WebSphere® Real Time Deterministic Java™

Alan Stevens
IBM Real-Time Technologies
Alan_Stevens@uk.ibm.com

Agenda

IBM’s perspective on real-time Java

WebSphere Real Time architecture

Metronome garbage collector

Tuning Fork demo

SMI considerations

Version 2 highlights

Trademarks and acknowledgements

IBM, WebSphere, alphaWorks, developerWorks are trademarks of International Business Machines Corporation in the United States, other countries, or both. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States and/or other countries. Other company, product, or service names may be trademarks or service marks of others.

All performance data contained in this presentation was obtained in a specific environment and is presented as an illustration. Performance obtained in other operating environments may vary

What does real-time mean?

Real-time : <u>predictability of performance</u>

hard : violation of timing constraints are hard failures

soft : timing constraints are simply performance goals

Constraints vary in magnitude (microseconds to seconds)

Consequences of missing a timing constraint:

from service level agreement miss (stock trading)

to life in jeopardy (airplanes)

Real-fast is not real-time, but *Real-slow is not real-good*

Need a balance between predictability and throughput

IBM's interest in real-time

- **Classical real-time systems are getting more complex**
  - Military, telecom, financial, industrial, automotive
- **Real-time systems becoming part of enterprise IT**
  - Sensor networks, Event processing
- **Commercial systems have unpredictable performance**
  - Service Level Agreement failures when overloaded
- **A need for a new way to build real-time systems**
  - Engineered for predictability and reliability
  - Using the latest programming tools and techniques

Why Java?

A business advantage over C, C++, Ada

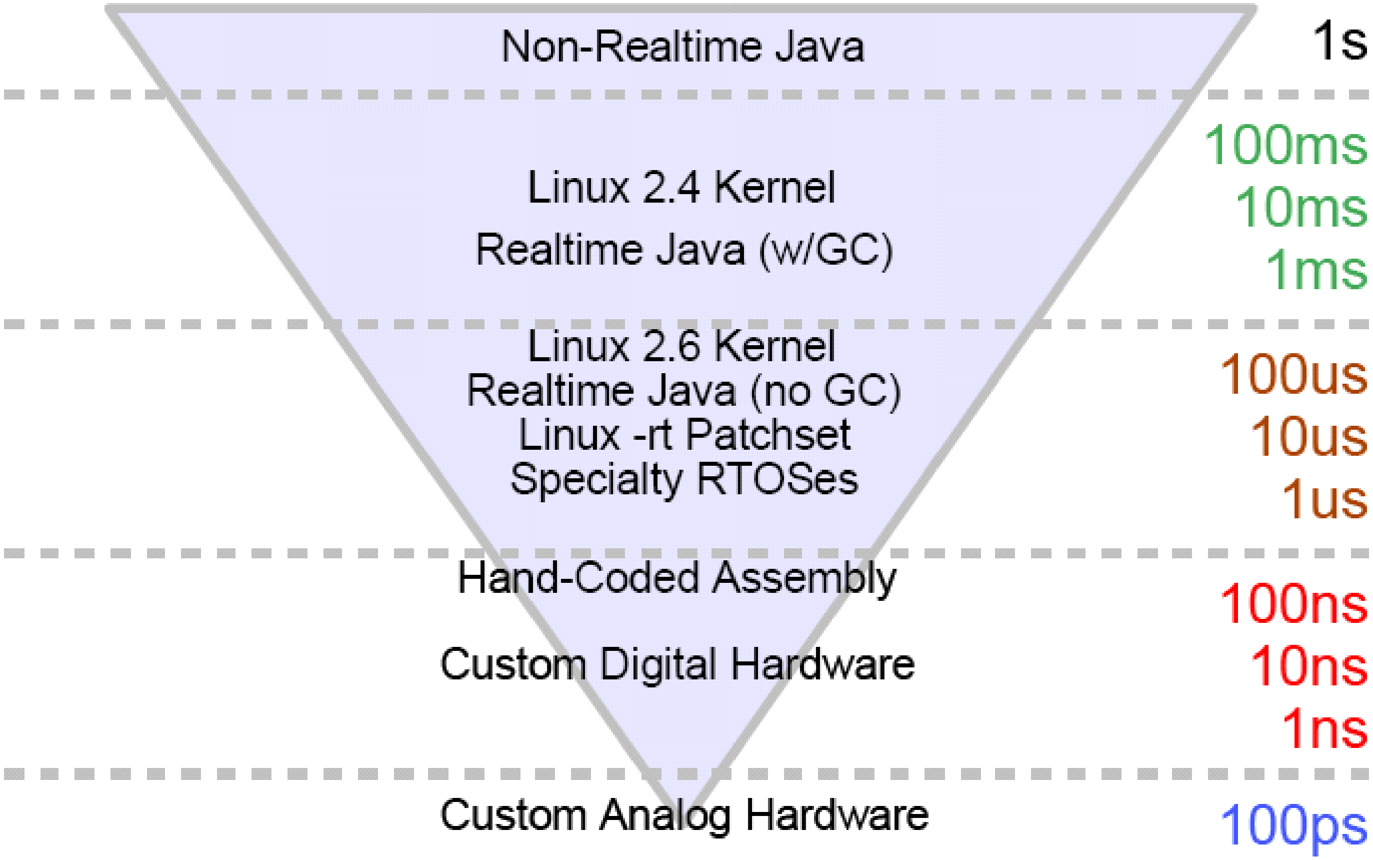
- Productivity from tools, portability, error checking, security
- Many skilled programmers available
- Massive community of ISVs

Java has problems in real-time environments

- Lazy class loading and initialization, dynamic compilation
- Garbage collection, system-specific thread management

IBM has solved these problems

Real-time Capability Triangle



Non-Realtime Java — 1s

Linux 2.4 Kernel
Realtime Java (w/GC)
100ms
10ms
1ms

Linux 2.6 Kernel
Realtime Java (no GC)
Linux -rt Patchset
Specialty RTOSes
100us
10us
1us

Hand-Coded Assembly
Custom Digital Hardware
100ns
10ns
1ns

Custom Analog Hardware — 100ps

Updated from: SMP and Embedded Real-time (article in the Linux Journal)

by Paul McKenney (Distinguished Engineer, Linux Technology Center) http://www.linuxjournal.com/article/9361

Real Time Specification for Java (RTSJ)

Augments Java with various services to support building real-time systems

Thread scheduling

- "RealtimeThread" allows specification of scheduling parameters
- Used in conjunction with Metronome, low latency achieved with no change in programming model
- Fixed priority scheduling and additional priority settings
- Many event management services provided

Real Time Specification for Java (cont.)

