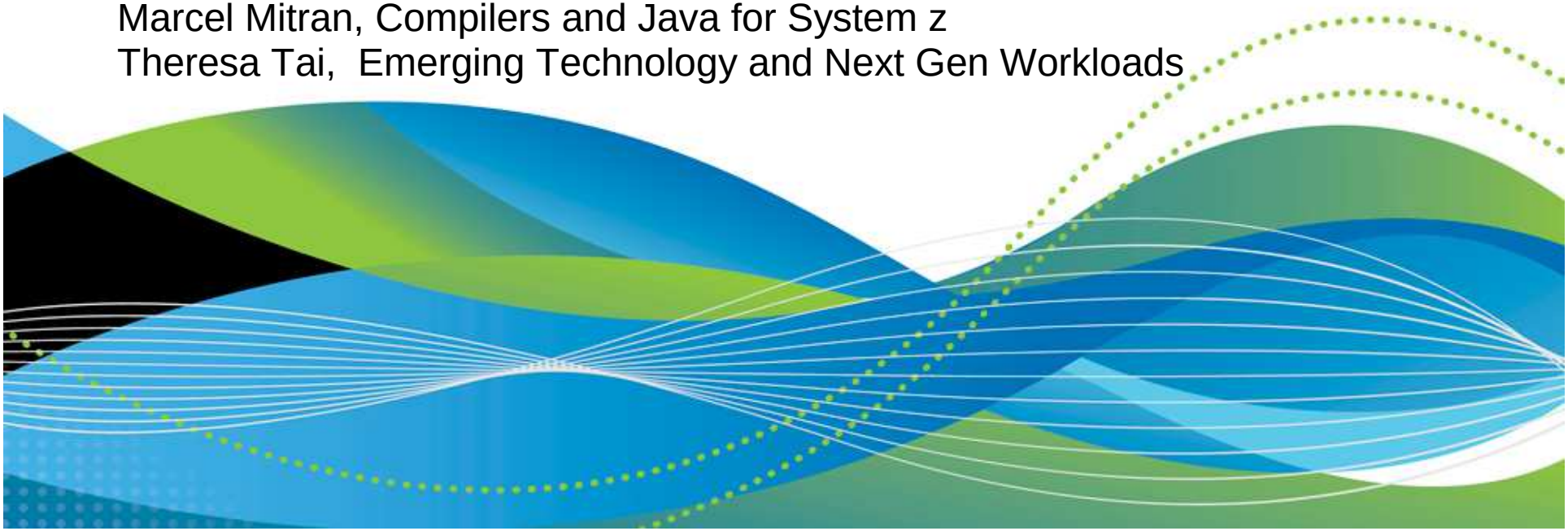
#SHAREorg

SHARE
Technology · Connections · Results

IBM Java 6.0.1 and 7.0 in zEnterprise and Technology Update

Session 10662

Ken Irwin, IBM, zOS Java L2 Service Support
Marcel Mitran, Compilers and Java for System z
Theresa Tai, Emerging Technology and Next Gen Workloads



Trademarks, Copyrights, Disclaimers

Content

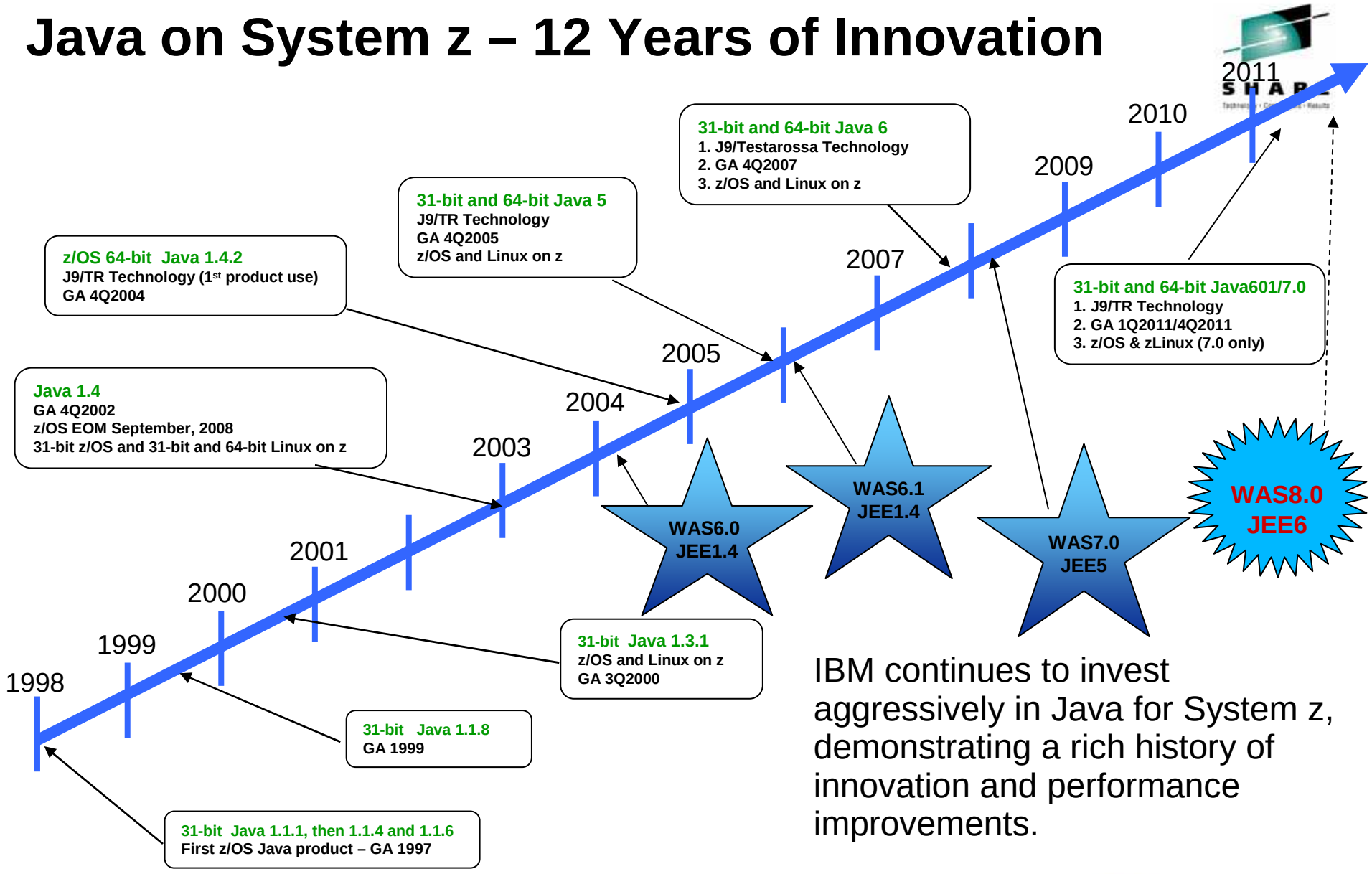
- IBM Java on Z (Marcel)
  - History, overview and roadmap
  - J9 R26 for Java601 and Java7
  - z196 exploitation and performance

- Java consumability and serviceability (Ken)

IBM and Java

- **Java is critically important to IBM**
  - Fundamental infrastructure for IBM's software portfolio
  - WebSphere, Lotus, Tivoli, Rational, Information Management (IM)
- **IBM is investing strategically for Java in Virtual Machines**
  - As of Java 5.0, single JVM support
    - JME, JSE, JEE
  - New technology base (J9/TR Compiler) on which to deliver improved performance, reliability, serviceability
- **IBM also invests in, and supports public innovation in Java**
  - Eclipse, Apache (XML, Aries, Derby, Geronimo, Harmony, Tuscany, Hadoop …)
  - Broad participation in relevant open standards (JCP, OSGi)

