



WebSphere software

IBM WebSphere Real Time: Providing predictable performance

*By Michael S. Fulton, Java chief architect;
Darren V. Hart, Real Time team lead; and
Gregory A. Porpora, IBM Software Group*

Contents

- 2 Introduction**
- 3 Real-time misconceptions**
- 2 Real-time determination compared to application complexity**
- 5 IBM WebSphere Real Time, Version 1.0 in detail**
- 9 Real-time Linux**
- 10 Possible future real-time solutions from IBM**
- 11 Conclusion**
- 12 For more information**

Introduction

Java™ technology has improved general software-development productivity and reliability while helping to reduce complexity and provide a structured, well-understood development, deployment and runtime environment. Java is available on all the key operating-system and hardware platforms, and has become the environment of choice for Web applications, middleware, tools and service oriented architecture (SOA). By employing the Real-Time Specification for Java (RTSJ), the Java platform (in this case Java 2 Standard Edition [J2SE]) has been extended to support real-time applications. And now, IBM WebSphere® Real Time provides real-time response while still using standard class libraries through its real-time incremental garbage collector known as *Metronome*. Programmers no longer have to rely on languages such as C, C++ or ADA 95 for real-time programming. This Java platform now represents a viable potential alternative environment for highly deterministic, distributed, real-time applications in critical system applications, ranging from command and control, weapons, industrial automation and financial systems to telecommunication infrastructures.

IBM is a major provider of Java technologies from enterprise to security, system management, development and portal applications with significant investments in research and development.

Real-time support, or more appropriately, support for a deterministic execution environment within a Java Virtual Machine (JVM) or a collection of JVMs, has become an increasingly important element of IBM's Java strategy. The IBM product that provides this real-time solution is called *IBM WebSphere Real Time*. It is based on IBM's real-time JVM extended with the IBM Metronome real-time incremental garbage collector, which enables the use of standard class libraries by real-time code, Ahead of Time (AOT) compilation and full support for RTSJ, Java Specification Request (JSR) 1. These components combine to run on a real-time operating system (RTOS) based on a modified Red Hat Enterprise Linux®, Version 4.0 with Update 2 system developed in partnership among the IBM Linux Technology Center, Red Hat and the Linux open-source community.

This white paper describes the components of the real-time Java environment, available today in the IBM WebSphere Real Time product. It also discusses some of the product's future capabilities as described in the IBM alphaWorks® Emerging Technologies site at www.alphaworks.ibm.com/topics/realtimejava.

Real-time misconceptions

Several misconceptions exist related to real-time systems. The first and most common is that real time is solely about minimal application latency and maximum data throughput. These capabilities are important attributes in critical systems or applications, but real time is foremost about latency determinism compared to data determinism. In other words, real-time is about predictable performance.

The second misconception is that real-time is solely for embedded applications and systems. Although this has been the heritage of many real-time systems, in the past five years, the need has grown for enterprise-level latency determinism. Only recently have open-standards-based technologies been able to provide a viable alternative to extremely expensive and complex proprietary systems. Financial, industrial automation, telecommunication, military and aerospace, and online gaming systems are all examples of enterprise systems that must be able to aggregate multiple events, messages and data streams across a heterogeneous distributed infrastructure, helping to ensure a well-defined and guaranteed end-to-end response time.

Real-time determination compared to application complexity

As the requirement for deterministic behavior moves from tightly coupled embedded tasks that control or monitor hardware subsystems to highly complex enterprise applications, software complexity and deterministic latency requirements become inversely proportional to solution needs and capabilities. Figure 1 depicts this situation by showing how, when applications become more complex, the time required for deterministically processing all necessary information increases. Smaller, less-complex solution components running closer to the hardware require tighter determinism and lower latency.

