



# IBM zEnterprise Technology Summit

Next-level solutions with IMS and Java  
on System z



## Session Objectives and Agenda

- **Java z/OS platform strategy**
- **Java and IMS strategy and direction**
- **Futures**



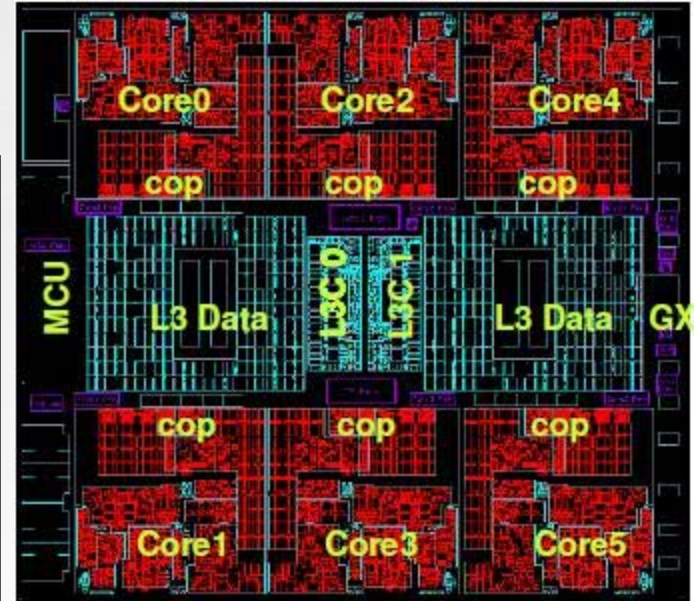
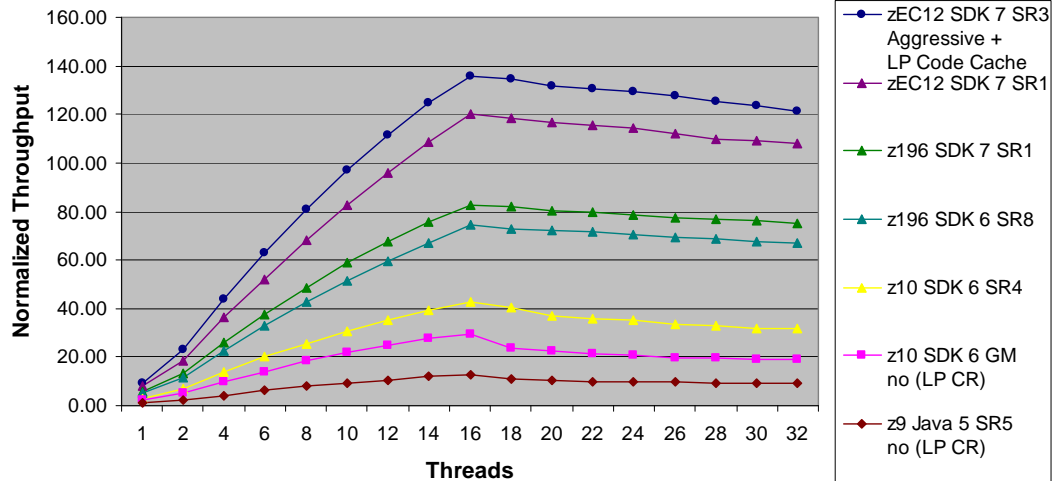
# zEC12 and Java

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

**Large caches** to optimize data serving

Second generation **OOO design**

**z/OS Multi-Threaded 64 bit Java Workload 16-Way**  
~12x Improvement in Hardware and Software



**Up-to 45% improvement in throughput amongst Java workloads measured with zEC12**

**Multi-threaded workload shows ~11x aggregate hardware and software improvement comparing Java5SR5 on z9 to Java7SR1 on zEC12**

