

IBM Java News – IBM SDK for Java 8 and z13

Joran Siu – IBM SDK for Java Development

Session 16815

Mon, March 2, 2015: 11:15 AM-12:15 PM



#SHAREorg



SHARE is an independent volunteer-run information technology association
that provides **education, professional networking and industry influence.**



Trademarks, Copyrights, Disclaimers

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

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

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2015. All rights reserved.

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Java™ on System z®?

Naturally.

Two
**pervasive
technologies...**

There are
9 million

Java developers

80% 
of the world's corporate data
resides on or originates on the
mainframe

...combine for
**powerful
performance...**

15%
increase
in application
performance

5x
faster
DB-response
time

20%
greater
processing
capacity

when DATEV eG
ported business
rules from a
distributed server
into CICS® Java



...that
**everybody's
talking about.**

z/OS is probably
the most efficient
place to run Java.

David Hodgson,
techrepublic.com

You put the
code where the
data is, and you
get to remove any
network latency...

Since the z9 was introduced, Java performance has
exploded five times and it hasn't finished on that curve...

Scott Fagen, enterprisesystemsmedia.com

I've been impressed of late with the mainframe's Java
support. It runs fast. It runs on the zAAPs. It runs all
sorts of Java things without any recoding effort.

Scott Chapman, cmg.org

Evolving Java as a Workload Optimized System on Z

Enable integration of Java-based applications with core Z environment for high performance, reliability, availability, security, and lower total cost of ownership

- **Portable and consumable**
 - First-class IBM Java SDK for z/OS and Linux on z
 - Providing seamless portability across platforms
- **Pervasive and integrated across the z eco-system**
 - Java business logic with all z middleware (IMS, CICS, WAS, etc.)
 - Inter-operability with legacy batch and OLTP assets
- **Deep z Systems exploitation**
 - SDK extensions enabled z QoS for full integration with z/OS
 - zAAP/zIIP specialty engines provide low-cost Java capacity
- **Performance**
 - A decade of hardware/software innovations and optimizations
 - Industry leading performance with IBM J9 Virtual Machine
 - Enabling tight data locality for high-performance and simplified systems



IBM Java Runtime Environment

- IBM's implementations of Java 5, 6, 7, 8 are built with **IBM J9 Virtual Machine** and **IBM Testarossa JIT Compiler** technologies
 - 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 z Systems exploitation – z13/zEC12/z196/z10/z9/z990 exploitation
 - Cost-effective for z - zIIP Ready!
- Millions of instances of J9/TR compiler

Java Road Map

Language Updates

Java 5.0

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

Java 6.0

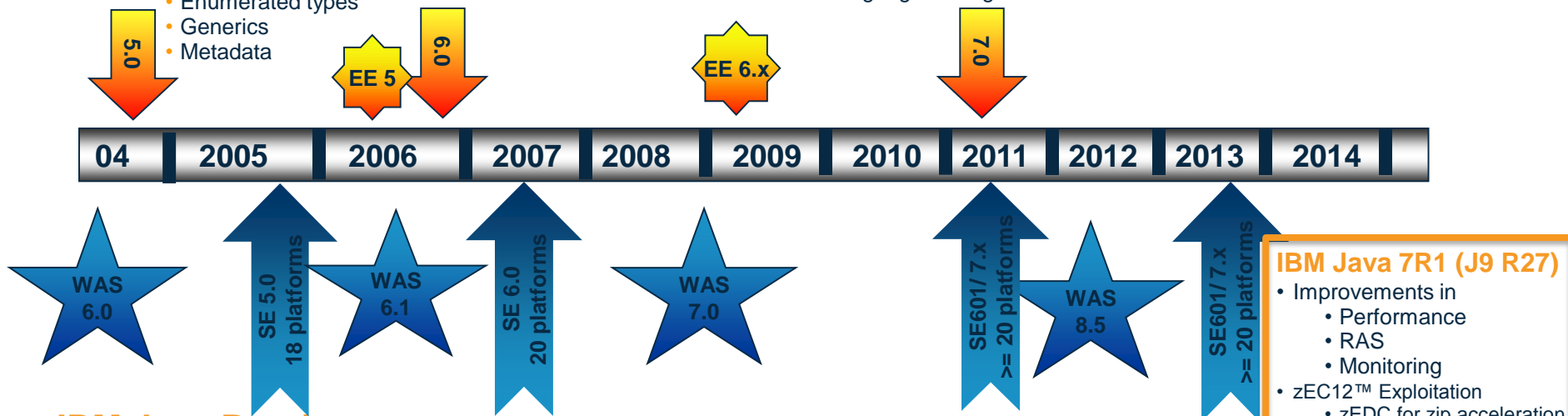
- Performance Improvements
- Client WebServices Support

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

Java 8.0**

- Language improvements
- Closures for simplified fork/join



IBM Java Runtimes

IBM Java 5.0 (J9 R23)

- Improved performance
 - Generational Garbage Collector
 - Shared classes support
 - New J9 Virtual Machine
 - New Testarossa JIT technology
- First Failure Data Capture
- Common runtime technology
 - ME, SE, EE

IBM Java 6.0 (J9 R24)

- 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/Java 7 (J9 R26)

- Improvements in
 - Performance
 - GC Technology
- z196™ Exploitation
 - OOO Pipeline
 - 70+ New Instructions
- JZOS/Security Enhanc

IBM Java 7 (J9 R26 SR3)

- Improvements in
 - Performance
- zEC12™ Exploitation
 - Transactional Exec
 - Flash 1Meg LPs
 - 2G large pages
 - Hints/traps

IBM Java 7R1 (J9 R27)

- Improvements in
 - Performance
 - RAS
 - Monitoring
- zEC12™ Exploitation
 - zEDC for zip acceleration
 - SMC-R integration
 - Transactional Execution
 - Runtime instrumentation
 - Hints/traps
- Data Access Accelerator

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

**Timelines and deliveries are subject to change.

zEC12 – More Hardware for Java

Continued aggressive investment in Java on Z
Significant set of new hardware features tailored
and co-designed with Java

Hardware Transaction Memory (HTM)

Better concurrency for multi-threaded applications
eg. ~2X improvement to `juc.ConcurrentLinkedQueue`

Run-time Instrumentation (RI)

Innovation new h/w facility designed for managed runtimes
Enables new expanse of JRE optimizations

2GB page frames

Improved performance targeting 64-bit heaps

Pageable 1M large pages with Flash Express

Better versatility of managing memory

Shared-Memory-Communication

RDMA over Converged Ethernet

zEnterprise Data Compression accelerator

gzip accelerator

New software hints/directives/traps

Branch preload improves branch prediction
Reduce overhead of implicit bounds/null checks

