Host Server running on an OS/2 WARP Server and the host system is over a TCP/IP – SNA bridge, which also was created at RBG.

Getting ready for CeBIT

This January, we were about half way through the development of our first two production applications based on the JBF when IBM asked us if we would like to do a demo at CeBIT in February. This demo was to have the server modules running on S/390. Assuming that OS/390 UNIX System Services are just about like any other UNIX, we thought moving the pure Java JBF onto S/390 would be an easy job. And it was, for the most part. After about only four hours, we had major components of the JBF running under UNIX System Services.

As in every project, however, details consume most of the time and are tricky. One of these details was the MQSeries Java implementation on UNIX System Services, which differs in some very essential aspects from the one for Windows NT, our development platform. To work around some programming fields that are missing on the S/390 side, for example, we implemented compile time switching between the APIs used – just like we'd done earlier when using C.



Matthias Schorer, RBG

1. Remote Method Invocation



Different code pages are another serious issue for anyone who deploys Java programs in a multi-platform environment. When working with S/390, EBCDIC coding must be considered. For byte array / string conversion, Java by default always uses the standard ASCII code page of the Java runtime system. The solution is to tell Java which code page to take for converting a string into a byte array and vice versa. We found 8859_1 to work best, because it is available on all our operating systems.

Moving forward

In about three weeks, we overcame all obstacles, and the JBF was extremely stable under UNIX System Services. One reason for this is that the Java Virtual Machines manage memory better when compared to JVMs on OS/2 or Windows NT. Of course, another reason is S/390's proven and solid system architecture, on which the JVMs are based.

During CeBIT, interest in the demo was overwhelming. The JBF servers ran flawlessly, and the RMI connection to the Remote Host Server was extremely stable for the whole show. Our CeBIT experience, plus the fast and successful port of the JBF to S/390, makes us confident that with the S/390 as our e-business server, we can again centralize many of the processes used at RBG.

Additional information

The Rechenzentrale Bayerischer Genossenschaften EG has its web site (in German) at: www.rbg.de

To learn more about Java for OS/390, access: ibm.com/s390/java

Matthias Schorer
RBG Java Chief Architect
matthias.schorer@rbg.de

Note: The web location of this article is: ibm.com/s390/ftp/bulletin/issue28/rbg.pdf

IBM supports new Java technology

Easing application deployment

This May, IBM announced delivery of technology which allows applications written in the latest version of Java to run on a variety of computer systems. With this technology, businesses will be able to ease deployment of Java-based applications across a variety of operating systems.

IBM will deliver nine Java Virtual Machines (JVMs) that support Java Standard Edition Version 1.3. A JVM is the technology that lies between the application and the operating system to give Java its cross-platform capabilities.

Offerings for Windows, Linux, AIX®, OS/390, OS/400® and OS/2 all will be available. In addition, IBM also revealed it will provide the industry's first Java 2 JVMs for Intel's Itanium processor running Windows 2000, Linux and AIX/Monterey, a code name for the delivery of AIX on the Intel IA-64 processor-based systems. Availability of the Windows 2000 and AIX/Monterey Java 2 JVMs will be simultaneous with the launch of Itanium this year. IBM, working with Intel, has made alpha versions available in an early support program for developers.

"Through it's speedy delivery of fully compatible Java technology on nine computing platforms, IBM reinforces its leadership in making the technology real for business both on the server and client," said Thomas Dwyer, Research Director for Enterprise Java, Aberdeen Group, Inc. *"IBM's work with major industry partners like Intel and the key Linux distributors shows the company is quickly becoming a preferred provider of Java technology."*

Addressing performance requirements

IBM also is working on enhancements to its JVM technologies to address the specific high performance requirements of short-lived, fast, transactional applications. This will allow Java to be used effectively in the place of traditional languages in high transactional environments. IBM plans to add these new technologies first to OS/390 and then expand the new capability to the Windows, AIX/Monterey for IA-64, and Linux platforms.

"Being the first to offer a broad line of production level 1.3 JVMs to the market underscores IBM's commitment to delivering industry leading, fully compliant Java technology," said Rod Smith, Vice President, Java Software, IBM. *"The need for cross-platform computing is becoming more evident as e-businesses begin to interact with e-marketplaces."*

Additional information

IBM's Java technology is long recognized by the industry as robust, fast, scalable and stable. As proof of this leadership, all four high volume distributors of Linux recently chose to ship IBM's Developers Kit for Linux, Java Technology Edition, with their products. IBM's JVM implementations are rated the highest performing and most scalable by industry benchmarks like VolanoMark and SpecJVM98.

For more information on IBM's support of Java 1.3 go to: www.developer.ibm.com/java/j2