# 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

## *Run-time Instrumentation (RI)*

Real-time feedback on program characteristics

Enables increased optimization by JRE

## *2GB page frames*

Improved performance targeting 64-bit heaps

## *Page-able 1MB large pages using flash*

Better versatility of managing memory

## *New software hints/directives*

Data usage intent improves cache management

Branch pre-load improves branch prediction

## *New trap instructions*

Reduce over-head of implicit bounds/null checks



IBM plans for future maintenance roll-ups of IBM 31-bit and 64-bit SDK7 for z/OS Java(TM) Technology Edition, Version 7 (5655-W43 and 5655-W44) (IBM SDK7 for z/OS Java), to provide exploitation of new IBM zEnterprise EC12 features, including: Flash Express and pageable large pages, Transactional Execution Facility, Miscellaneous-Instruction-Extension Facility, and 2 GB pages. In addition, IBM SDK7 for z/OS Java is available for use by IBM middleware products running Java, such as IBM IMS 12 (5635-A03), IBM DB2 10 for z/OS (5605-DB2), and the Liberty profile of IBM WebSphere Application Server for z/OS v8.5 (5655-W65); and is planned for use by a future release of CICS

# Java z/OS

## z196 and Java6.0.1: 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

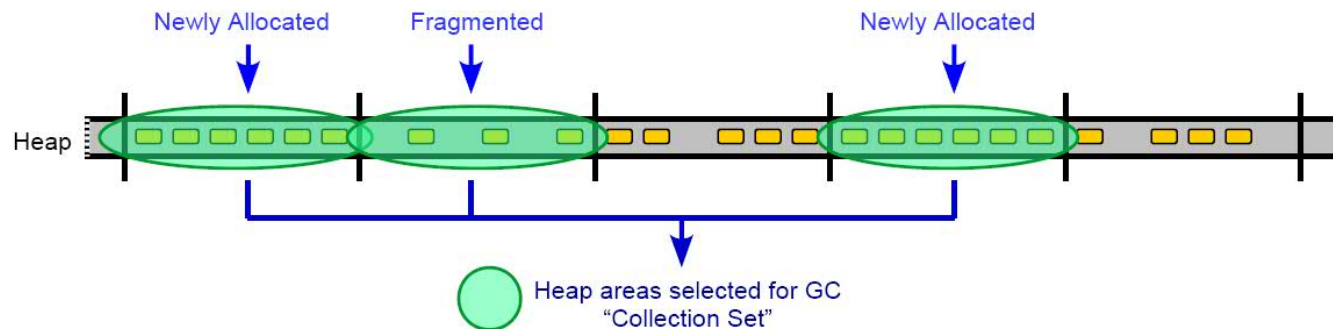
## Performance

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



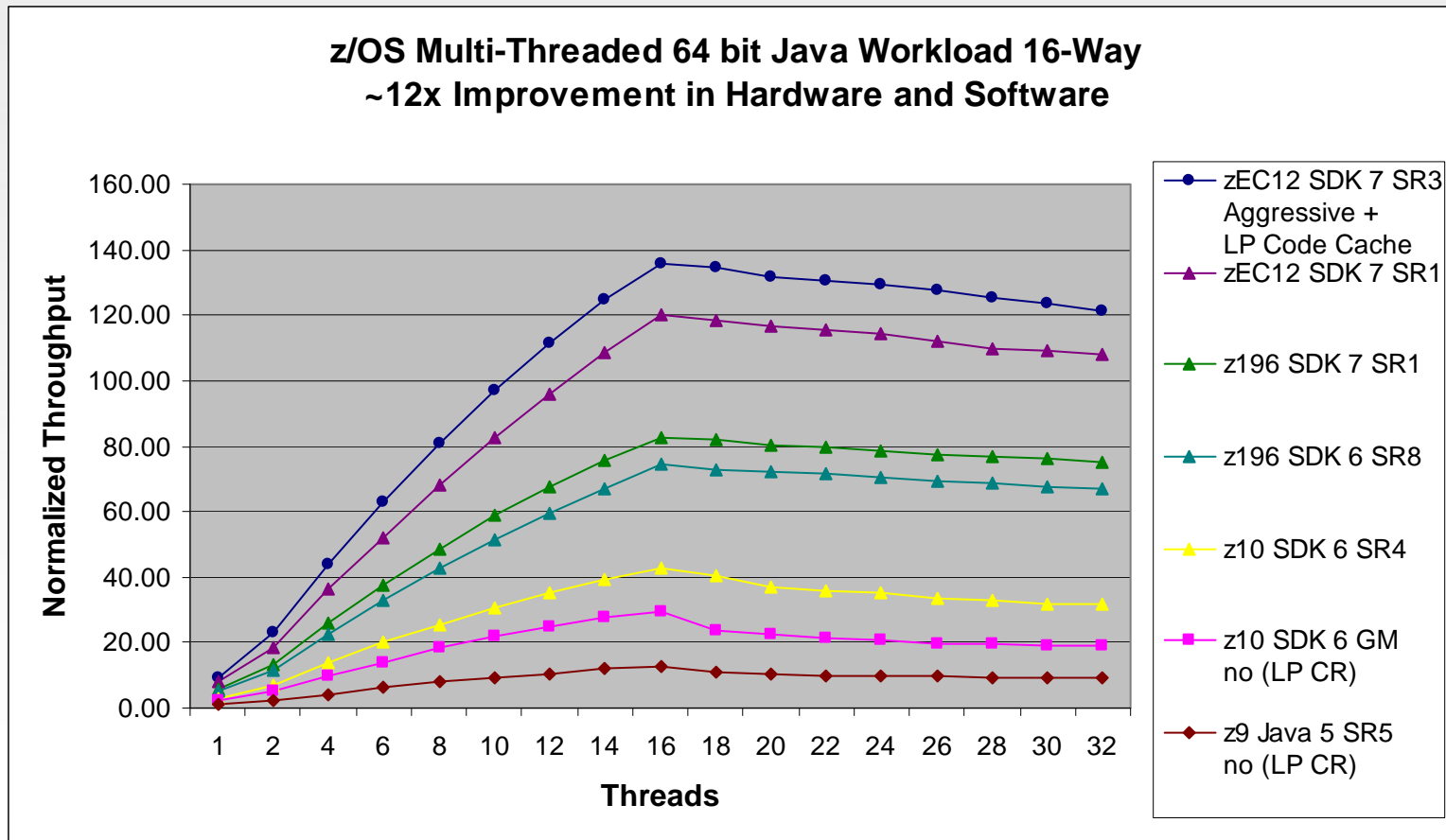
## IBM J9 2.6 Technology Enhancements - Garbage Collection: Balanced Policy