Memory Management

- Partitioned, non-garbage collected memory spaces
- No Heap Realtime Threads (NHRTs) can run independent of GC
- Very low latency achieved using standard RTSJ scoped memory techniques with NHRTs

Synchronization

Priority inversion avoidance (priority inheritance) on Java monitors as well as locks managed by the JVM and operating system

Non-blocking queues to communicate between real-time and non-real-time threads

WebSphere Real Time – hard real time

WebSphere Real Time (WRT) V2 is Generally Available

- WebSphere Real Time JVM is fully Java SE 6.0 compliant
- Full support for RTSJ (JSR #1) on Real Time Linux

Rigorously tested on:

RHEL MRG V1, SLERT 10,
RHEL4 Update 7, RHEL5 Update 2, SLES 10 SP2

The -Xrealtime **option gives additional Real Time function**

- Incremental GC and Incremental JIT

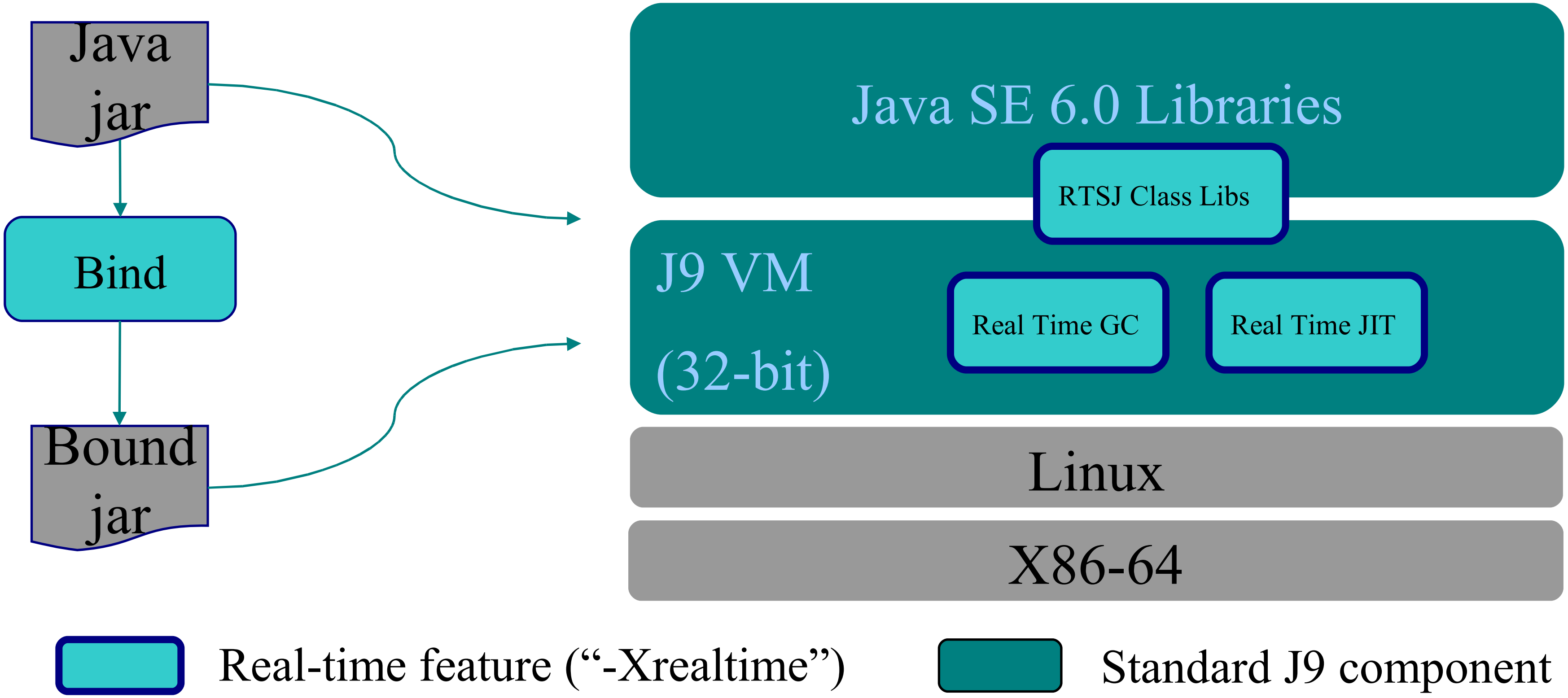
Java applications will run under WebSphere Real Time

- ... but will have more predictable performance
- ... and can be extended, where required, to use RTSJ

WebSphere Virtual Enterprise Exploitation – soft real time

- **WRT V2 can plug-replace your Web Services JVM**
  - Combine with WVE On-Demand Router (ODR)
    - Provides intelligent work-queue prioritization
    - User-controlled routing of work by priority
  - Full-stack of WVE, WAS ND, WAS, WRT
    - Rigorously tested by IBM on xSeries RedHat/SuSE
    - Available as of December 19th with WVE 6.1.0.5
  - Soft Real-Time provides more consistent response times
  - Certified on IBM xSeries Hardware
  - Supported on most Intel/AMD Hardware configs

WRT architecture



Compilation Strategies for Real Time

Compilation in J9 is dynamic by default

High throughput, but JIT may not run early enough in non-real-time JVM to guarantee consistent performance

Multiple compilation choices with WRT:

Ahead-of-time (AOT) (much better than interpreted performance)

User-controlled JIT (faster than AOT, controlled via API)

New with V2: Ability to mix'n'match AOT'ed and JIT'ed code

JIT-at-low-priority (best performance, runs on low priority thread)