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

New **5.5 GHz** 6-Core Processor Chip

Large caches to optimize data serving

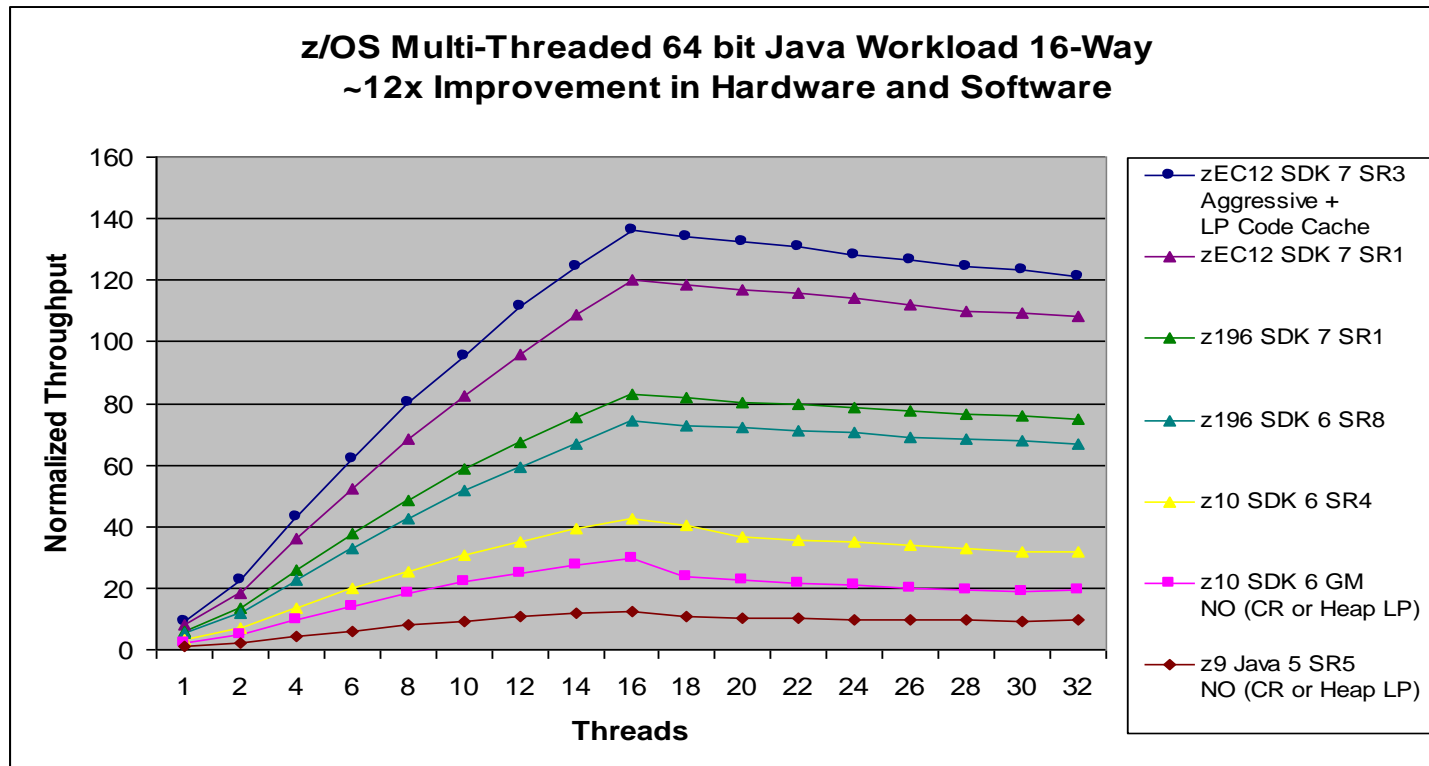
Second generation **OOO design**



Up-to 60% improvement in throughput amongst Java workloads measured with zEC12 and IBM Java 7

z/OS IBM Java 7 on zEC12

- 64-bit Java Multi-threaded Benchmark on 16-way



~12x aggregate hardware + software improvement from IBM Java5SR4 on z9 to IBM Java7 on zEC12

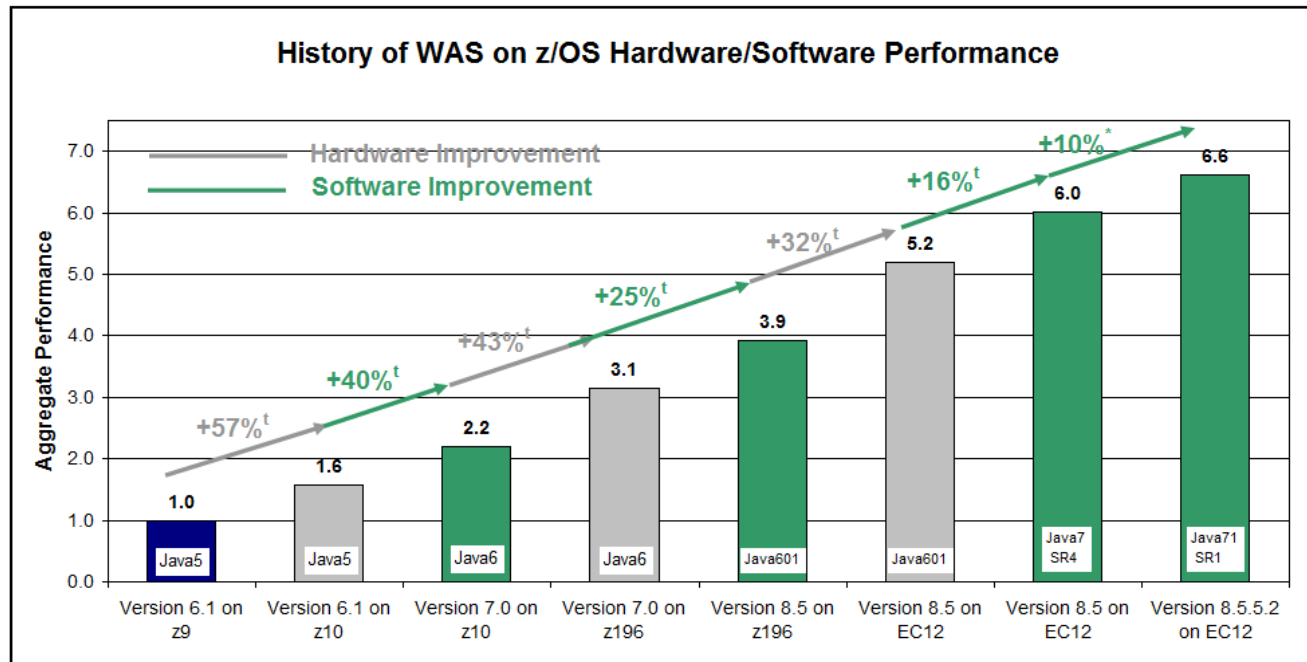
- LP=Large Pages for Java heap CR= Java compressed references
- IBM Java7SR3 using -Xaggressive + 1Meg large pages

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

(Controlled measurement environment, results may vary)

WAS on z/OS – DayTrader

- Aggregate HW, SDK and WAS Improvement: WAS 6.1 (IBM Java 5) on z9 to WAS 8.5 (IBM Java 7R1) on zEC12



* DayTrader3
t DayTrader2

6.6x aggregate hardware and software improvement comparing WAS 6.1 IBM Java5 on z9 to WAS 8.5.5.2 IBM Java7R1 on zEC12

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

(Controlled measurement environment, results may vary)

Java 8!!