- **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**



# z/OS Java SDK 7: 16-Way Performance

Aggregate HW and SDK Improvement z9 Java 5 SR5 to zEC12 Java 7

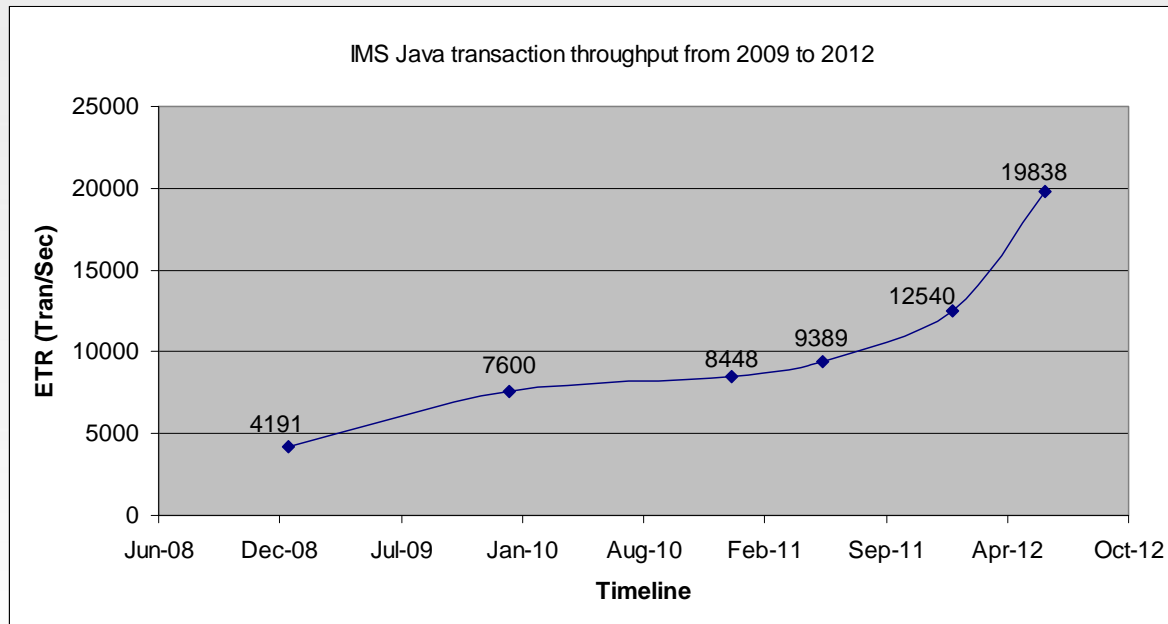


~12x aggregate hardware and software improvement comparing Java5SR5 on z9 to Java7SR3 on zEC12