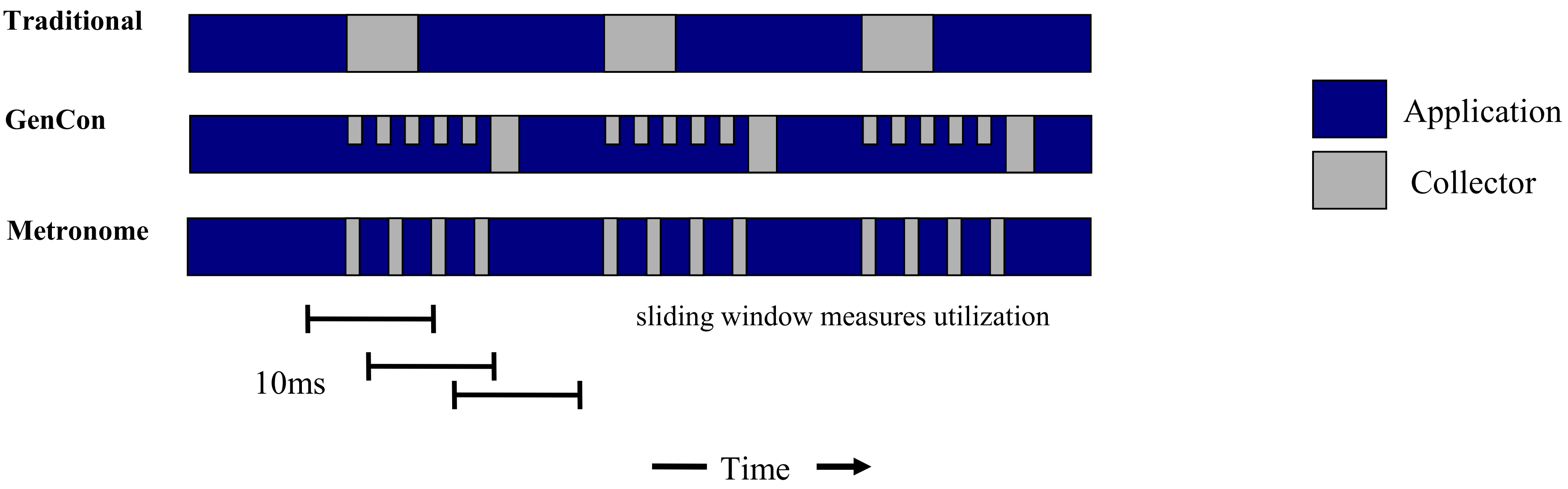
Tooling-controlled compilation as part of application start-up

Real-time Garbage Collection: The Metronome

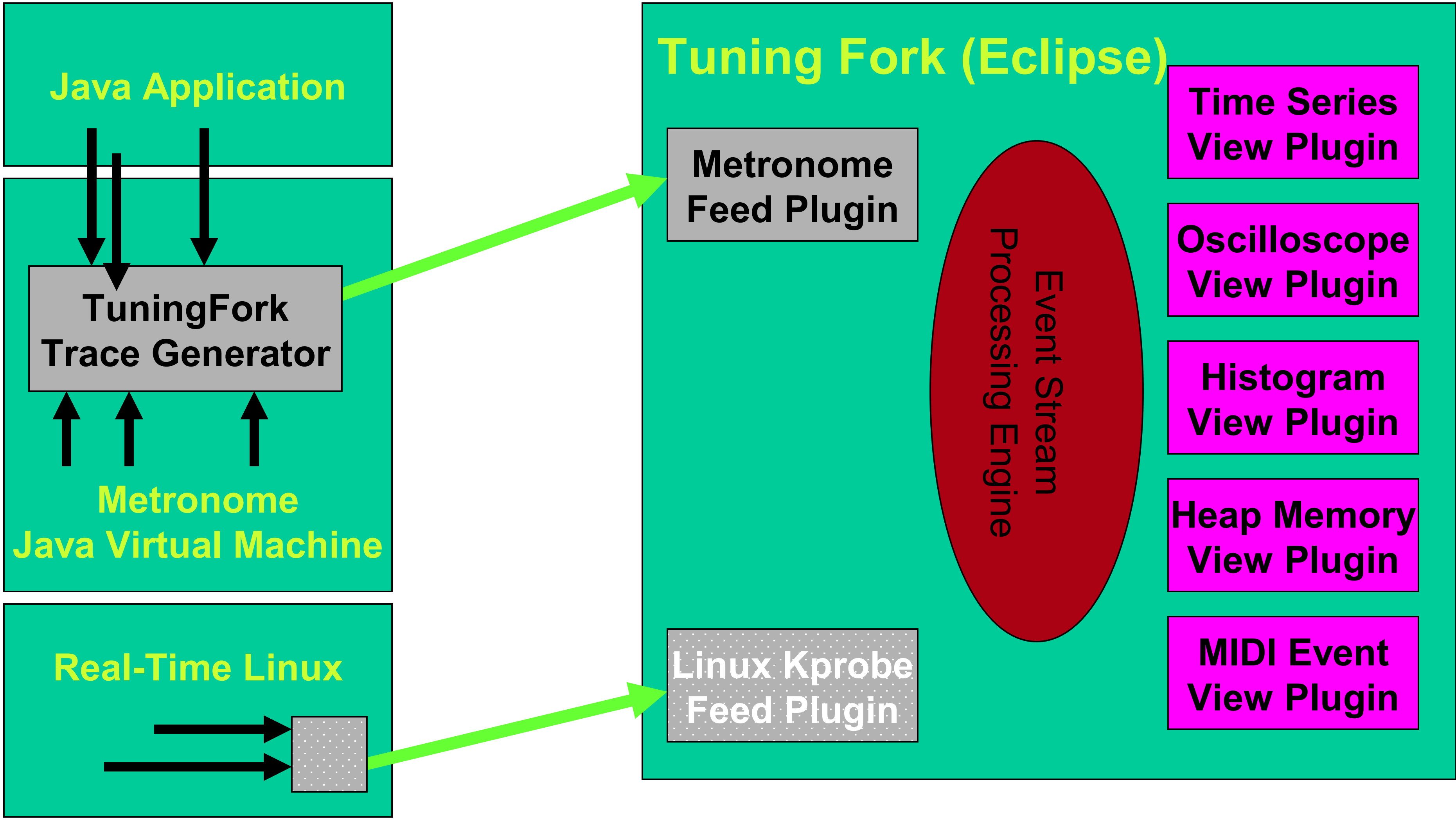
- Unique technology originally from IBM T.J. Watson research
  - Garbage collection is scheduled as just another periodic real-time task
  - Provides bounded pause times as small as 1ms and a minimum utilization level for application tasks
- Enables the use of off-the-shelf Java code
  - No need for specialized allocation schemes outside the Java heap
  - Greatly simplifies real-time application development
  - Enables complex real-time applications through easier composition
- Simple configuration
  - Based on allocation rate and live heap data
  - For a given time interval, configuration trades off minimum application utilization against required heap memory