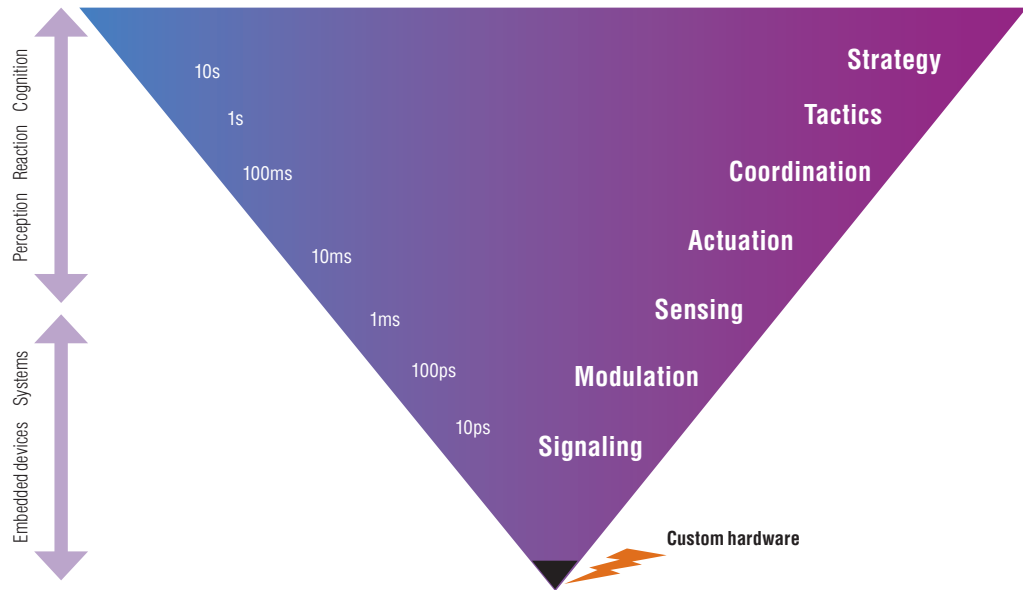


Figure 1. Real-time systems: software complexity compared to time domain

Within this overall complexity, architects, systems designers and developers need an environment that can span a significant portion of the response spectrum. This model must not only meet a broad bandwidth of performance and latency requirements, but also be scalable and relatively simple to use. IBM WebSphere Real Time technology meets the challenge – providing an environment that can address real-time processing from seconds to the millisecond down to the tens of microseconds.

IBM WebSphere Real Time technology was recently chosen as the core development and runtime environment for a major weapons program for the U.S. Navy’s newest destroyer. IBM’s responsibility in this multibillion dollar effort is called the *Total Ship Computing Environment (TSCE)*. After an exhaustive comparison and benchmarking against the incumbent Java runtime, IBM WebSphere Real Time and Red Hat technology-based real-time Linux running on X86 blades were selected.

The real-time, Java technology-based solution from IBM used in this weapons program is able to address distributed real-time processing from seconds down to one millisecond. It provides predictable performance without the need to move away from the traditional Java programming model, using the standard class libraries all Java developers are familiar with, by taking advantage of innovative IBM garbage collection and compilation technologies. These capabilities were a key factor in the selection. By enabling developers to use tools, skills and development best practices, organizations can develop real-time applications with a higher potential for a lower total cost of ownership (TCO), better performance, lower development risk and reduced complexity. This same technology is being examined for use by other U.S. Navy, U.S. Air Force, U.S. Army and U.S. intelligence programs, as well as financial institutions and telecommunications companies.

Also, as this paper later discusses, you can use standards-based Java extensions like RTSJ to achieve deterministic behavior well below one millisecond.

IBM WebSphere Real Time, Version 1.0 in detail

The key to the IBM WebSphere Real Time, Version 1.0 solution is that at its core it is designed to preserve the value of the Java language and platform through predictable garbage collection. IBM helps reduce or eliminate the primary source of nondeterminism through its incremental garbage collection technology known as *Metronome*.

By allowing garbage collection within the real-time Java programming model, memory management remains transparent to developers, unlike JSR 1 RTSJ scoped memory. Scoped memory can be extremely difficult to manage, introducing complexities in programming and forcing developers to determine their memory requirements prior to run time. Further, by using unchanged existing J2SE, Version 5.0 class libraries, IBM WebSphere Real Time retains the productivity and reusability aspects that make the Java development environment so attractive.

IBM continues to extend the reach of Java through its technological advancements in garbage-collection technology, just-in-time (JIT), and AOT compilation. These features enable programmers to use the standard Java programming model and libraries for real-time applications requiring predictable maximum latencies down to one millisecond. For latencies below one millisecond down to 50 microseconds, programmers can use the fully compliant RTSJ No Heap Real Time Thread (NHRT) features that are also available with IBM WebSphere Real Time. For programmers, this capability means that they can write the majority of their real-time applications using standard Java class libraries and methodologies. The ability to perform this task offers a major advantage over other real-time Java solutions that require programmers to always use the extended programming model in RTSJ and deal with difficult techniques, such as user-defined memory management, as well as complex and error-prone problems, requiring a deep understanding of real time in addition to Java development.