LP=Large Pages for Java heap CR= Java compressed references  
(Controlled measurement environment, results may vary)

## IMS JMP region performance

### Aggregate SDK, software and hardware improvements



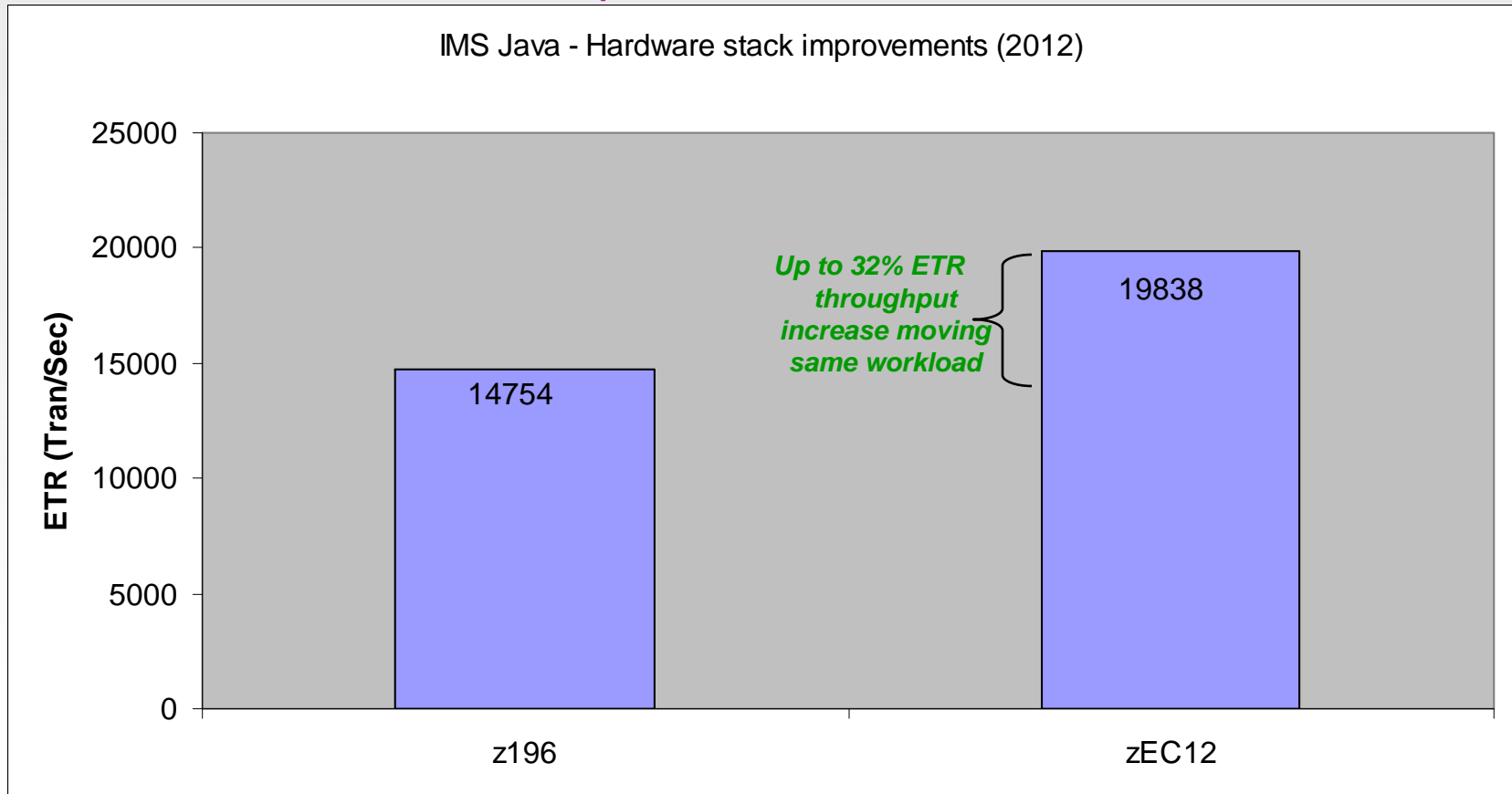
**Over 4x aggregate throughput improvement from 2009 to 2012 due to the following enhancements**