Language Updates

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

Java 8.0**

- Language improvements
- Closures for simplified fork/join

IBM Java 8 (J9 R28)

- Improvements in
 - Performance
 - RAS
 - Monitoring
- z13™ Exploitation
 - SIMD
 - SMT
 - Crypto acceleration

SE8
≥ 20 platforms



IBM Java Runtimes

IBM Java 5.0 (J9 R23)

- Improved performance
 - Generational Garbage Collector
 - Shared classes support
 - New J9 Virtual Machine
 - New Testarossa JIT technology
- First Failure Data Capture
- Common runtime technology
 - ME, SE, EE

IBM Java 6.0 (J9 R24)

- 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/Java 7 (J9 R26)

- Improvements in
 - Performance
 - GC Technology
- z196™ Exploitation
 - OOO Pipeline
 - 70+ New Instructions
- JZOS/Security Enhanc

IBM Java 7 (J9 R26 SR3)

- Improvements in
 - Performance
- zEC12™ Exploitation
 - Transactional Exec
 - Flash 1Meg LPs
 - 2G large pages
 - Hints/traps

IBM Java 7R1 (J9 R27)

- Improvements in
 - Performance
 - RAS
 - Monitoring
- zEC12™ Exploitation
 - zEDC for zip acceleration
 - SMC-R integration
 - Transactional Execution
 - Runtime instrumentation
 - Hints/traps
- Data Access Accelerator

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

**Timelines and deliveries are subject to change.

IBM SDK Java Tech. Edition, Version 8 (IBM Java 8)

- **New Java8 Language Features**
 - Lambdas, virtual extension methods
- **IBM z13 exploitation**
 - Vector exploitation and other new instructions
 - Instruction scheduling
- **General throughput improvements**
 - Up-to 17% better application throughput
 - Significant improvements to ORB
- **Improved crypto performance for IBMJCE**
 - Block ciphering, secure hashing and public key
 - Up-to 4x improvement to Public Key using ECC
 - CPACF instructions: AES, 3DES, SHA1, SHA2, etc
- **Significantly improved application ramp-up**
 - Up-to 50% less CPU to ramp-up to steady-state
 - Improved perf of ahead-of-time compiled code
- **Improved Monitoring**
 - JMX beans for precise CPU-time monitoring
- **Enhancements to JZOS Toolkit for Java batch**



Java 8 – Lambdas

- New syntax allows for concise and expressive code snippets
 - Similar to ‘anonymous methods’

```
Collections.sort(people, new Comparator<Person>() {  
    public int compare(Person x, Person y) {  
        return x.getLastName().compareTo(y.getLastName());  
    }  
});
```



```
people.sort(comparing(Person::getLastName));
```

http://www.dzone.com/links/presentation_languagelibraryvm_coevolution_in_jav.html

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

Java 8 – Lambdas for Streaming Operations

- Lambdas can be pipelined to enable data stream operations
 - Intermediate operations on streams produce new streams
 - Terminal operations produce results

```
int totalWeight = widgets.stream()  
                        .filter(w->w.getColor() == RED)  
                        .mapToInt(w->w.getWeight())  
                        .SUM();
```

- Enables exploitation of parallelism and supports multi-core programming

Java 8 – Virtual Extension Methods

- Extend well established data structures while retaining compatibility
- Language enhancement to provide default implementations in interfaces
 - Interface declarations run if classes do not provide an implementation

```
public interface Iterator, E> {  
    public boolean hasNext();  
    public E next();  
    ...  
    public default skip(int i){  
        for(; i > 0 && hasNext(); i--)  
            next();  
    }  
}
```

IBM z13 – Taking Java Performance to the Next Level

Continued aggressive investment in Java on Z

Significant set of new hardware features tailored and co-designed with Java

Simultaneous Multi-Threading (SMT)

- 2x hardware threads/core for improved throughput
- Available on zIIPs and IFLs

Single Instruction Multiple Data (SIMD)

- Vector processing unit
- Accelerates loops and string operations

Cryptographic Function (CPACF)

- Improved performance of crypto co-processors

New Instructions

- Packed Decimal ↔ Decimal Floating Point
- Load Immediate on Condition
- Load Logical and Zero Rightmost Byte

New **5.0 GHz** 8-Core Processor Chip

480Mb L4 cache to optimize for data serving



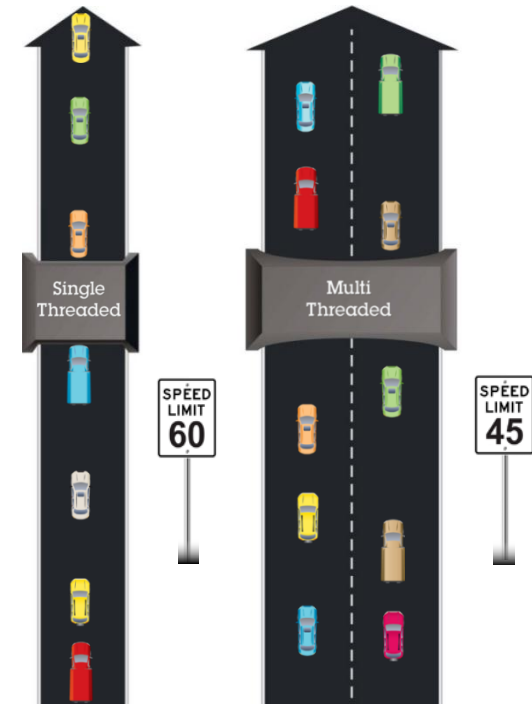
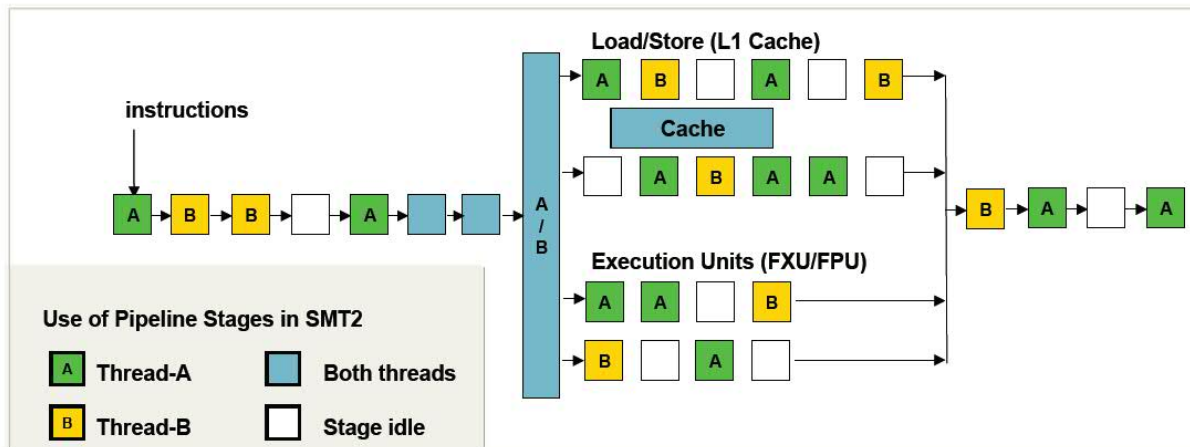
Up to **50%** improvement
in throughput for generic
applications