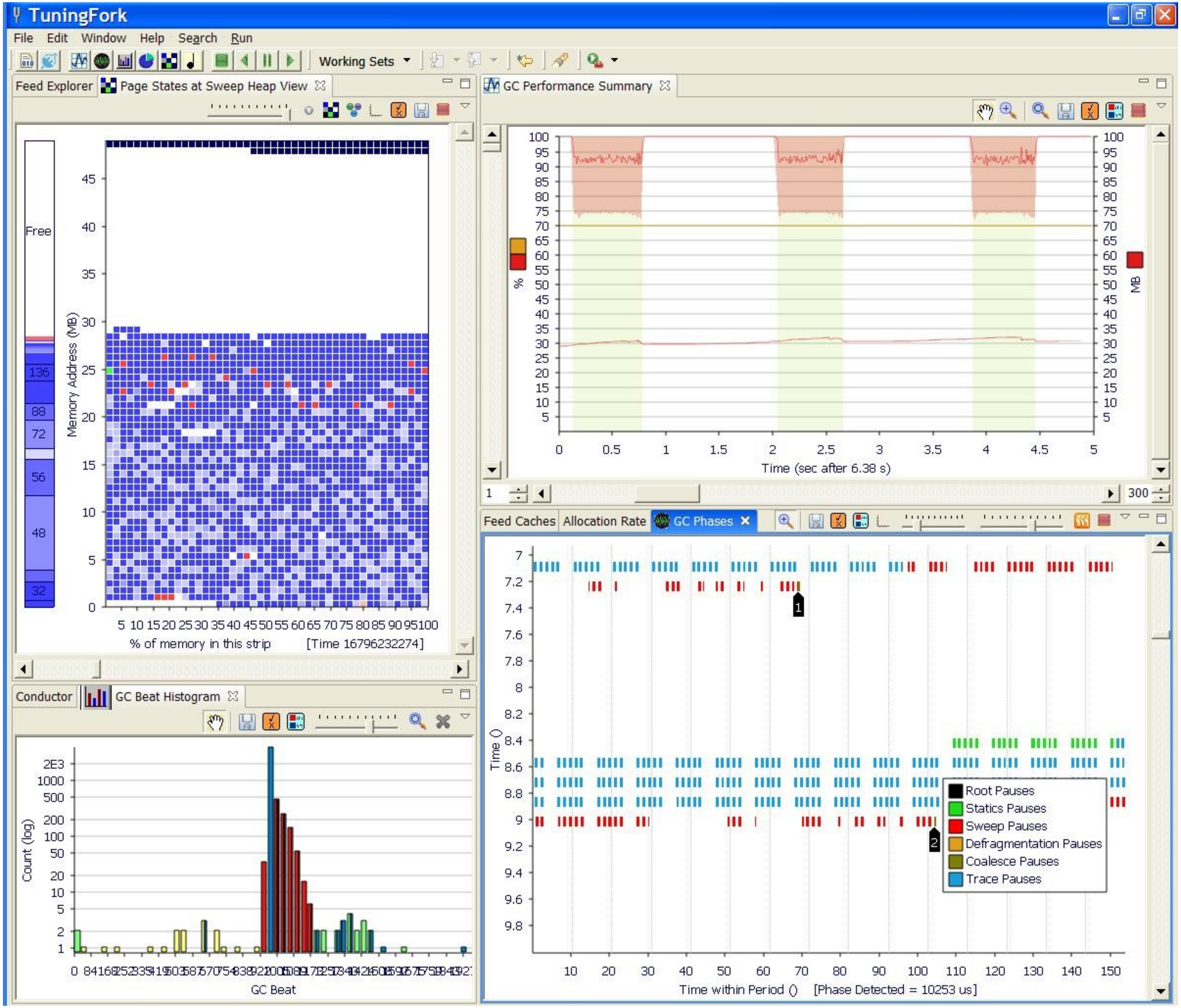
Comparison of Different Garbage Collection Policies

- **Traditional garbage collection requires a single Stop-the-World event**
  - Stop-the-World: all Java threads stop to permit collection
- **Generational Concurrent (GenCon) garbage collection**
  - primarily shorter collections concurrent with application thread on multi-processor systems
  - very infrequent stop-the-world global collections, typically shorter than traditional garbage collection
- **Metronome garbage collection guarantees maximum pause times with a minimum utilization**
  - Utilization is processor time dedicated to the application
  - Shortest pause times, but may have greater performance impact



Tuning Fork Architecture




TuningFork
File Edit Window Help Search Run
Working Sets
Feed Explorer
Page States at Sweep Heap View
Free
136
88
72
56
48
32
Memory Address (MB)
45
40
35
30
25
20
15
10
5
0
5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
% of memory in this strip
[Time 16796232274]
GC Performance Summary
%
MB
100 95 90 85 80 75 70 65 60 55 50 45 40 35 30 25 20 15 10 5
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
Time (sec after 6.38 s)
1
300
Conductor
GC Beat Histogram
2E3
1000
500
200
100
50
20
10
5
2
1
Count (log)
GC Beat
Feed Caches
Allocation Rate
GC Phases
7 7.2 7.4 7.6 7.8 8 8.2 8.4 8.6 8.8 9 9.2 9.4 9.6 9.8
Time ()
1
2
Root Pauses
Statics Pauses
Sweep Pauses
Defragmentation Pauses
Coalesce Pauses
Trace Pauses
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
Time within Period ()
[Phase Detected = 10253 us]