- Java version to version performance improvements
- IMS improvements
- Hardware improvements
- DASD improvements



# IMS JMP region performance

## Hardware stack improvements



(Controlled measurement environment, results may vary)



# Java and IMS

## Java is an integral component of the IMS modernization strategy

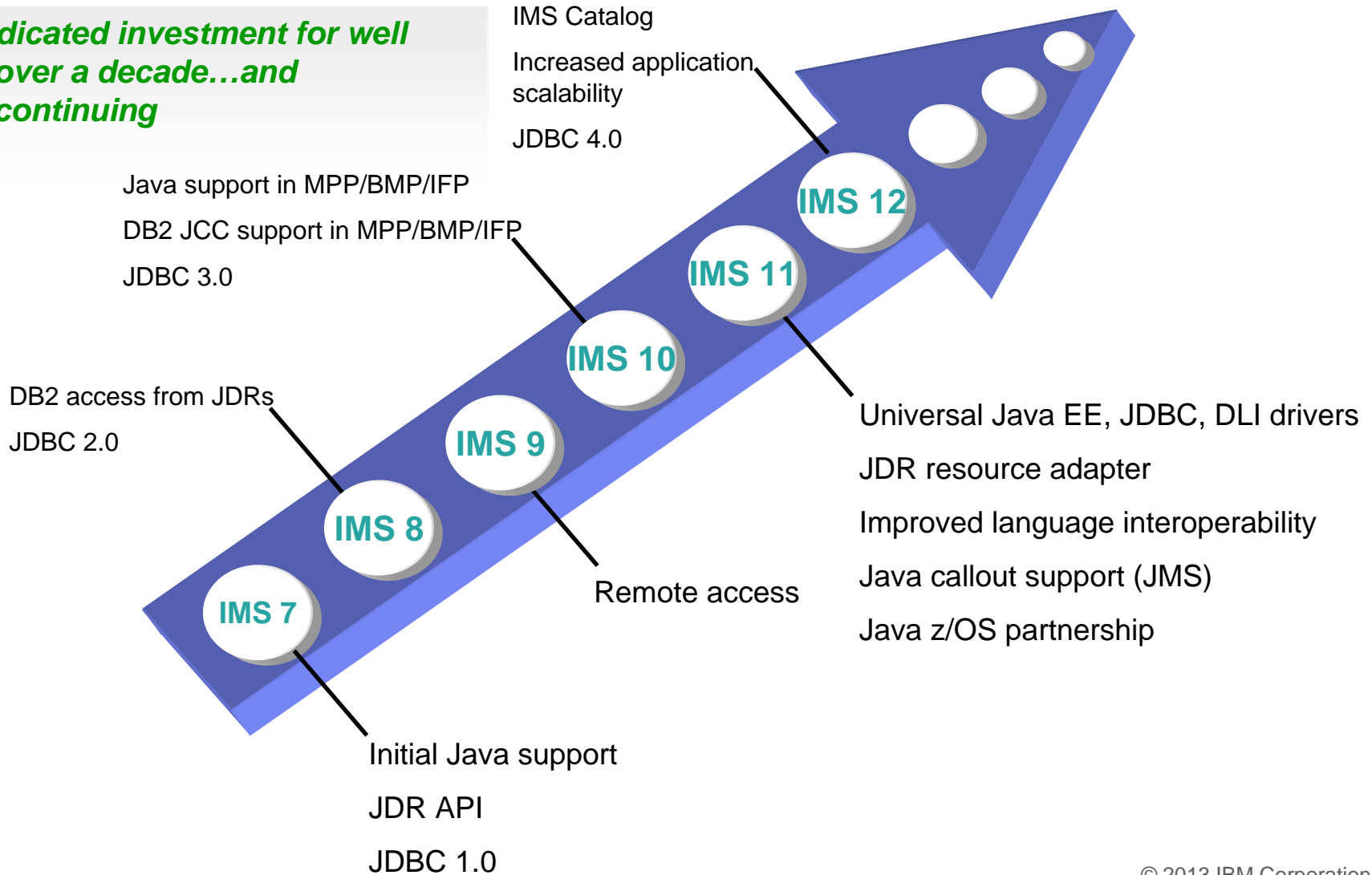
- Enable customers to quickly achieve IMS value while significantly reducing development costs and improving productivity
- IMS leverages the IBM JVM for System z and integrates it into the IMS runtime containers

## IMS family has a long-term commitment to Java

- Investing over 50 FTEs (full-time equivalents) in Java technology moving forward
  - IMS dependent region types (JMP, JBP, MPP, BMP, IFP)
  - Java EE platform (WebSphere Application Server)
  - z/OS and open systems access to IMS assets

# Java and IMS – IMS 7 to IMS 12 highlights

***Dedicated investment for well over a decade...and continuing***



# Java dependent region deployment

## Java dependent region resource adapter

- **Allows new IMS transactions (JMP, JBP) to be written in Java and managed by the IMS transaction manager**
- **Complete Java framework for applications operating in an IMS container**
  - Message queue processing
  - Program switching
    - Deferred and immediate
  - Transaction demarcation
  - GSAM support
  - Additional IMS call support necessary for IMS transactions
    - INQY
    - INIT
    - LOG
    - Etc
- **Shipped with type 2 Universal drivers**



# IMS Open Database

## Solution statement

- **Extend the reach of IMS data**
  - Offer scalable, distributed, and high-speed local access to IMS database resources

## Value

- **Business growth**
  - Allow more flexibility in accessing IMS data to meet growth challenges
- **Market positioning**
  - Allow IMS databases to be processed as a standards-based data server

## Key differentiators

- **Standards-based approach (Java Connector Architecture, JDBC, SQL, DRDA)**
- **Solution packaged with IMS**