IBM could not have delivered these impressive performance results without the dedicated collaboration between IBM, Red Hat and the Linux open-source community, all of whom worked together to define a solid Linux RTOS with predictable scheduling and performance – all in less than a year.

IBM WebSphere Real Time also provides support for the full RTSJ specification, enabling programmers to use the many standard features, including priority-based thread scheduling, high-resolution timers and asynchronous event handlers. Other features in IBM WebSphere Real Time include improved class loading through the use of Java executable (JXE) or Java archive (JAR) files to support fast, compatible class loading. Programmers can also use JIT and AOT compilation to optimize the implementation environment. You can choose between dynamic (JIT) and static (AOT) compilation. When running with JIT, compilation is performed on an asynchronous, low-priority thread, helping to ensure that critical regions running on high-priority threads can preempt compilation so that it runs consistently. Alternatively, programmers can use AOT technology to precompile code, enabling them to run their applications without an active JIT compilation, helping to reduce the complexity of ensuring critical region code is compiled before its first invocation.

Figure 2 summarizes the attributes of IBM WebSphere Real Time.

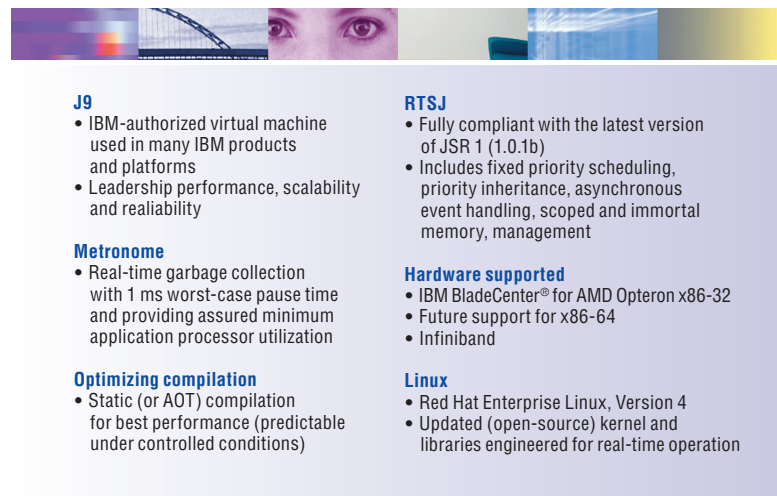


Figure 2. IBM WebSphere Real Time attributes

Figure 3 depicts the demarcation lines highlighting where a programmer can write standard Java code for real-time applications, using the Metronome nonintrusive garbage collector, compared to using NHRT and scoped-memory-management techniques.

The RTSJ specification allows for three different thread contexts. First is standard Java Threads (`java.lang.Thread`), second is Real Time Threads (RTTs) (`javax.realtime.RealtimeThread`), and third is NHRT. RTTs in conjunction with the Metronome garbage collector can achieve predictable latencies as low as one millisecond. NHRTs get lower predictable latencies, especially when combined with RTSJ memory-management techniques such as immortal and scoped memory.

Applications requiring latencies below one millisecond need to be written to run on NHRTs that use RTSJ memory-management capabilities. However, this development model is significantly more difficult and error-prone than the basic Java development model, requiring similar expertise of a real-time programmer working in C or ADA. Also, the programmer must pay significant attention to memory management as opposed to focusing on the actual application solution being created.