Up to **2X** improvement in
throughput per core for security
enabled applications

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

IBM z13: SMT – Simultaneous Multi-Threading

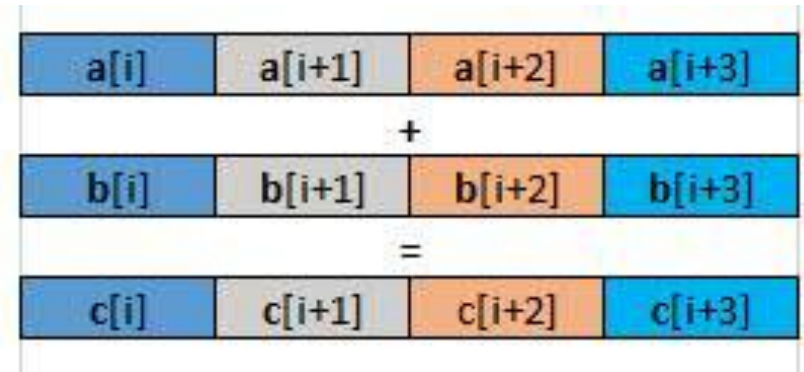
- Double the number of hardware threads per core
 - Independent threads can be more effective utilizing pipeline
- Threads share resources – may impact single thread perf
 - Pipeline (eg. physical registers, fxu, fpu, lsu etc)
 - Cache
- Throughput improvement is workload dependent



Two zIIP lanes handle more traffic overall

IBM z13: SIMD – Single Instruction Multiple Data

- Hardware for exploiting data-parallelism
 - Large uniform data-set that needs the same operation performed on each element
 - Can offer dramatic speedup to data-parallel operations (matrix ops, string processing, etc)



4x less iterations in the SIMD loop

```
//SISD C example, adding two arrays
for (i=0;i<128;i++)
{
  c[i] = a[i] + b[i];
}
```

```
//SIMD C example, adding two arrays
for (i=0;i<32;i++)
{
  vec_add(c[i*4], a[i*4], b[i*4]);
}
```

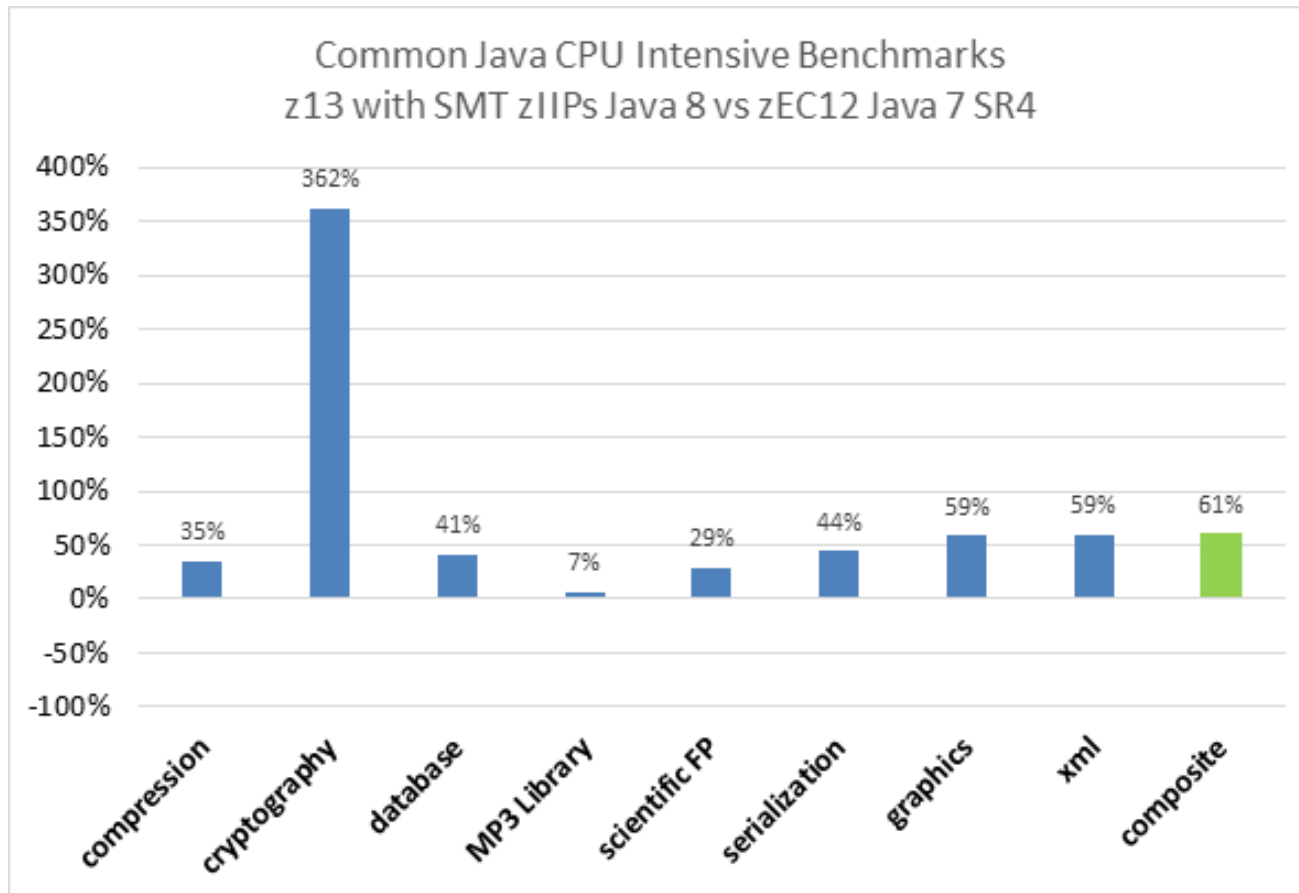
IBM Java 8 - String, Character Conversion and Loop Acceleration with SIMD

IBM z13 running Java 8 on zOS Single Instruction Multiple Data (SIMD) vector engine exploitation

- | | |
|---|---|
| <ul style="list-style-type: none">• java/lang/String<ul style="list-style-type: none">• compareTo• compareToIgnoreCase• contains• contentEquals• equals• indexOf• lastIndexOf• regionMatches• toLowerCase• toUpperCase• getBytes | <ul style="list-style-type: none">• java/util/Arrays<ul style="list-style-type: none">• equals (primitive types)• String encoding converters<ul style="list-style-type: none">• ISO8859-1• ASCII• UTF-8 / UTF-16• Auto-SIMD<ul style="list-style-type: none">• Simple loops• (e.g. Matrix Multiplication) |
|---|---|