Real-time Garbage Collection: The Metronome

TuningFork tooling demo

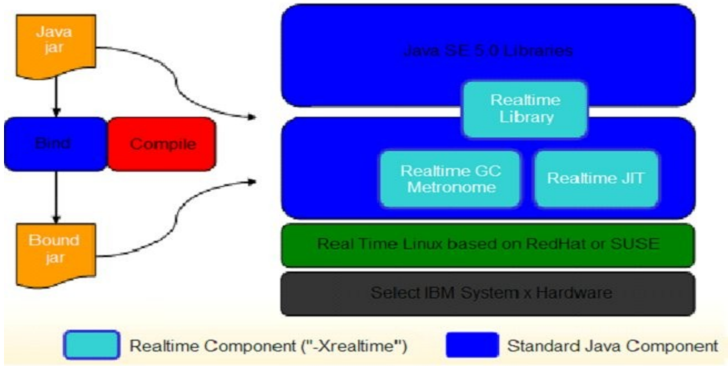
Real-Time OS and IBM Hardware Exploitation

Certified on Select IBM Hardware

- LS21 and HS21XM xSeries blades
- Enhancements for real-time workloads
- SMI Enhancements

Exploits RT Linux (RedHat MRG, Novell SLERT)

- High resolution time and timers
- Fully pre-emptible kernel
- Threaded interrupt handlers
- Priority inheritance & fast user-space mutexes
- Symmetric Multiprocessing (SMP) RT scheduling



System Management Interrupts (SMI)

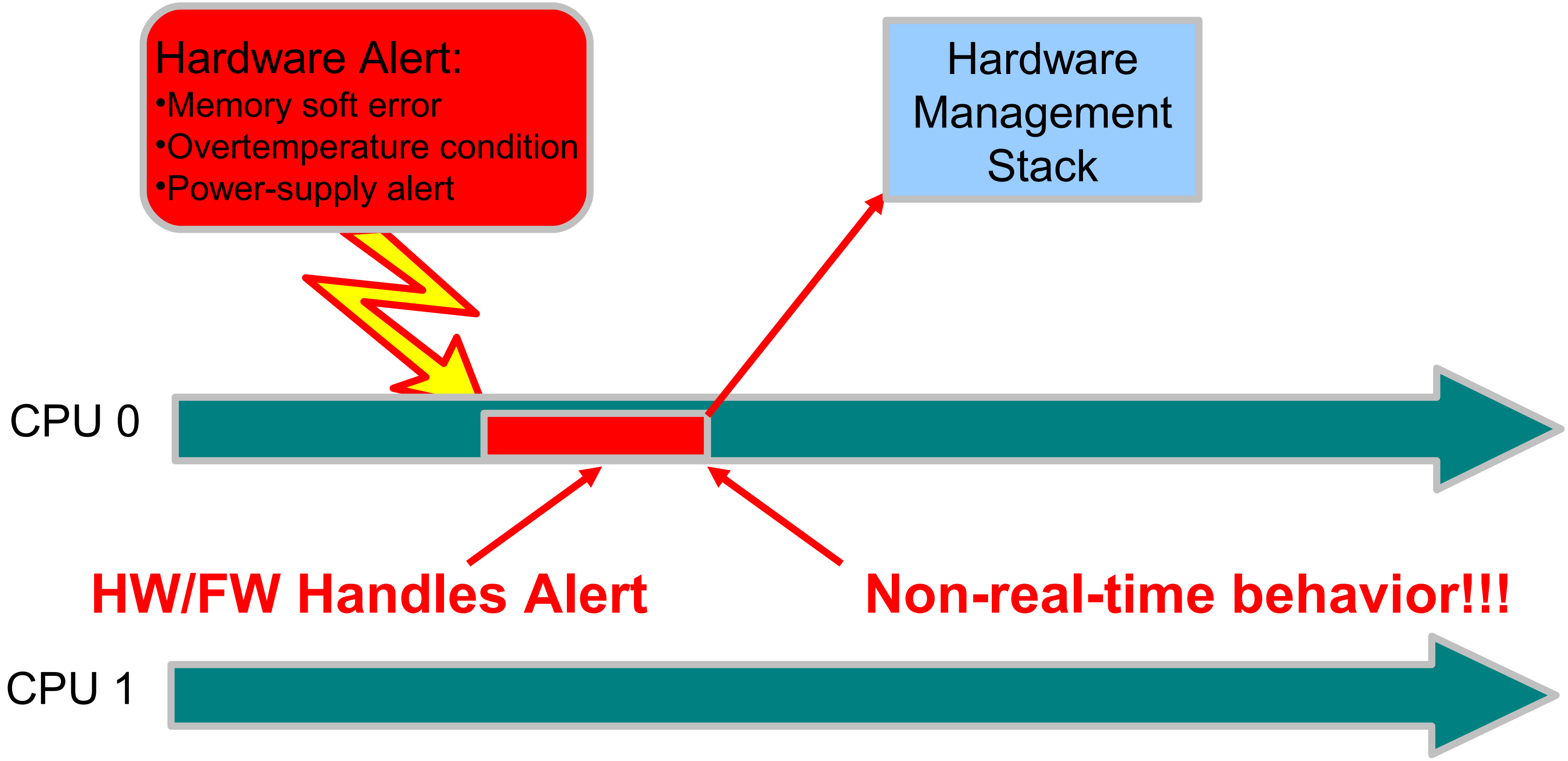
SMIs traditionally used to perform a variety of tasks

- Reporting of hardware errors (fatal and nonfatal)
- Thermal throttling
- Power capping

The nature of these interrupts causes latencies

- Not optimal for real time systems
- No Operating System (OS) notification or control
- Hard to detect
- Source of unwanted/unaccounted latencies in a real time system

Non-Real-Time Hardware Error Behavior



Hardware Alert:

- Memory soft error
- Overtemperature condition
- Power-supply alert

Hardware Management Stack

CPU 0

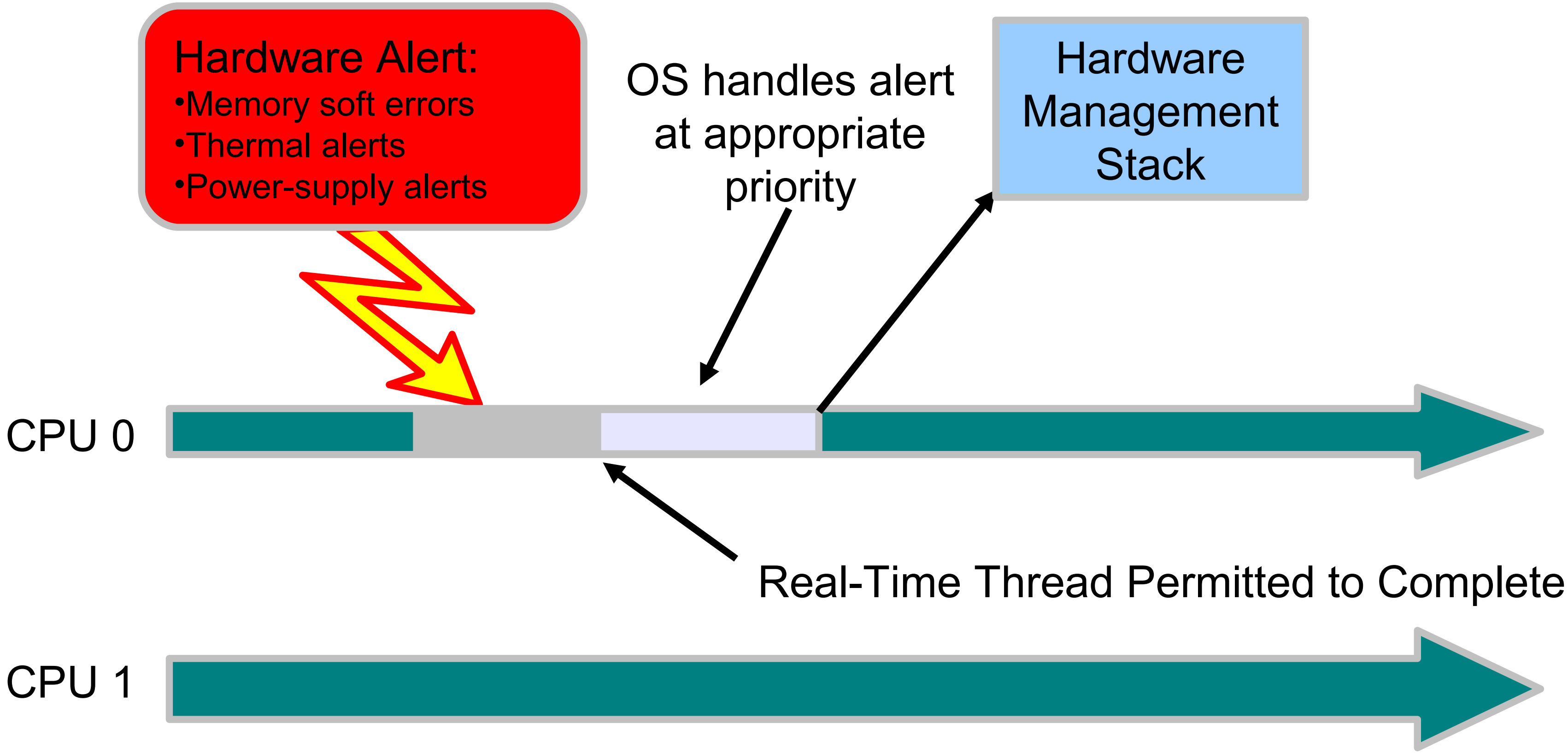
HW/FW Handles Alert

Non-real-time behavior!!!

CPU 1

There is nothing that the OS or higher-level software can do to make up for this HW/FW non-realtime behavior.

IBM xSeries Real-Time Hardware Error Behavior


Hardware Alert:
•Memory soft errors
•Thermal alerts
•Power-supply alerts
OS handles alert at appropriate priority
Hardware Management Stack
CPU 0
Real-Time Thread Permitted to Complete
CPU 1

The OS and higher-level software now see Real-Time behavior.

Hardware Health is Not Abandoned

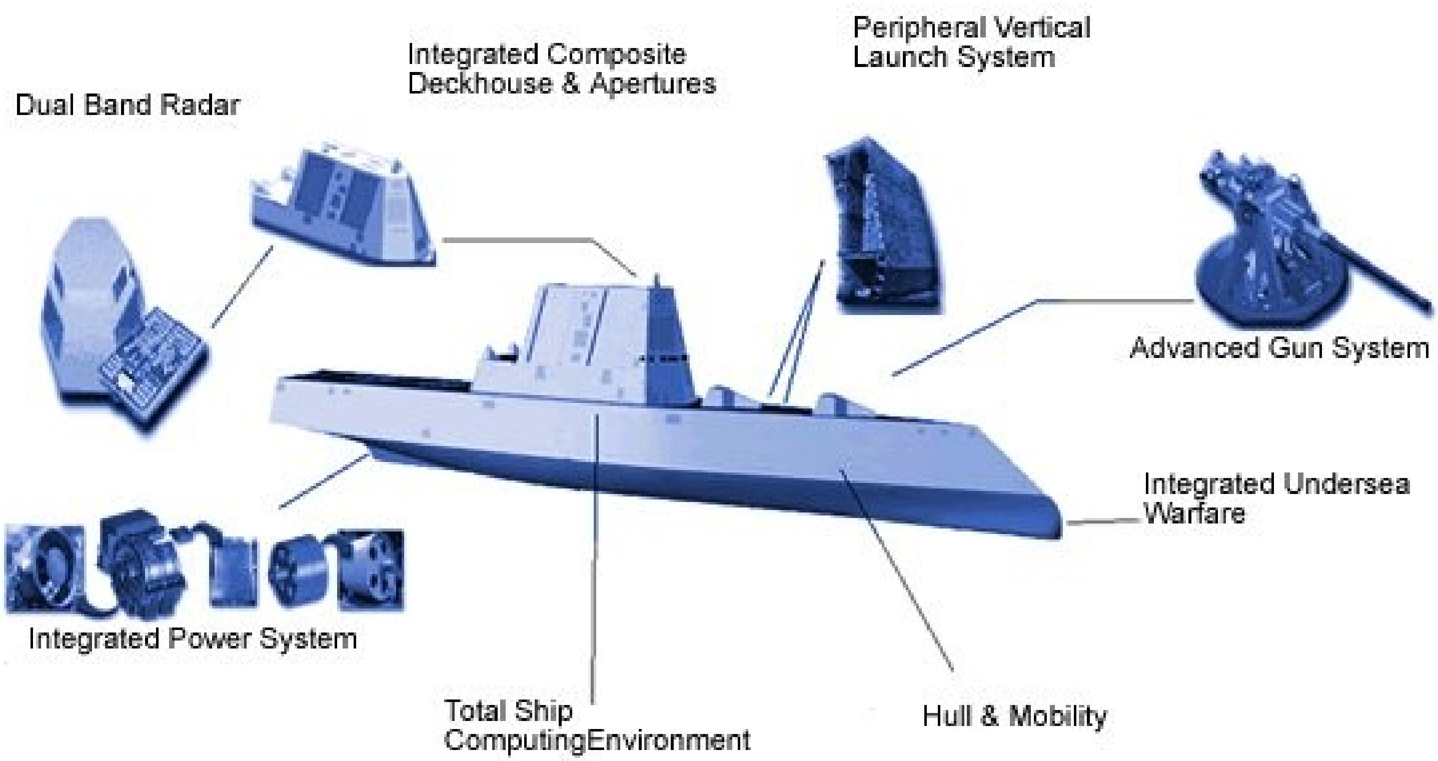
Thermal Considerations:

- System will not throttle the system in an over temp situation
- System will do a hard shutdown at critical temperature
- SNMP and polling of the hardware can provide temperature status information

Power Considerations:

- Systems in real time mode will not automatically throttle to reduce power usage

WRT V2 In The Real World


Dual Band Radar
Integrated Composite Deckhouse & Apertures
Peripheral Vertical Launch System
Advanced Gun System
Integrated Undersea Warfare
Integrated Power System
Total Ship ComputingEnvironment
Hull & Mobility

http://findarticles.com/p/articles/mi_pwwi/is_200702/ai_n17168257

http://www.raytheon.com/capabilities/products/zumwalt/index.html

WebSphere Real-Time V2

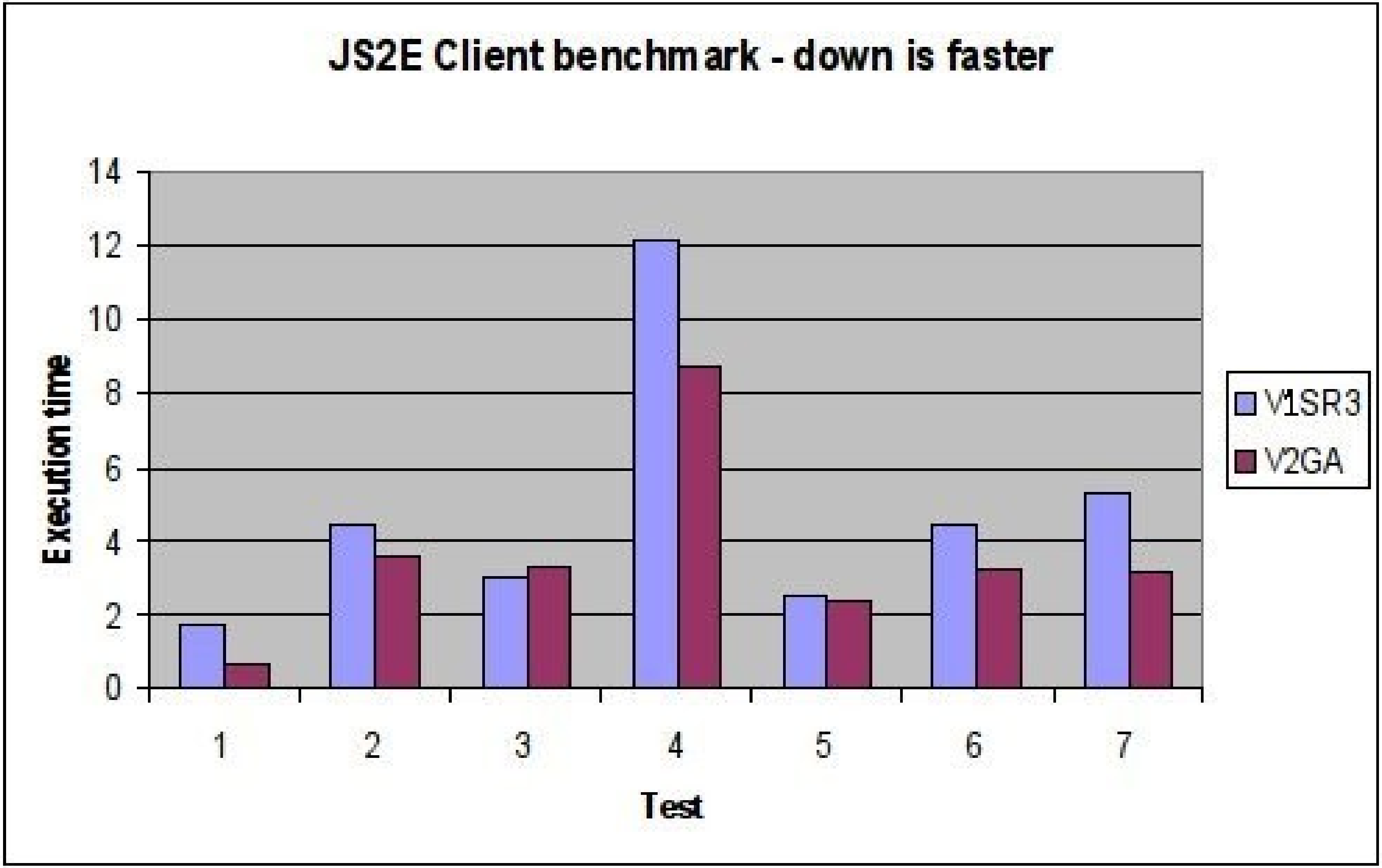
WRT V2 is Generally Available (October 31st, 2008)