Java on System z – 12 Years of Innovation

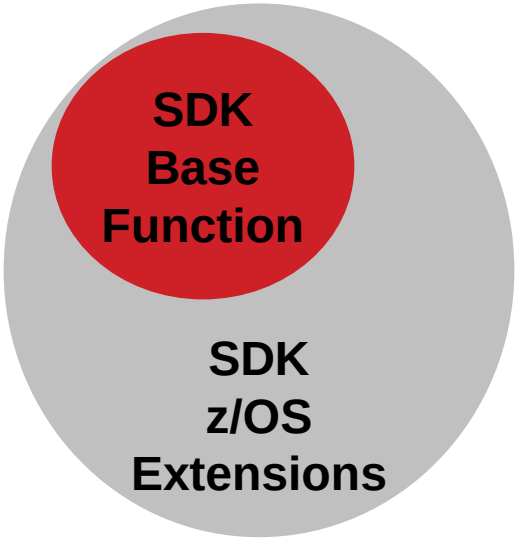


IBM continues to invest aggressively in Java for System z, demonstrating a rich history of innovation and performance improvements.

Testimonials: http://www-01.ibm.com/software/os/systemz/testimonials/
http://www.centerline.net/review/#/3332_B

Timelines and deliveries are subject to change.

z/OS – System z Java Extensions



All SDKs support the ‘standards’, Java on z/OS extends the SDK

- Access to z/OS services
- Access to all types of data
- Access under control of z/OS security mechanisms
- Integration into existing operational infrastructure

Services available in JEE and JSE environments under the restrictions of the container.

System specific extension allow you to write robust middleware and applications that integrate with traditional z/OS operating environment

- Allow for maintaining platform independent design development.
- Platform specific implementations when required
- Allows for operational and resource optimization

e.g. JAAS wrapper of SAF (RACF, ACF2, or TopSecret), Traditional OS dataset access, Cryptographic hardware (Cards and CPACF), z/OS Console (modify and messages), z/OS system logger, JES job submission, DFSORT, SMF, etc.

Java Execution Environments and Interoperability

Capitalize on Pre-existing Assets, Artifacts, Processes, Core Competencies and Performance Strengths.

- **IBM Java Execution Offerings**
  - Transactional/Interactive
    - WebSphere for z/OS (WAS z/OS)
    - WebSphere Process Server for z/OS (WPS)
    - JCICS
    - IMS Java
    - DB2 Stored Procedures
  - Batch oriented
    - WebSphere Compute Grid (WAS-CG)
      - *WAS/JEE runtime extensions*
    - JZOS component of z/OS SDK
      - *JES/JSE-based environment*
    - z/OS V1R13 Java/COBOL Batch Runtime Env.*
      - *JES/JSE-based, designed to inter-op with DB2 while maintaining transaction integrity*

- **Open Source or non-IBM vendor Application Server and Frameworks**
  - Tomcat, JBoss
  - iBatis, Hibernate, Spring
  - Ant

- **COBOL/Native Interoperability**
  - COBOL Invoke maps to JNI
  - RDz and JZOS** have tooling to map COBOL copy books to Java classes
  - JCICS
  - IMS Java, JMP/JBP
  - WAS CG, WOLA
  - etc

* See http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=ca&infotype=an&supplier=897&letternum=ENUS211-252

** Alphaworks only, and hence currently un-supported

The New z/OS Batch Runtime Environment

- z/OS V1.13 is intended to be the foundation for developing "real-time batch" applications that enable concurrent batch and online data access
  - A new z/OS base component
  - Provides the framework for
    - Java-to-COBOL interoperability
    - Transactional updates to DB2®
    - Sharing database connections between Java and COBOL
  - New Java-COBOL interoperability capabilities are designed to enable re-use valuable COBOL assets by developing new and/or enhancing existing batch applications with Java
    - An example - use Java subroutines directly in lieu of Java stored procedures
  - Leverage Specialty Engine zAAP

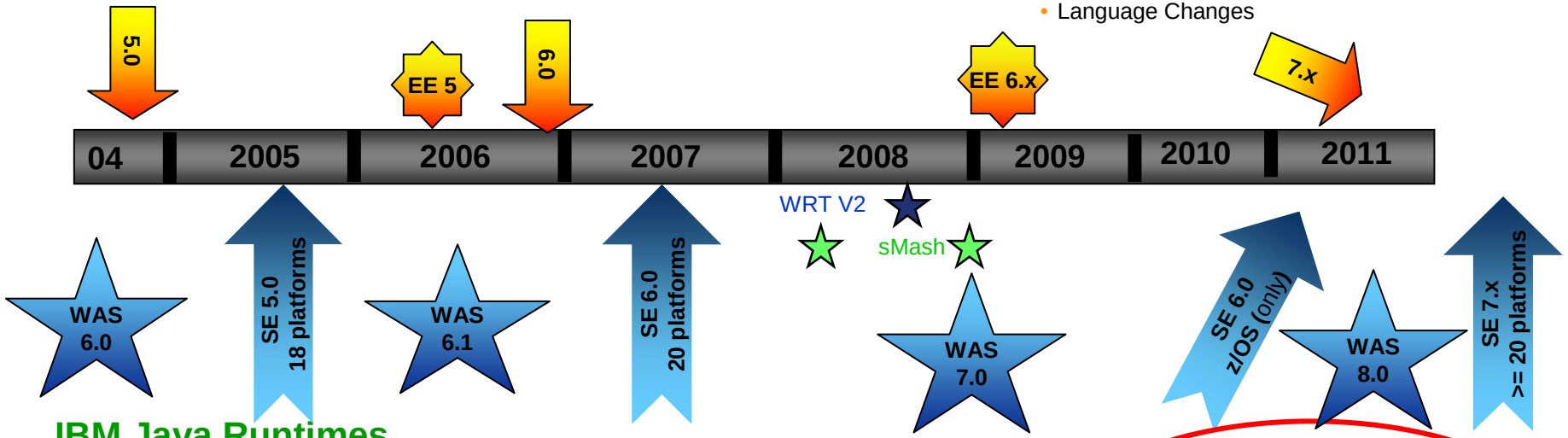
Java Road Map

Oracle Java Runtimes

Java 5.0
- New Language features:
  - Autoboxing
  - Enumerated types
  - Generics
  - Metadata

Java 6.0
- Performance Improvements
- Client WebServices Support

Java 7.0
- Support for dynamic languages
- Improve ease of use for SWING
- New IO APIs (NIO2)
- Java persistence API
- JMX 2.x and WS connection for JMX agents
- Language Changes



IBM Java Runtimes

IBM Java 5.0 (J9R23)
- Improved performance
  - Generational Garbage Collector
  - Shared classes support
  - New J9 Virtual Machine
  - New Testarossa JIT technology
- First Failure Data Capture
- Full Speed Debug
- Hot Code Replace
- Common runtime technology
  - ME, SE, EE

IBM Java 6.0 (J9R24)
- Improvements in
  - Performance
  - Serviceability tooling
  - Class Sharing
- XML parser improvements
- z10™ Exploitation
  - DFP exploitation for BigDecimal
  - Large Pages
  - New ISA features

IBM Java 6.0.1/Java7.0 (J9R26)
- Improvements in
  - Performance
  - GC Technology
- z196™ Exploitation
  - OOO Pipeline
  - 70+ New Instructions
- JZOS/Security Enhancements

Timelines and deliveries are subject to change.