## Enables new application design frameworks and patterns

- **JCA 1.5 (Java EE)**
- **JDBC**

# Java and IMS moving forward

## Java z/OS stakeholder

- Continued partnership to maximize synergy between IMS and Java z/OS

## Performance

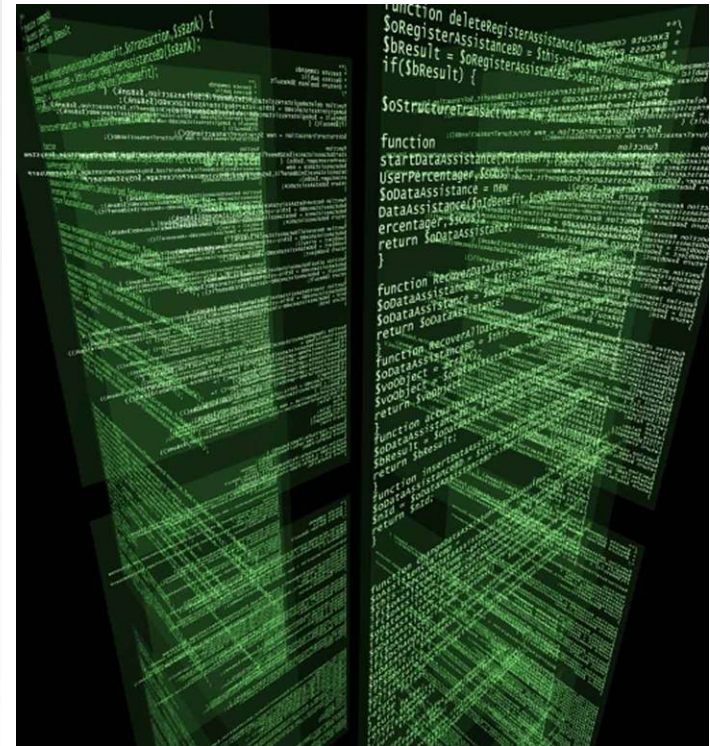
- Aggressive performance analysis and cooperative approach to continue h/w and s/w exploitation

## Enterprise modernization

- Language interoperability
- Universal drivers/JDR resource adapter

## Integration

- Aggressive approach to horizontal integration across IBM portfolio
  - Rational
  - Cognos
  - Data Studio
  - InfoSphere



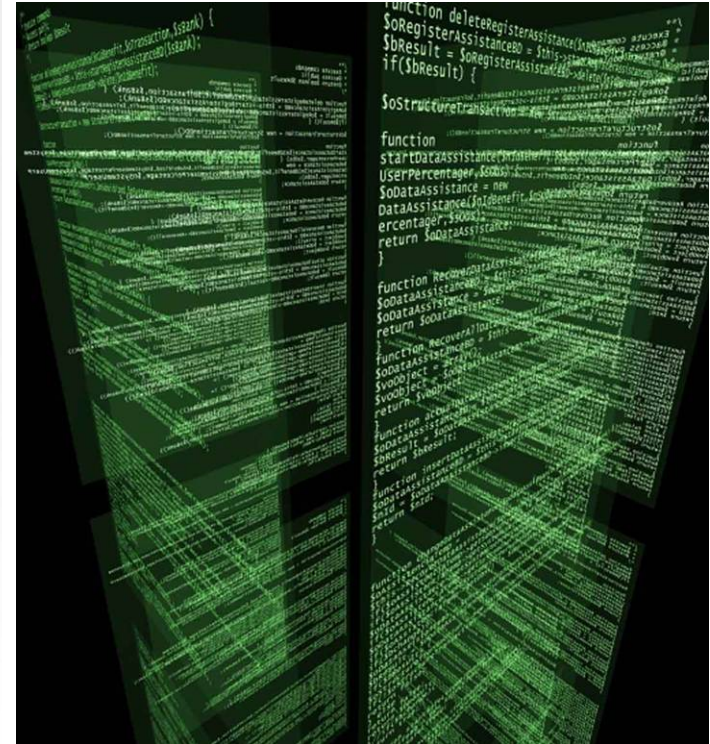
# Language interoperability (Java and COBOL)