Primitive operations are between 1.6x and 60x faster with SIMD

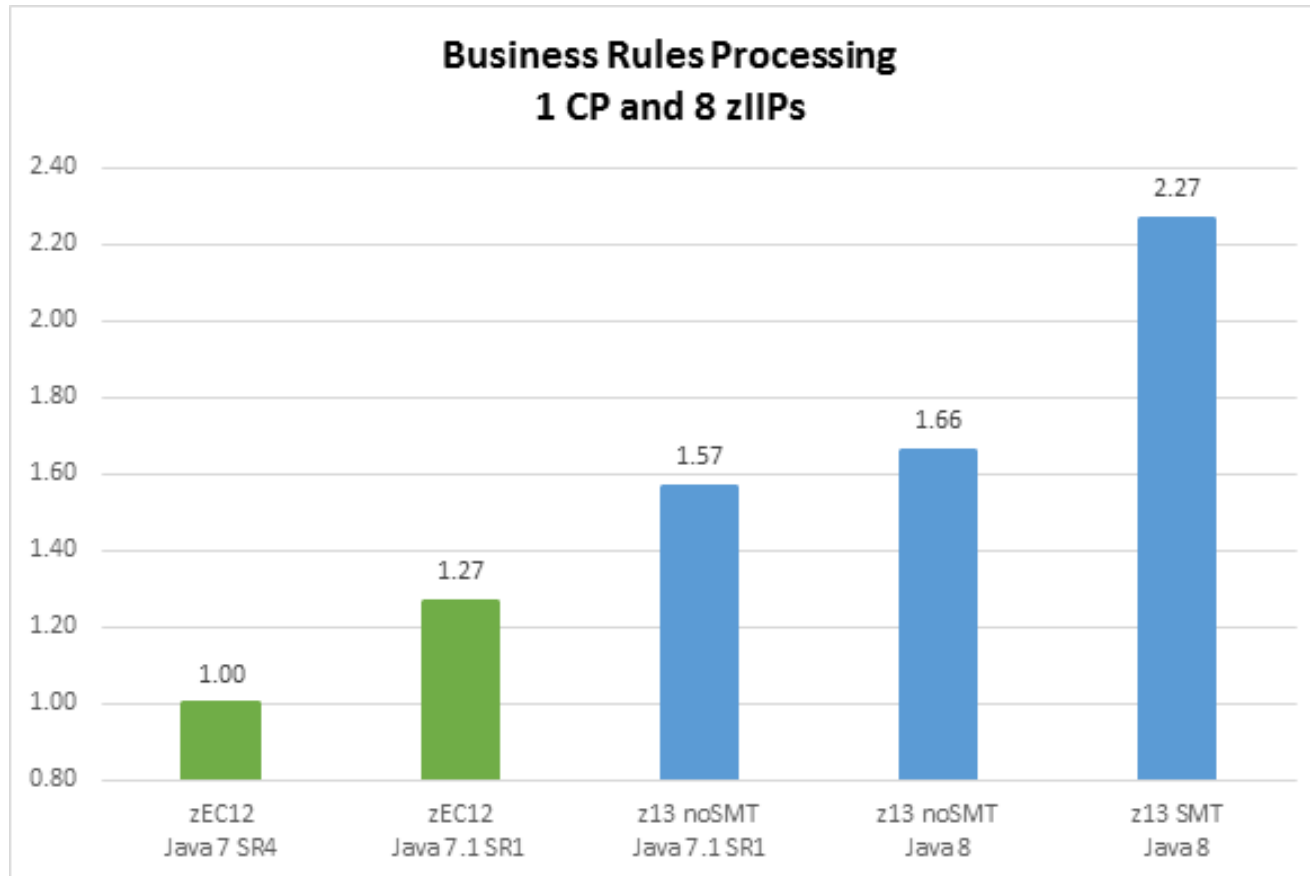
IBM Java 8: CPU-Intensive Benchmark



IBM z13 and IBM Java 8 show a composite improvement of 61% over zEC12 and Java7 SR4 running CPU Intensive benchmark

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

IBM Business Rules Processing with IBM Java 8 and IBM z13

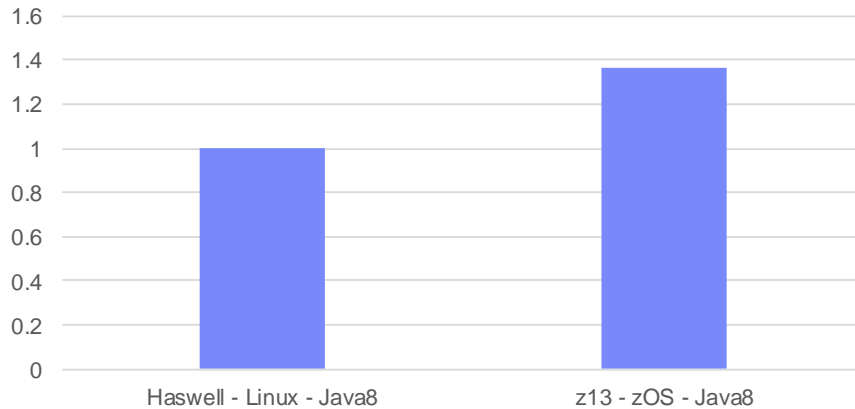


Aggregate 2.27x improvement from IBM Java 8 and IBM z13

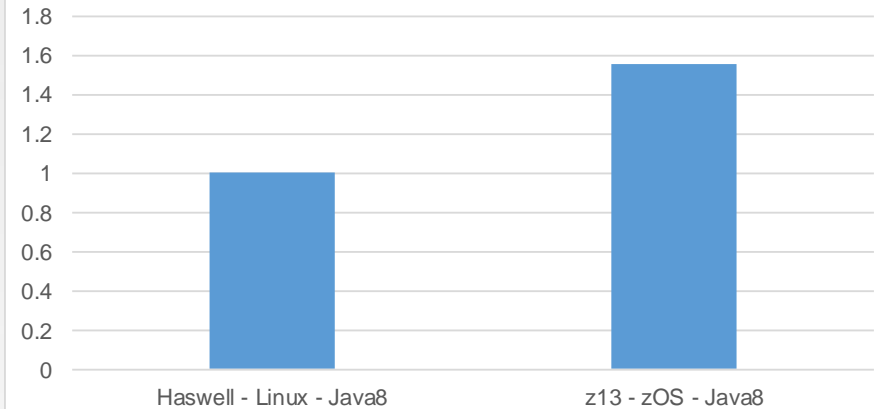
Complete your session evaluations online at www.SHARE.org/Seattle-Eval