Java 7.0 – What to look for

New I/O

- Meets the increasingly I/O intensive demands of data mining and analytics workloads
- Significant performance and footprint gains from async I/O

Concurrency Libraries

- Exploit larger multi-core systems, such as next generation Power and System z, by providing better scalability, higher throughput and lower total cost of ownership from server consolidations

Dynamic language support

- Leverage the advantages of a single runtime for dynamic language applications written in PHP, Groovy, jRuby and jython

Language improvements

- Improved efficiency through simplified day-to-day programming tasks
- Protect developer commitment to, and customer/ISV investment in, the Java ecosystem.

IBM Java Runtime Environment

- IBM's implementation of Java 5, Java 6 and java7 are built with **IBM J9 Virtual Machine** and **IBM Testarossa JIT Compiler** technology
  - Independent clean-room JVM runtime & JIT compiler

- Combines best-of breed from embedded, development and server environments… from a cell-phone to a mainframe!
  - Lightweight flexible/scalable technology
  - World class garbage collection – gencon, balanced GC policies
  - Startup & Footprint - Shared classes, Ahead-of-time (AOT) compilation
  - 64-bit performance - Compressed references & Large Pages
  - Deep System z exploitation – z196/z10/z9/z990 exploitation
  - Cost-effective for z - zAAP Ready!

- Millions of instances of J9/TR compiler

Java601/Java7 Executive Summary

z196 and Java6.0.1/Java7: Engineered Together

- Up-to 2.1x improvement to Java throughput
- Reduced footprint
- Tighter integration with z/OS facilities
- Improved responsiveness in application behavior

J9 R2.6 Virtual Machine

- Significant enhancements to JIT optimization technology
- z196 exploitation of instructions and new pipeline
- New Balanced GC policy to reduce max pause times
- Default GC policy changed to gencon

z/OS Unique Enhancements

- JZOS 2.4.0
- z/OS Java unique security enhancements

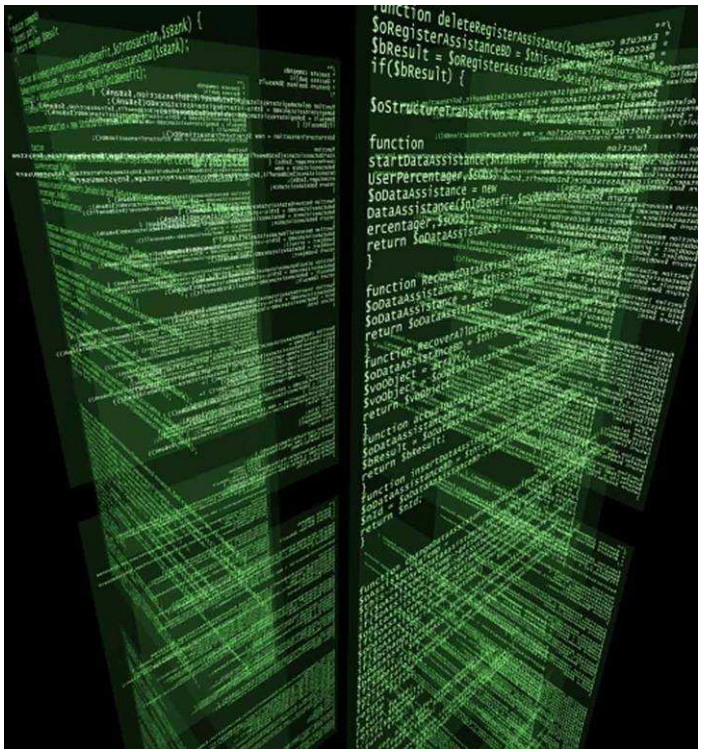
Performance

- 2.1x improvement to multi-threaded workload
- 1.93x improvement to CPU-intensive workload

z/OS Unique functional enhancements

JZOS 2.4.0

- Enables the submission of jobs to the MVS internal reader and retrieval of the submitted JOB ID
  - Text, record, and binary JCL will be supported.
  - The MVS internal reader attributes LRECL, RECFM, and Class will be configurable from this Java class.
- Enhances the zlogstream class to support
  - IXGBRWSE (read) and IXGDELT (delete)
  - InputStream/OutputStream Java wrappers
- Added the method ZFile.readDSCBChain() to support
  - Reading all of the DSCBs associated with a dsn/volume
  - Extended access volumes (Format-8 and Format-9 DSCBs.)
- Adds a new package com.ibm.jzos.wlm with selected z/OS Workload Manager (WLM) APIs
- Removes all JRIO code dependencies
  - JRIO is deprecated in z/OS V6.0.1 products
  - Java SE 7 is planned to be the last release with JRIO



X11 Library

- Dynamically links X11 library (Java 7)

z/OS Java Unique Security Enhancements

- IBMJCECCA provider support for AES Secure Keys
- RAS: Provide Enhanced ICSF Exception Handling in IBMJCECCA

IBM J9 Garbage Collector

- **IBM J9 VM garbage collector family**
  - Parallel global (mark, sweep, compact)
  - Generational collection
  - Partial concurrency at global level
  - Use of OS level features (Virtual Memory, large pages, etc)
  - Type accurate stacks, cooperative suspend
  - No pinned objects for JNI (less heap fragmentation)
  - Tunable garbage collection policies to best match application behaviour

Policy	Recommended usage	Notes
`optThroughput`	optimized for throughput	default in Java5 and Java6
`optAveragePause`	optimized to reduce pause times	
`gencon`	optimized for transactional workloads	default in Java601/Java7
`subPools`	optimized for large MP systems	deprecated in Java601/Java7
`balanced`	optimized for large heaps	added in Java601/Java7

IBM J9 2.6 Technology Enhancements:
Garbage Collection: Balanced Policy

Improved responsiveness in application behavior

- Reduced maximum pause times to achieve more consistent behavior
- Incremental result-based heap collection targets best ROI areas of the heap
- Native memory aware approach reduces non-object heap consumption

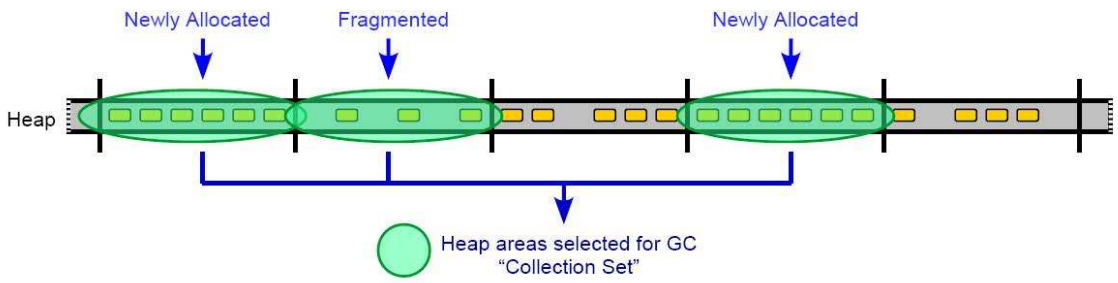
Next generation technology expands platform exploitation possibilities

- Virtualization – Group heap data by frequency of access, direct OS paging decisions
- Dynamic reorganization of data structures to improve memory hierarchy utilization (performance)

Recommended deployment scenarios

- Large (>4GB) heaps
- Frequent global garbage collections
- Excessive time spent in global compaction
- Relatively frequent allocation of large (>1MB) arrays