- **Significant collaboration over the past year to enrich language interoperability in IMS dependent regions**
  - IMS, Language Environment (LE), Java z/OS, COBOL organizations have all worked together
    - Including a major European customer (Fiducia)
- **Specific areas of focus in order to ensure a robust offering**
  - Exception handling and percolation
  - Real-time debugging (stepping through the stack across language boundaries)
  - Cleaning out (optionally) COBOL working storage areas across application schedules
  - Performance
  - Several others
- **Continued collaboration**
  - Want to start a working group with direct focus in this space between IBM and interested parties
    - Direct IBM assistance to propel your organization forward
    - Interested? Let me know.
- **What about Java and PL/I?**
  - Would like to start investing in this space
  - Interested? Let me know.

# Java and IMS moving forward

## Continued modernization of the core system

- IMS catalog
- Database versioning
- Dynamic database
- Native SQL
- Programming models







## Who

- Caterpillar
  - Core manufacturing system managed by IMS

## Business Challenge

- Required open systems access to IMS database assets
- Error-prone process to accomplish task
  - Unloaded databases and did manual entry into open system database

## Solution

- Leverage IMS Open Database technology and the Universal JDBC driver

## Benefits

- Real-time access to data
- Confident decision making
- Trusted information



## Who

- Northwest Airlines/Delta
  - Largest airline in the world
  - Technical operations managed by IMS

## Solution

- Implement IMS/JDBC on z/OS to integrate technical operations data via ESB and WebSphere Application Server

## Business Challenge

- Integrate critical applications after merger with Delta
- Implement a distributed application front-end using SOA on top of existing z/OS

## Benefits

- Technical infrastructure is much more open and primed for integration across the enterprise
- Smooth integration of all critical applications running on z/OS after merger with Delta



## Who

- Worldwide bank
  - Core banking system managed by IMS TM/DB and written mostly in COBOL

## Solution

- Leverage the JDR resource adapter and Universal JDBC and Universal DLI drivers for IMS
- Integration of existing assembler modules common to the application framework
- Deployment in JMP regions
- Initially no language interoperability (pure Java)
  - Future direction

## Business Challenge

- Modernize existing core services
  - Offer new services framework to business partners
- Impaired ability to deliver new function

## Benefits

- Leverage abundant Java domain knowledge in industry
- Dramatically increased time to market
- IMS API consistency with relational databases



## Who

- Bank in US
  - Several banking channels managed by IMS and written mostly in COBOL

## Solution

- Introduce a new banking channel implemented in Java using the Universal JDBC and Universal DLI drivers for IMS
- Deployment in CICS JCICS regions
- Initially no language interoperability (pure Java)
  - Future potential

## Business Challenge

- Introduce additional core services to support new banking channels
- Impaired ability to deliver new function

## Benefits

- Leverage abundant Java domain knowledge in industry
- Dramatically increased time to market
- IMS API consistency with relational databases

## Who

- German bank
  - Framework mainly PL/I with conversational transactions

## Business Challenge

- Integration of 3<sup>rd</sup> party credit checking technology that was part of a Java package

## Solution

- Leverage the deferred program switching support in Java class libraries to switch conversation iterations from MPP to JMP regions and back

## Benefits

- Ability to leverage decades of existing assets and add in new Java-based services into the architecture transparently
- Just another service
- In production within a month with this solution



# Summary

## IMS is committed to enterprise modernization

- Deep synergy across many organizations within IBM
- Portfolio integration is very important
- Constantly validating the enterprise roadmap with customers

## The partnership of IMS and Java technology is capable of handling mission-critical workload

- IMS is an important stakeholder in the IBM Java on System z strategy
- Java running in IMS regions has been benchmarked at over 12,500 transactions per second

## Many customers are modernizing their IMS application development patterns and access paradigms around Java as the primary language of choice

- Over 40 proof of concepts in the last year alone