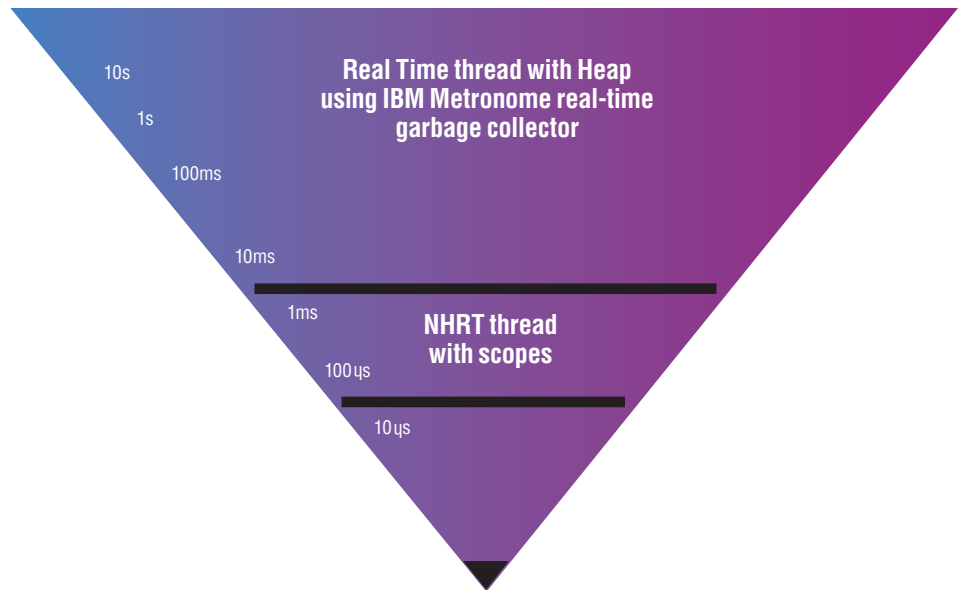


Figure 3. Metronome with RTSJ

The IBM WebSphere Real Time software development kit (SDK) and Java Runtime Environment (JRE) key features include:

- *Full support for J2SE, Version 5.0.*
- *Full support for JSR 1, the JCP Real-time Specification for Java 1.0.1b, commonly referred to as RTSJ.*
- *Support for symmetric multiprocessing (SMP) (four-way) AMD Opteron multiprocessor x86 Linux platforms (including IBM BladeCenter servers).*
- *The ability to run more than 2000 concurrent real-time and NHRT threads across all SMP (four-way) blade-server adapters.*
- *JXE support.*
- *A well-defined list of Java classes that are NHRT-safe.*
- *Hard real time (Metronome) to less than one millisecond latency.*

Figure 4 depicts the IBM WebSphere Real Time architectural stack that was instrumental in the DDG-1000 solution.

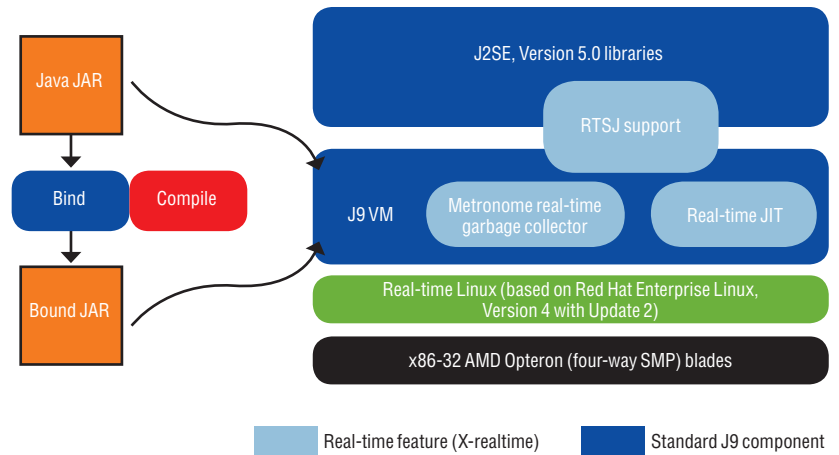


Figure 4. IBM WebSphere Real Time, Version 1.0 architecture

Real-time Linux

A core feature of the real-time Java environment is its underlying RTOS. In keeping with IBM’s philosophy of being a leader in open standards, the first supported RTOS will be an enhanced Linux kernel and supporting libraries based on Red Hat Enterprise Linux, Version 4 with Update 2. IBM’s Linux involvement is managed through the IBM Linux Technology Center

Visit ibm.com/linux/ltc/mission.shtml for more information about the missions of IBM Linux Technology Center. The real-time Linux extensions for IBM WebSphere Real Time are built on the work lead by Ingo Molnar of Red Hat and Thomas Gleixner of Linuxtronix.

IBM has an exceptional record and much experience working with the Linux development community in getting its contributions accepted into the mainstream sources and adopted by Linux distributors such as Red Hat and Novell. The development work necessary for enhancing the Linux kernel to support a real-time environment is no exception; IBM has been working with the Linux community to develop the real-time function needed to support the real-time Java environment. Some current releases of real-time Linux efforts restrict real-time tasks to a single processor or a subset of the processors on the system, and lack the security protections of arbitrary code running with kernel privileges. IBM’s approach is to enable all processes to have access to all processors on the system. Also, many of these recent real-time Linux variants are proprietary, with patches that are not likely to be adopted into mainline Linux.