Input welcome: Help set directions by telling us your needs



IBM J9 2.6 and z196

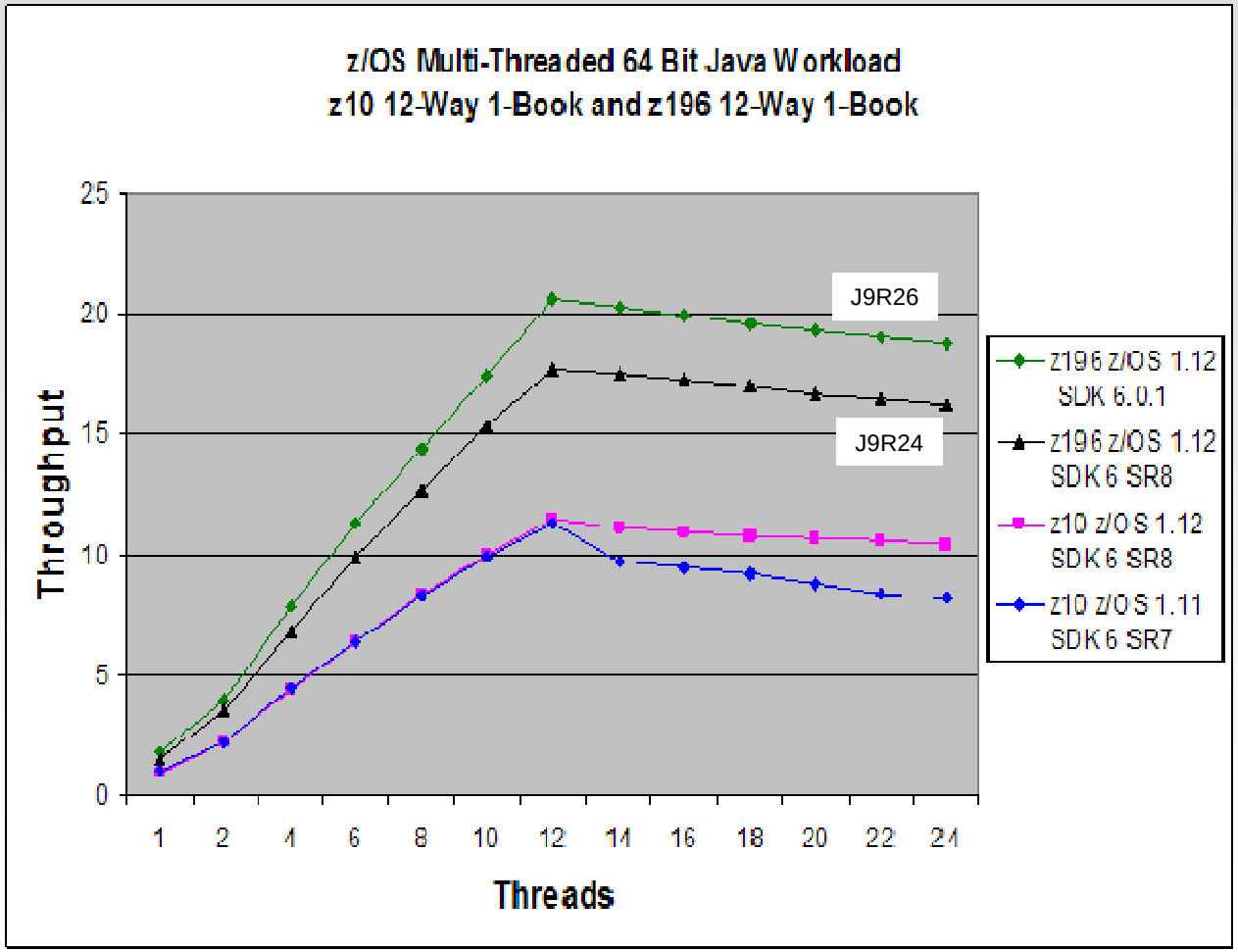
z196: Hardware for Java

- New Out-Of-Order pipeline design
- New larger cache structure
- Higher clock speed (~5.2GHz)

J9 R26: JRE for z196

- Reducing pressure on the data/instruction cache
  - Enables better exploitation of new OOO compute bandwidth
  - Mitigates effects of cache latencies for leveraging core speed
- Concurrency improvements
  - Better scalability
- General optimizer and codegen improvements
  - Reduced path-length

z/OS Java SDK 7: Multi-threaded Benchmark



2.1x Aggregate improvement

- **16% Java 6.0.1 improvement**
- **56% Hardware improvement**
- **17% z/OS 1.12 improvement****

Note: Java 7 and Java 6.0.1 have almost identical performance for this workload

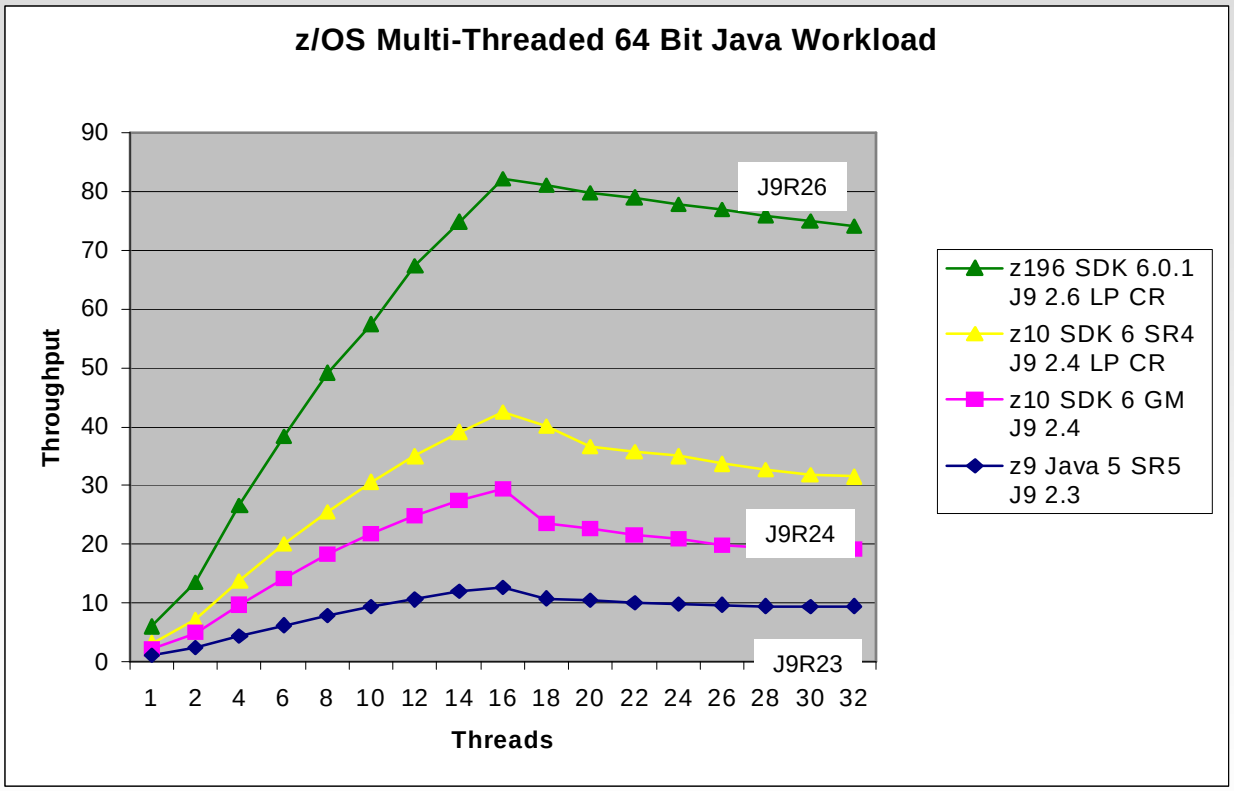
CR = Compressed Reference
LP = Large Pages