Business Rules Processing – IBM z13 vs Intel Haswell

**Business Rules Processing with IBM z13
with Java8 - Small Ruleset**

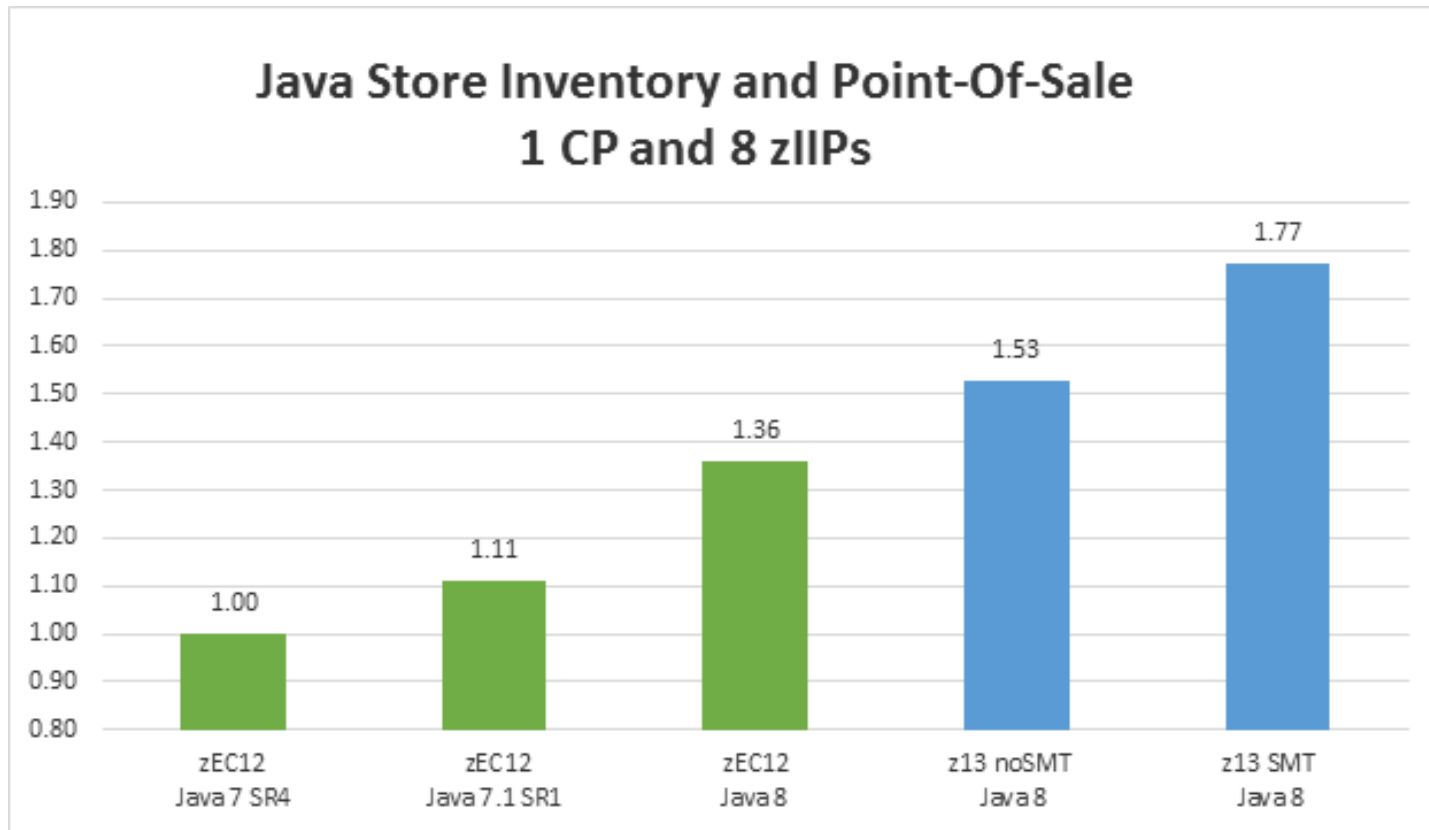


**Business Rules Processing with IBM z13
with Java8 - Large Ruleset**



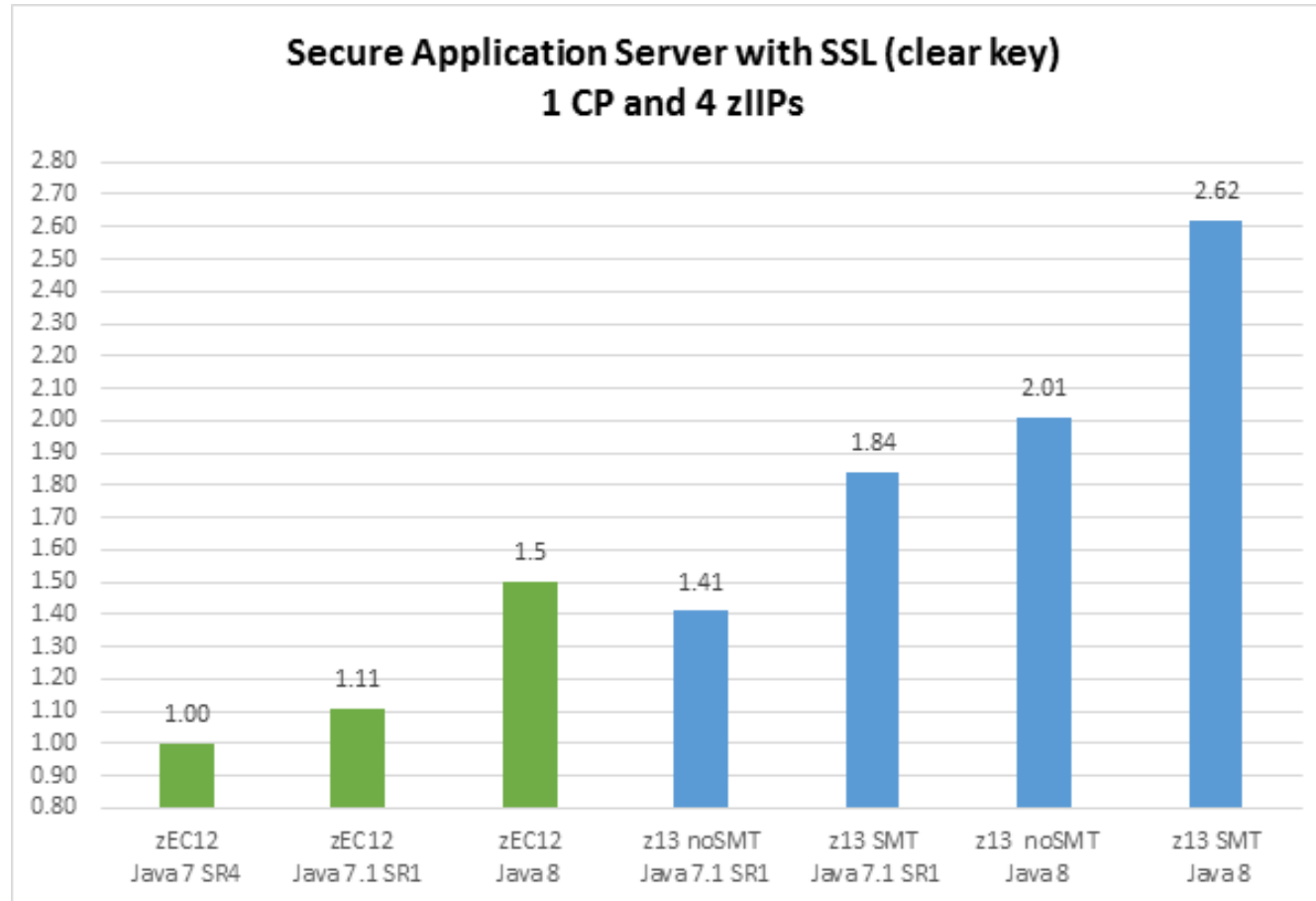
IBM z13 up-to 1.5x better throughput/core processing business rules than Intel Haswell