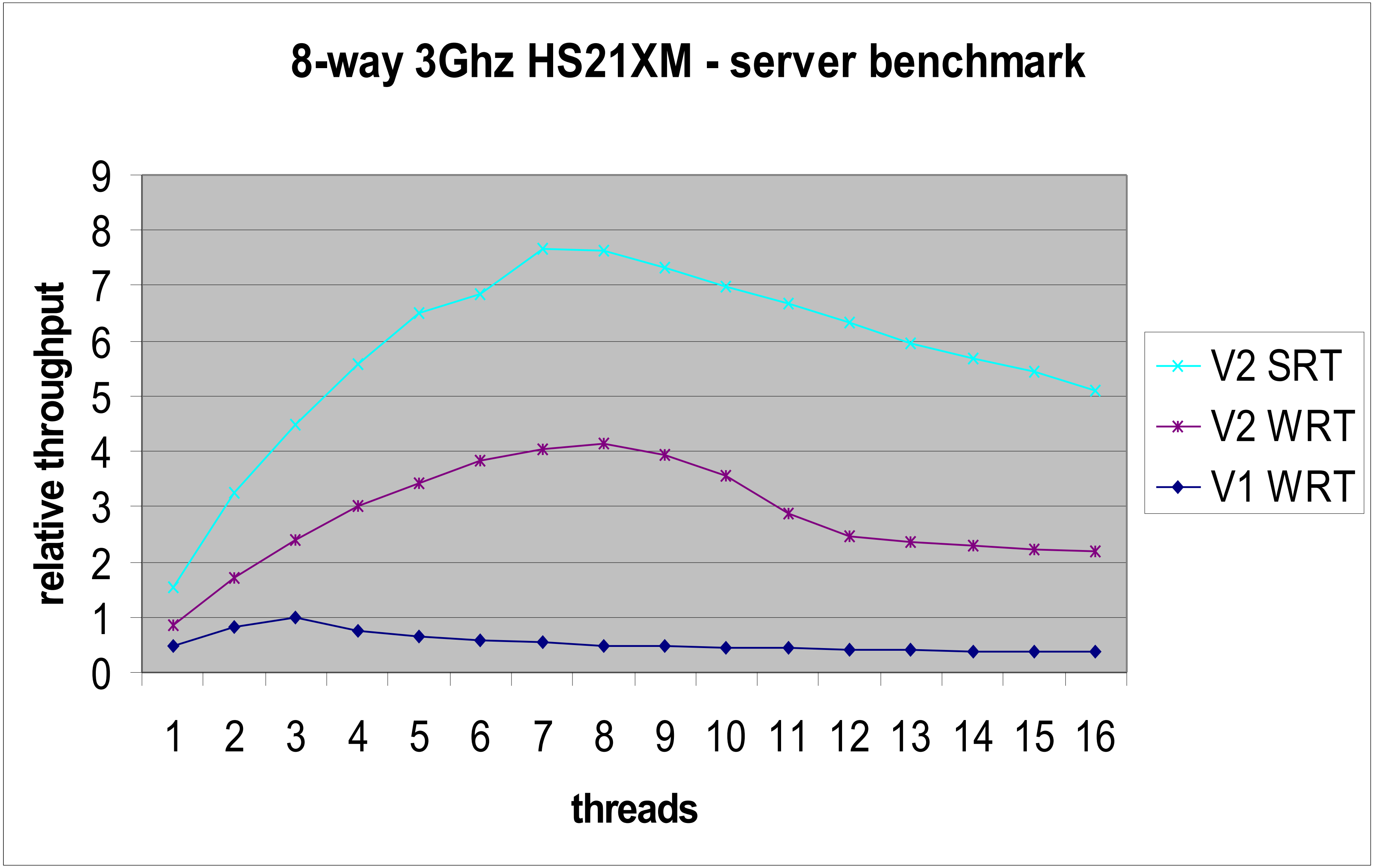
- Support for the latest RTSJ (1.0.2) [up from RTSJ 1.0.1B in WRT V1]
- Support for the latest JSE (Java 6) [up from Java 5 in WRT V1]
- Throughput/scalability improvements
  - Specifically in compilation and garbage collection
  - Exploitation of the largest xSeries blades
- Support for the latest xSeries blades, Red Hat and Novell RT distros
- Mixed AOT/JIT/Interpreter with shared classes

- ***Soft* Real-Time Offering now available for Standard x86 Linux**

Distros

- Available either stand-alone or combined with IBM WebSphere Virtual Enterprise
- Provides Deterministic JVM without RTSJ for JSE 6 applications


JS2E Client benchmark - down is faster
Execution time
14
12
10
8
6
4
2
0
1
2
3
4
5
6
7
Test
V1SR3
V2GA


8-way 3Ghz HS21XM - server benchmark
relative throughput
0
1
2
3
4
5
6
7
8
9
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
threads
V2 SRT
V2 WRT
V1 WRT

Summing up: What makes WRT tick (and tock)?

J9 JVM technology

- IBM-authored virtual machine used in all IBM products and platforms
- Leadership performance, scalability and reliability

Optimizing compilation

- Static (aka ahead-of-time - AOT) compilation for predictable performance
- Dynamic (aka just-in-time - JIT) compilation for best performance (running on a low priority thread)

RTSJ

- Fully compliant to latest level
- Includes fixed priority scheduling, priority inheritance, asynchronous event handling, scoped and immortal memory management

Metronome

- Real-time garbage collection with 1ms worst case pause time

Linux

- RHE MRG, SLERT , RHEL, SLES
- Updated (open source) kernel and libraries engineered for real-time

Hardware

Hard Real-Time Certified on xSeries Hardware
Soft Real-Time supported on most Intel/AMD hardware

More Reading Material (For Cut-n-Paste)

DeveloperWorks Articles:

- 6 Part Series: http://www.ibm.com/developerworks/views/java/libraryview.jsp?search_by=Real+time+Java+Part

AlphaWorks Site:

- RaTCAT, TuningFork, XRTs: http://www.alphaworks.ibm.com/topics/realtimejava

IBM Real-Time Research:

- Metronome & more: http://domino.research.ibm.com/comm/research_projects.nsf/pages/metronome.javiator.html

Announce Page:

- http://www-01.ibm.com/common/ssi/index.wss?DocURL=http://www-01.ibm.com/common/ssi/rep_ca/7/897/ENUS208-

WebSphere Real Time Books:

- General Web Site: https://infocenters.hursley.ibm.com/java/index.jsp