**using IEASYSxx new option TIMESLICE=64

(Controlled measurement environment, results may vary)

z/OS Java SDK 7: 16-Way Performance

Aggregate HW and SDK Improvement z9 Java 5 SR5 to z196 Java 6.0.1 and Java 7



~7x Improvement from z9, Java5 SR5 to z196 Java 6.0.1 and Java 7

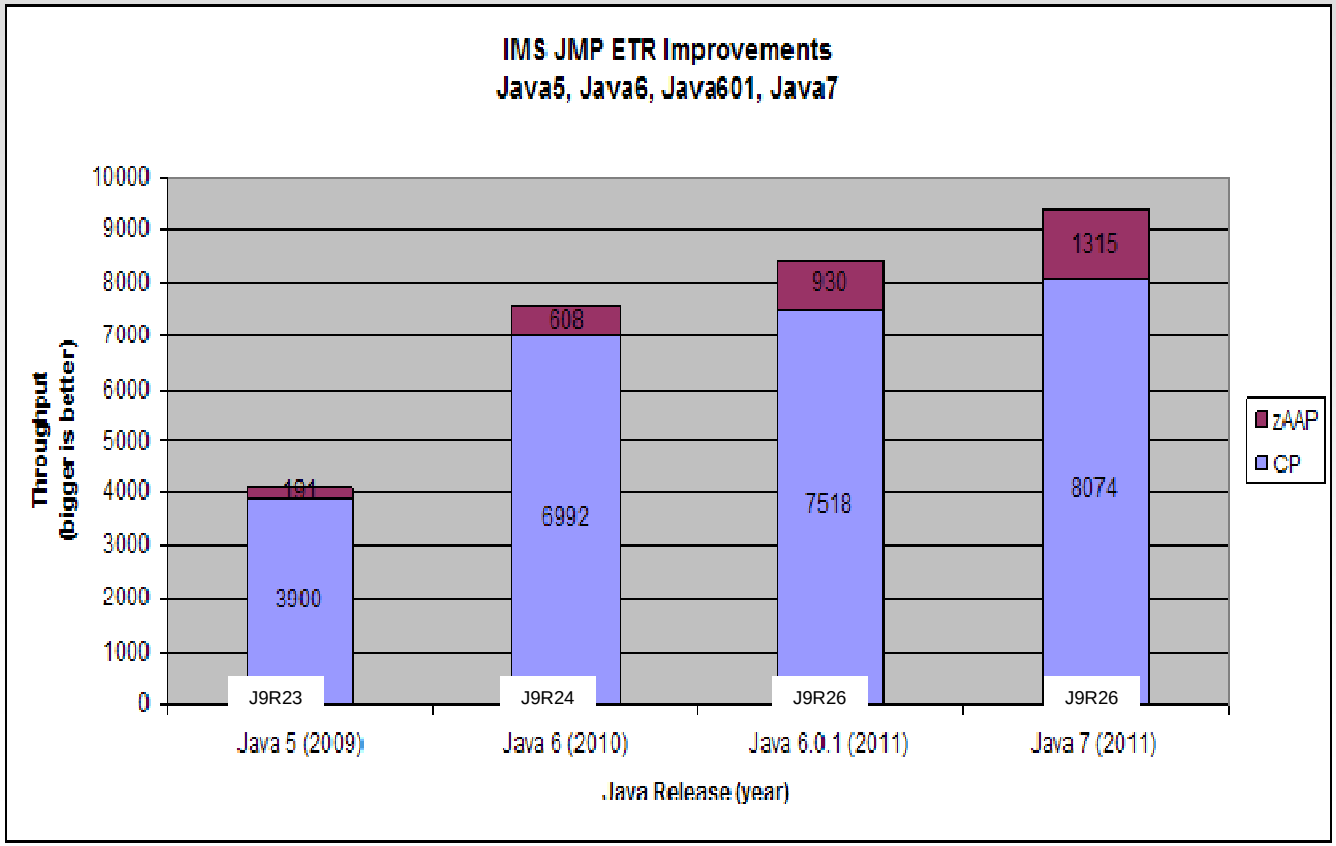
Note: Java 7 and Java 6.0.1 have almost identical performance for this workload

CR = Compressed References

LP = Large Pages

(Controlled measurement environment, results may vary)

z/OS Java SDK 6.0.1 and 7.0: IMS JMP



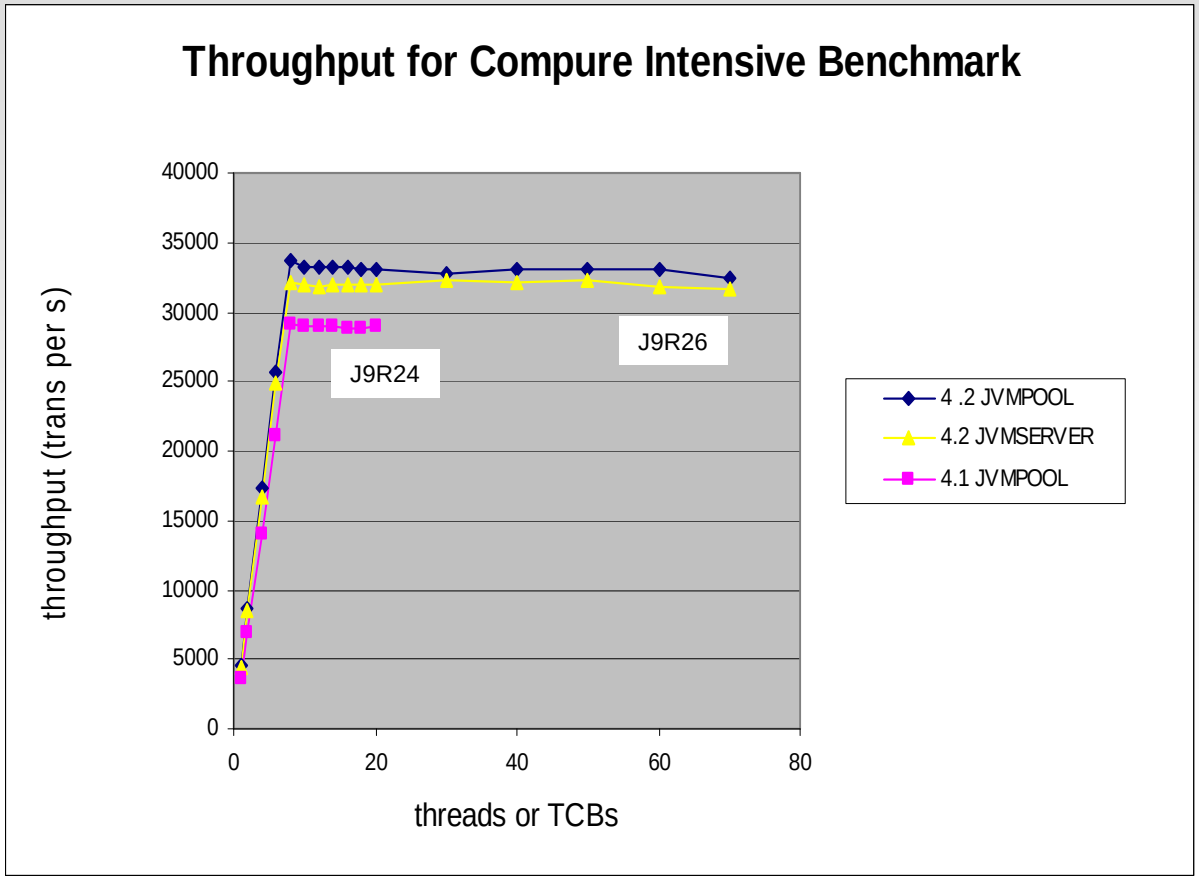
~2.5x aggregate throughput improvement from Java5 to Java7