Java Store, Inventory and Point-of-Sale App with IBM Java 8 and IBM z13



1.77x improvement in throughput with IBM Java 8 and IBM z13

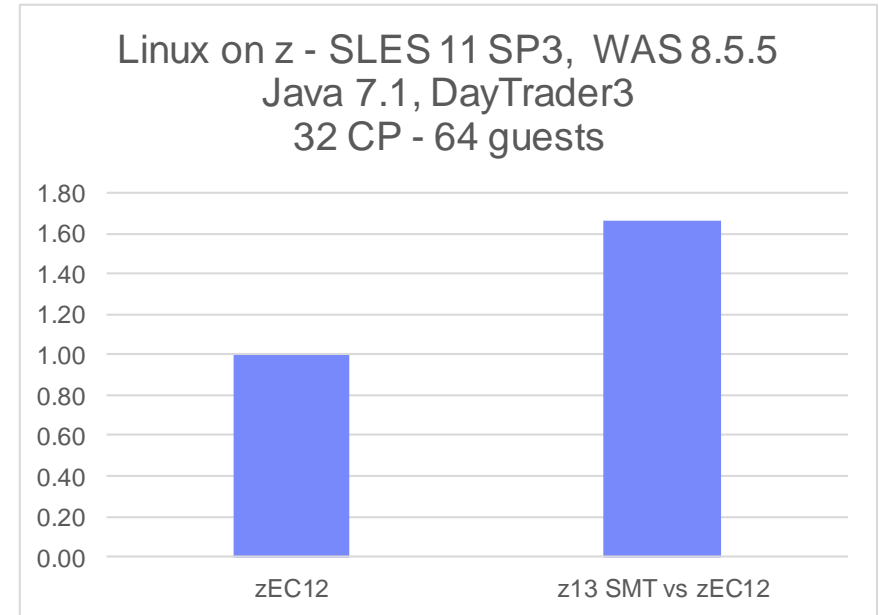
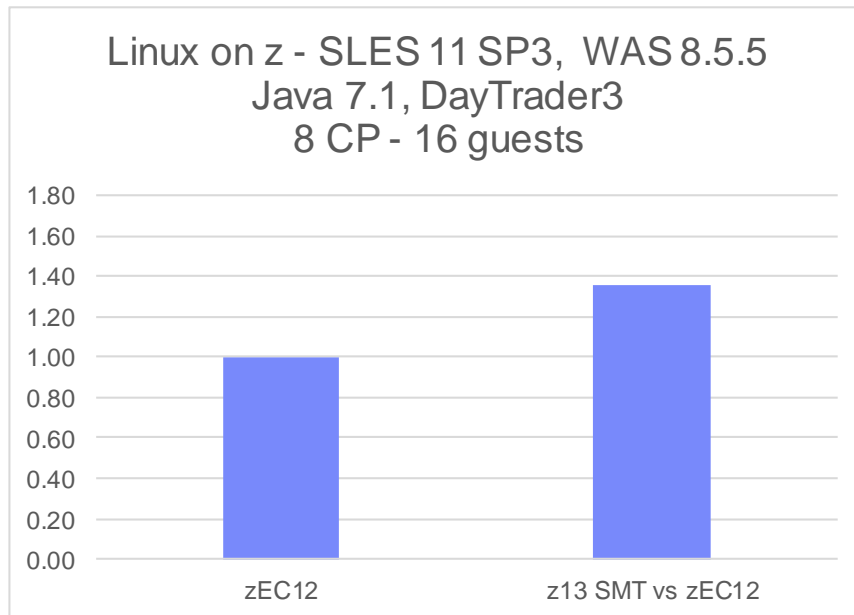
Application Serving – SSL-Enabled DayTrader3.0



2.6x improvement in throughput with IBM Java 8 and IBM z13

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

WebSphere – Linux on z Virtualized Cluster

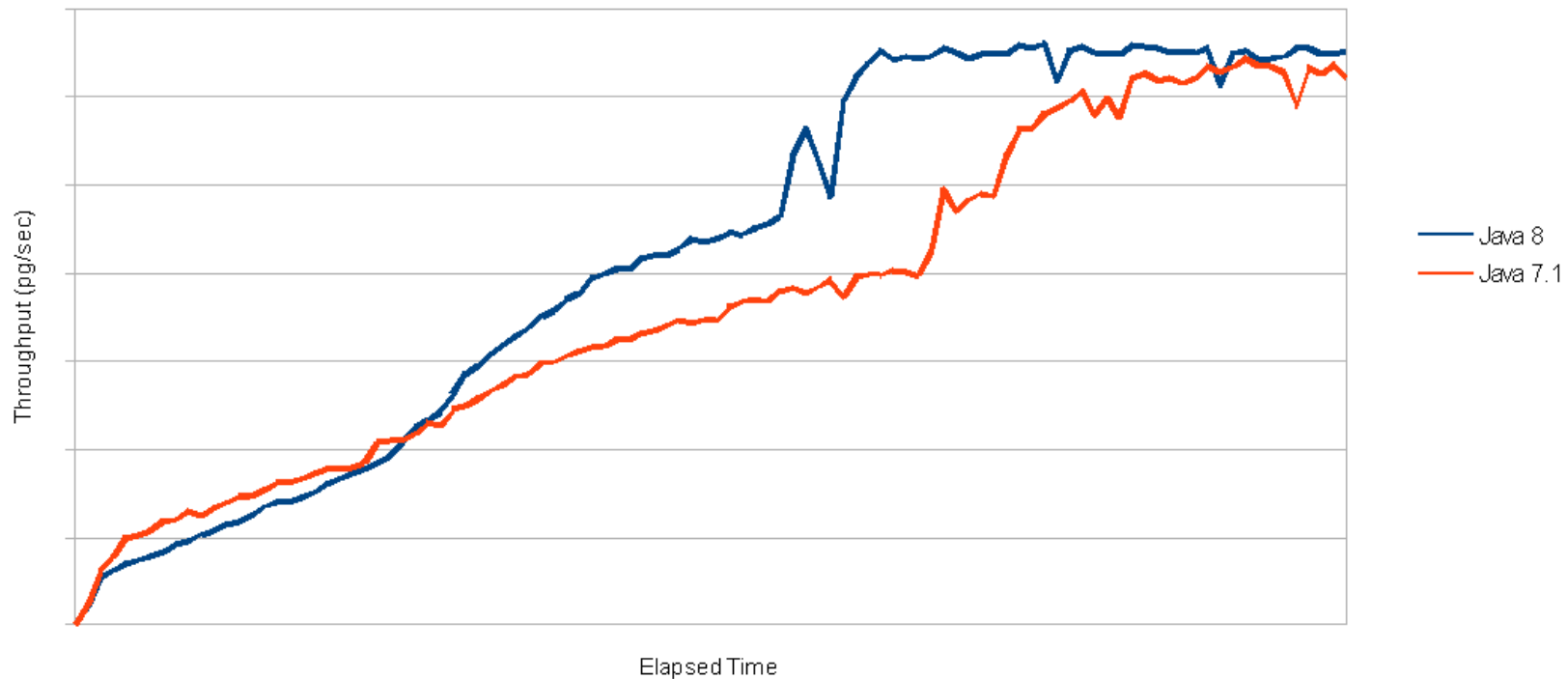


Between 1.36x to 1.66x improved throughput for a virtualized WAS cluster running DayTrader 3.0 on IBM z13 when compared to zEC12