Key features of the real-time Linux extensions include:

- *Based on Red Hat Enterprise Linux, Version 4 with Update 2.*
- *Support for extended real-time POSIX application programming interfaces (APIs).*
- *Supported on select IBM System x™ machines.*
- *Fully preemptive kernel, helping to reduce critical path latencies.*
- *Priority inheritance-enabled kernel and userspace locking.*
- *High-resolution timer support for enhanced scheduling, with better than 10 μs lowest-achievable sleep time (compare to 3*HZ in the standard Linux kernel, typically 3 to 12 milliseconds).*
- *Highly accurate gettimeofday implementation, with nanosecond resolution.*
- *Robust mutexes that enable userspace to recover from application failures while holding a mutex.*
- *Enhancements to the Pluggable Authentication Modules (PAM) libraries to allow nonroot users access to real-time facilities.*
- *Direct physical memory access as required for RTSJ conformance.*
- *Context switch latency* under 25 μs (99.9999 percent under 20 μs).*

Possible future real-time solutions from IBM

IBM expects that as it continues to develop a family of real-time middleware products, the core enabling technology will be IBM WebSphere Real Time, Version 1.0 JRE running on top of an RTOS such as Real Time Linux.

It is IBM's current plan, in future releases of IBM WebSphere Real Time, to continue to introduce leading-edge innovation to provide tighter predictable performance for its hard real-time clients, as well as targeting a broader range of hardware and software configurations. For details, visit the IBM alphaWorks Web site at www.alphaworks.ibm.com/topics/realtimejava.

Conclusion

Figure 6 depicts IBM's ultimate distributed real-time platform, which includes middleware, tools and management systems. This infrastructure can support end-to-end critical deterministic systems that can interface with real-time middleware such as RTI Data Distribution Service and PrismTech OpenFusion RTORB Java Edition and PrismTech OpenSplice DDS, both of which support IBM WebSphere Real Time, Version 1.0 for critical real-time Java applications.

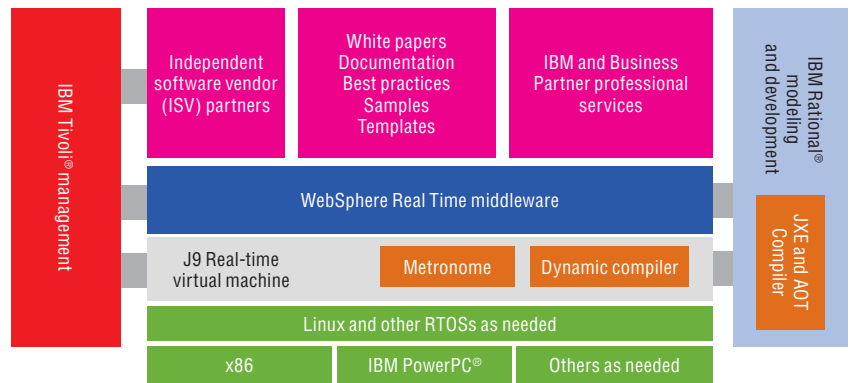


Figure 6. The IBM real-time platform and support tools based on IBM WebSphere Real Time

Real-time programming with an IBM WebSphere Real Time, Version 1.0 technology-based solution provides IBM's best Java technology-based platform for addressing the sources of nondeterminism while maintaining the benefits of the core Java development environment. It helps eliminate the need for users to perform memory management for all but the most-stringent latency domains (such as, below one millisecond). It has potential for application across many industries including defense, telecommunications, gaming, industrial automation, finance and banking. IBM continues its research and development effort to drive latencies in the IBM WebSphere Real Time virtual machine down to the microsecond levels. IBM also continues to invest in growing the capabilities of the real-time platform as well as to increase the ease of use, making it easier to develop, deploy and manage real-time solutions based on IBM's middleware products and tools.

For more information

To learn more about IBM WebSphere Real Time, Version 1.0, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/software/webservers/realtime

To join the Global WebSphere Community, visit:

www.websphere.org



© Copyright IBM Corporation 2006

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
12-06
All Rights Reserved

alphaWorks, BladeCenter, IBM, the IBM logo, PowerPC, Rational, System x, Tivoli and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries or both.

Other company, product, or service names may be trademarks or service marks of others.

* Latency measurements are dependent on hardware and system configuration.