2 GCP + 2 zAAP

(Controlled measurement environment, results may vary)

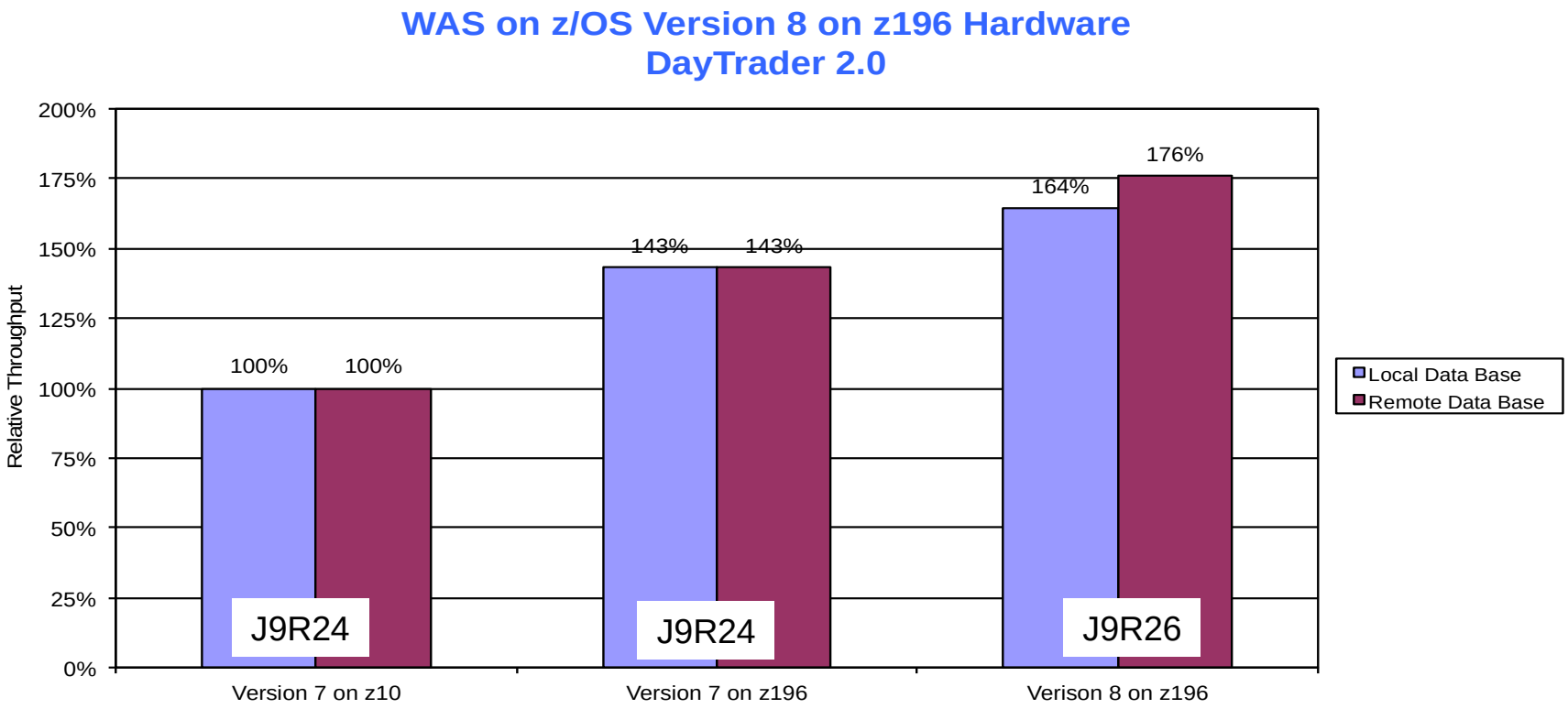
z196™ – z/OS V1.12

Comparing Java Throughput on CICS 4.2 with CICS 4.1



- **CICS 4.2 JVMPOOL and JVMSERVER use 64-bit Java 6.0.1 relieving 31-bit storage constraint**
- **~17% improvement to throughput with CICS 4.2/Java 6.0.1**
- **JVMSERVER slightly more expensive than JVMPOOL in CPU usage but requires less memory**
- **All configurations scale well**

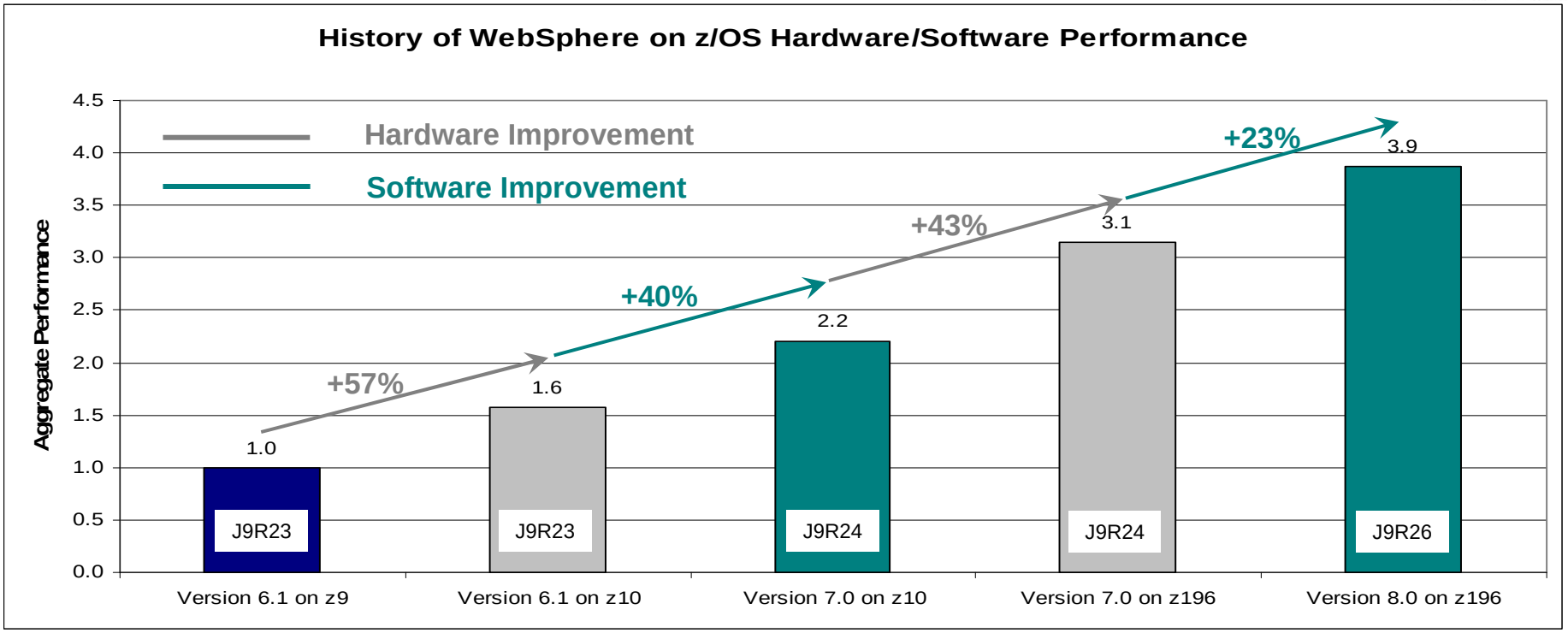
Performance on z/OS: WAS on z/OS



- z196 hardware measured 43% more throughput for local and remote database configurations
- Version 8 improved throughput of 2-tier configurations by another 15% for an aggregate benefit of 64%
- Version 8 improved throughput of 3-tier configurations by another 23% for an aggregate benefit of 76%