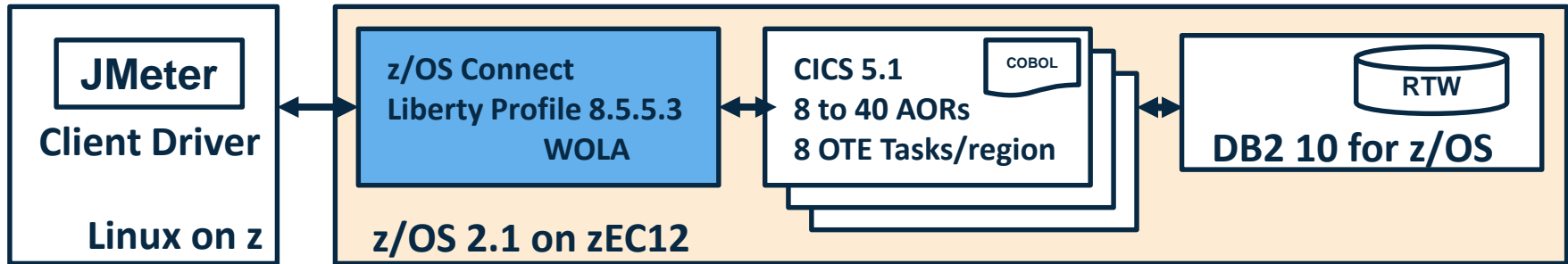
DayTrader Ramp-up

DayTrader 3 Rampup

zOS 64-bit, 4 zEC12 cores, Liberty 8.5.5.5

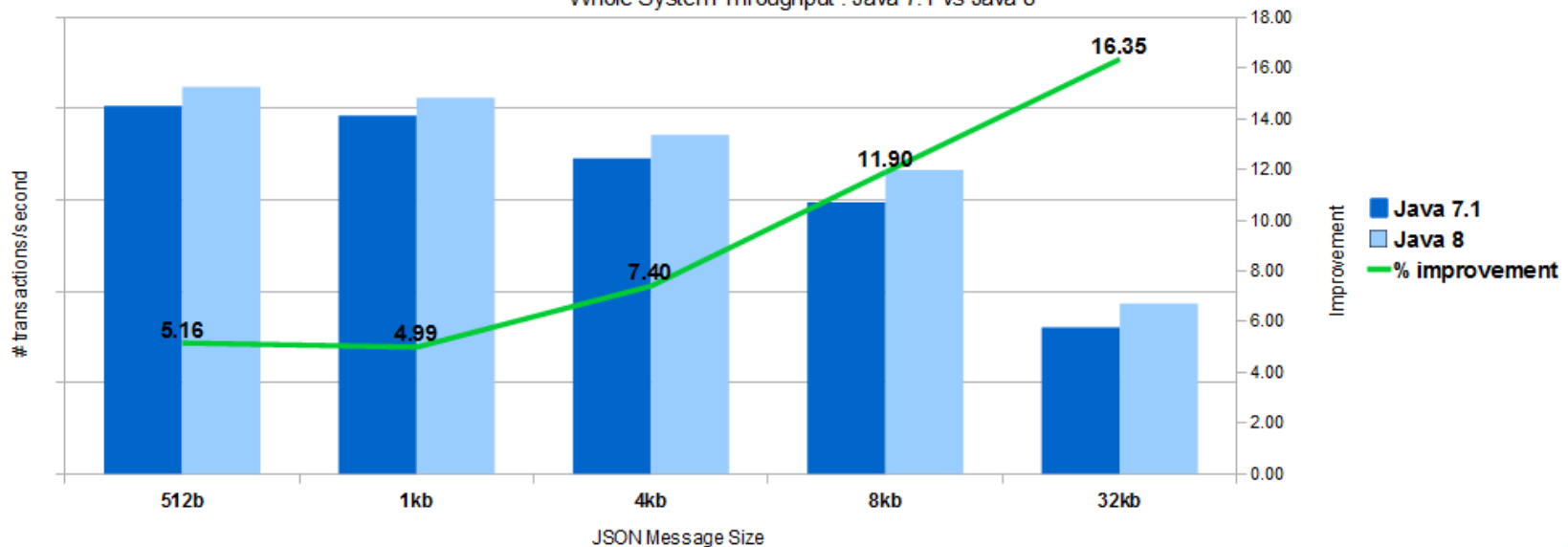


Mobile on z – z/OS Connect on IBM Java 8 and zEC12



z/OS Connect on zEC12

Whole System Throughput : Java 7.1 vs Java 8



5-16% throughput improvement from IBM Java 8 and IBM zEC12

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

(Controlled measurement environment, results may vary)

Java8 - JMX Beans for Precise CPU Monitoring

New JMX Beans for reporting CPU usage categorized by:

1. JVM System threads (JIT, GC, etc)
2. Application threads
3. Monitoring threads (to be able to excl. monitoring overhead)

Intended use-cases

- Reporting transaction cpu usage
- Identifying "expensive" transactions
- Reporting JVM overhead over specific intervals
- Foundation for future work on tracking idle behaviour

New classes

- `com.ibm.lang.management.JVMCpuMonitorMXBean` (Bean to request Data)
 - `getThreadsCpuUsage()`
 - `setThreadCategory() / getThreadCategory()`
- `com.ibm.lang.management.JVMCpuMonitorInfo` (Object with Data)

Overhead may be visible on some platforms

Option to trade-off more precise GC-time reporting vs. reduced overhead

`-XX:+ReduceCPUMonitorOverhead(default.) / -XX:-ReduceCPUMonitorOverhead`

(z/OS cannot enable more precise GC-time reporting today)

Other cool sessions!

- **IBM Java: JVM Tuning**
 - Tuesday @ 1:45 PM Metropolitan A
- **WebSphere Liberty Profile and Traditional WebSphere Application Server - What's New?**
 - Tuesday @ 3:15 PM Leschi
- **Debug 101-Using ISA Tools for Apps in WebSphere Application Server z/OS**
 - Wednesday @ 3:15 PM Virginia
- **Enterprise JavaScript with the IBM SDK for Node.js**
 - Wednesday @ 8:30 AM Greenwood
- **JavaScript with the IBM SDK for Node.js Hands-on Lab**
 - Wednesday @ 12:30 PM Redwood

Thank You!



- Please complete your session evaluations!

Session 16815:
IBM Java News

- www.share.org/Seattle-Eval

Joran Siu

Email: joransiu@ca.ibm.com



Important references

- IBM Java for Linux website
 - <http://www.ibm.com/developerworks/java/jdk/linux>
- z/OS Java website
 - <http://www.ibm.com/systems/z/os/zos/tools/java>
- IBM SDK Java Technology Edition Documentation
 - <http://www.ibm.com/developerworks/java/jdk/docs.html>
- 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>