(Controlled measurement environment, results may vary)

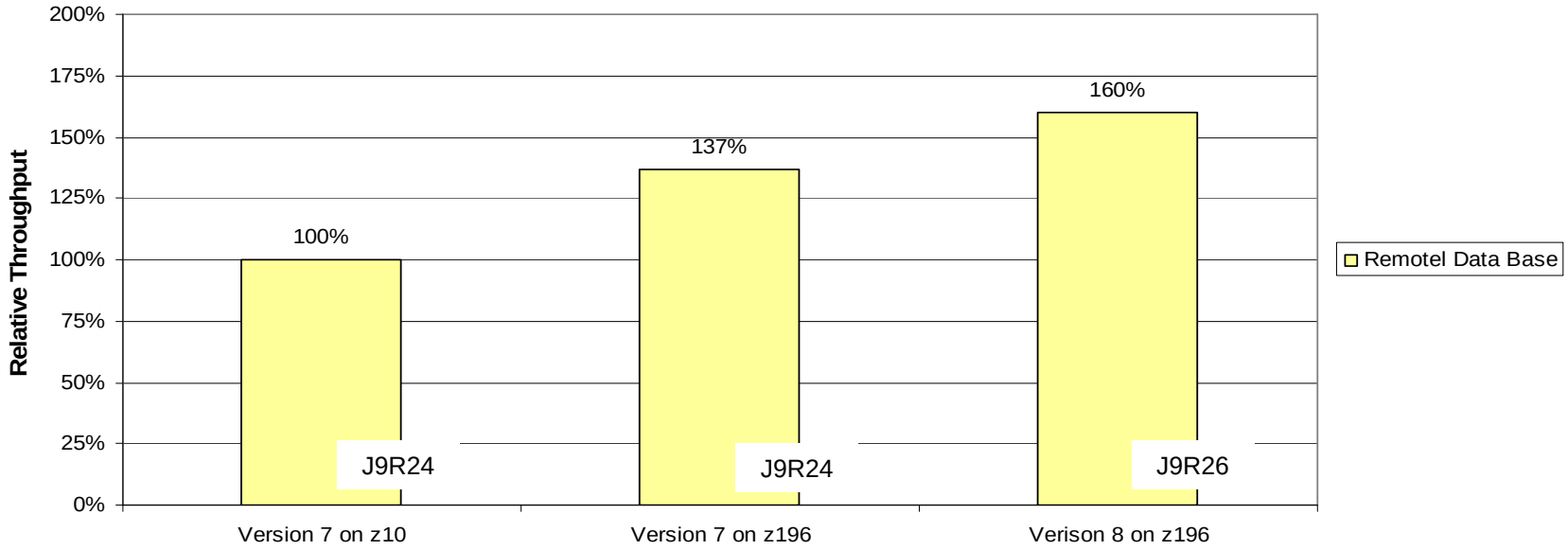
Performance on z/OS: WAS on z/OS



- This chart shows a history of improvements made from zSeries hardware (from z9 to z196) and software (from V6.1 to V8.0). The data is from measurements done using the DayTrader EJB workload.
  - The chart shows an aggregate performance improvement of almost 4x moving from WAS V6.1 on a z9 to WAS V8.0 on a z196.
  - The hardware component of this increase is about 2.25x (1.57 x 1.43)
  - The software component is about 1.72x (1.40 x .123)

(Controlled measurement environment, results may vary)

Performance – WAS8.0 on zLinux

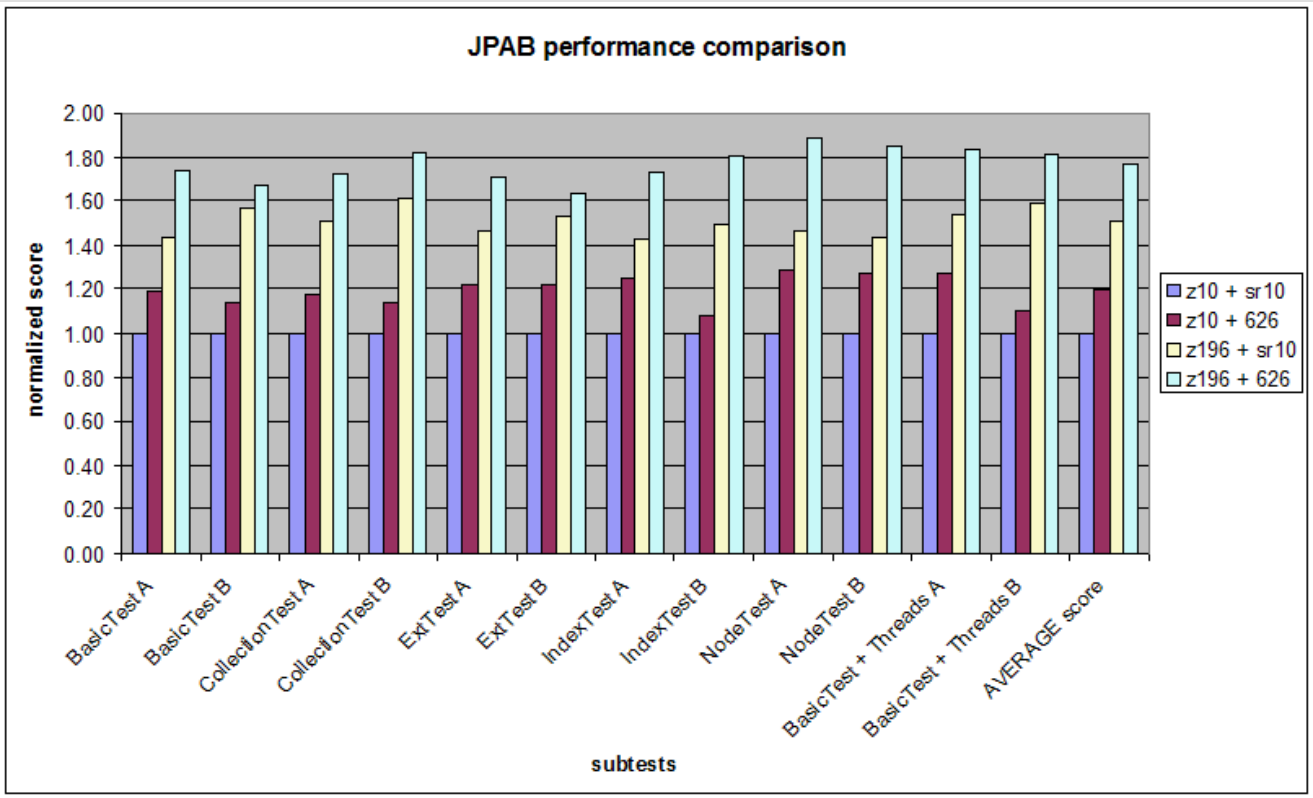


- Upgrading from z10 to z196 improved throughput by 37% using our DayTrader 2.0 EJB benchmark.
- Additionally, upgrading to WAS V8.0 improved performance by another 17%. This increase is a result of improvements to the following areas:
  - JVM and JIT optimizations
  - OpenJPA code paths
- The combine hardware and software improvement is 60%.

(Controlled measurement environment, results may vary)

Java Persistency API and J9 R26

- JPAB benchmarks running OpenJPA and HSQLDB (http://www.jpab.org/OpenJPA/HSQLDB/embedded.html)



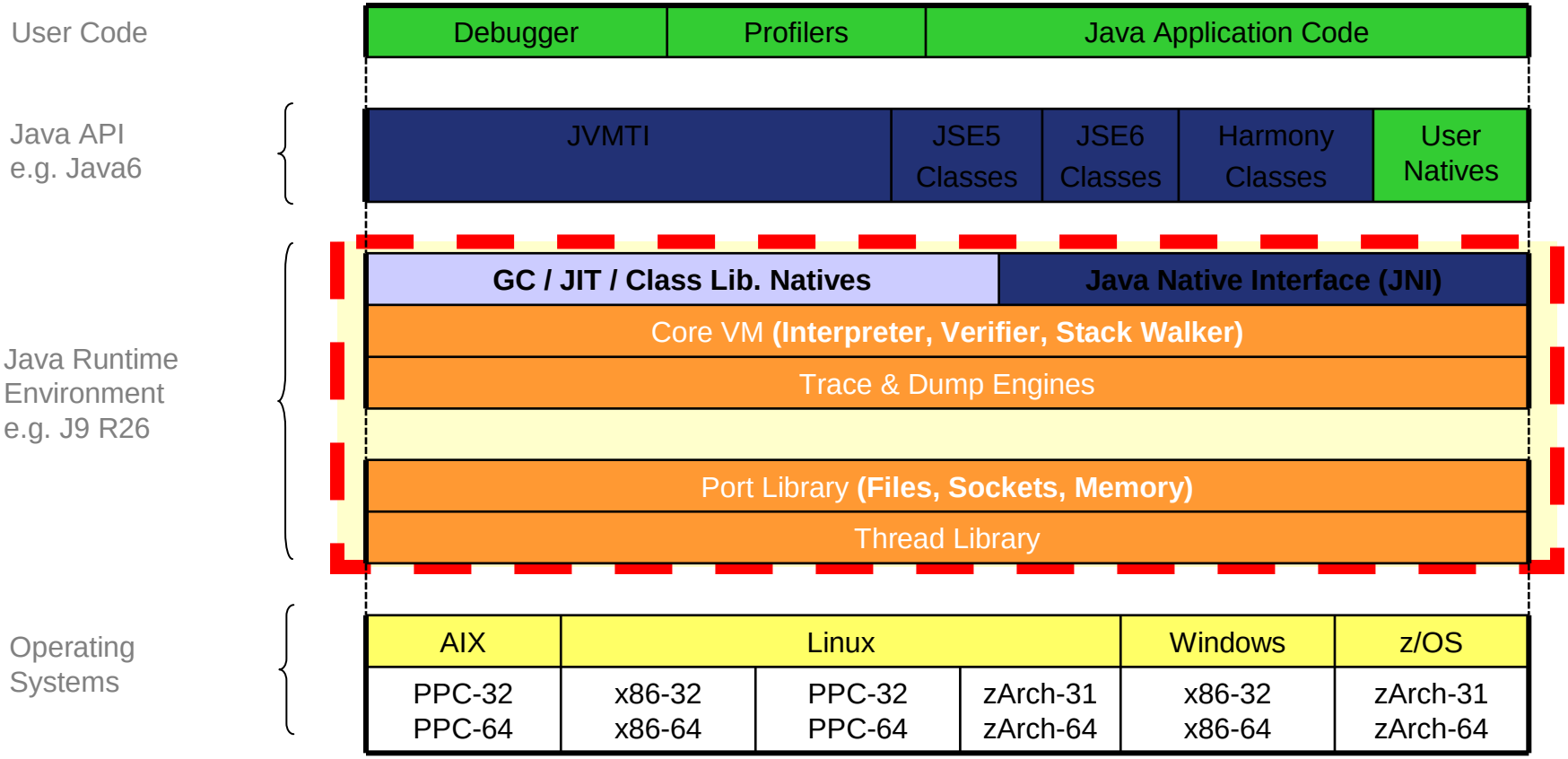
1.77x Aggregate software/ hardware improvement

- 19% Java6R26 vs Java6R24 improvement on z10
- 26% Java6R26 vs Java6R24 improvement on z196
- 51% improvement z10 to z196

(Controlled measurement environment, results may vary)


Questions

JVM Architectural Overview



= User Code
= Java Platform API
= VM-aware
= Core VM

Java 6.0.1:

- Also referred to as Java6 R26, ships with WAS8 across platforms, or standalone on z/OS
- Fully compatible/compliant Java6 (JSE6)
- Includes new J9 R26 JRE (replacing J9 R24 in Java6.0.0)
  - Transparent z196 and new optimization exploitation
  - New balanced GC policy

New and existing supported Java products – z/OS

- IBM 31-bit SDK for z/OS, Java Technology Edition, Version 7.0
  - Web available on October 4, 2011 at Java SE 6 level
  - Product 5655-R31, supported on z/OS V1.10 and above
- IBM 64-bit SDK for z/OS, Java Technology Edition, Version 7.0
  - Web available on October 4, 2011 at Java SE 6 level
  - Product 5655-R32, supported on z/OS V1.10 and above
- Earlier Deliveries
  - IBM 31-bit SDK for z/OS, Java Technology Edition, Version 6.0.1
    - Web available on March 15, 2011 at Java SE 6 level
    - Product 5655-R31, supported on z/OS V1.10 and above
  - IBM 64-bit SDK for z/OS, Java Technology Edition, Version 6.0.1
    - Web available on March 15, 2011 at Java SE 6 level
    - Product 5655-R32, supported on z/OS V1.10 and above
  - IBM 31-bit SDK for z/OS, Java Technology Edition, Version 6.0.0
    - Web available on December 14, 2007 at Java SE 6 level
    - Product 5655-R31
  - IBM 64-bit SDK for z/OS, Java Technology Edition, Version 6.0.0
    - Web available on December 14, 2007 at Java SE 6 level
    - Product 5655-R32
  - IBM 31-bit SDK for z/OS, Java 2 Technology Edition, Version 5.0
    - Web available on November 30, 2005
    - Product 5655-N98
  - IBM 64-bit SDK for z/OS, Java 2 Technology Edition, Version 5.0
    - Web available on November 30, 2005
    - Product 5655-N99
- All products are delivered via the z/OS Java website in non-SMP/E format and via ShopIBM in SMP/E format
- All products are independently orderable and serviceable and follow the z/OS RFA rules for Withdrawal from Marketing and End of Service

Important references

- z/OS Java web site
  - http://www.ibm.com/systems/z/os/zos/tools/java/
- IBM SDK Java Technology Edition Version 7 Information Center
  - http://publib.boulder.ibm.com/infocenter/java7sdk/v7r0/index.jsp
- IBM SDK Java Technology Edition Version 6 Supplement
  - http://public.dhe.ibm.com/common/ssi/ecm/en/zsl03118usen/ZSL03118USEN.PDF
- JZOS Batch Launcher and Toolkit Installation and User's Guide (SA38-0696-00)
  - For JZOS function included in IBM Java SE 7 SDKs for z/OS
  - http://publibz.boulder.ibm.com/epubs/pdf/ajvc0110.pdf
- JZOS Batch Launcher and Toolkit Installation and User's Guide (SA23-2245-03)
  - For JZOS function included in IBM Java SE 6 and SE 5 SDKs for z/OS
  - http://publibfi.boulder.ibm.com/epubs/pdf/ajvc0103.pdf