Java Overview and zSeries Update

Theresa Tai
March 07, 2006 Session 8365
IBM System z New Technology Center
Poughkeepsie, New York
ttai@us.ibm.com

Trademarks

IBM, AIX, CICS, DB2, IMS, z/OS, OS/390, S/390, System/390, VisualAge, WebSphere Application Server, WebSphere Studio, z/VM - are trademarks or registered trademarks of the IBM Corporation

Sun, Sun Microsystems, JavaSoft, Java, JavaBeans, JDK, Java 2 Micro Edition, J2ME, Java 2 Standard Edition, J2SE, Java 2 Enterprise Edition, J2EE - are trademarks or registered trademarks of Sun Microsystems Inc.

Content

IBM JVM on zSeries Update

Overview: IBM Java Virtual Machine

IBM Investment in Java

- **Market leader in delivering Java technology**
- **z/OS platform for mission critical workload**

2006 focus for Java on z/OS

- **IBM SDK for z/OS, Java 2 Technology Edition V5**
- **Supporting key IBM middleware & Operating Systems**
- **IBM zAAP Engines for Java Workloads**

Looking Ahead: What's New and Exciting

Wrap-up and final Q&As

SDK for z/OS and OS/390

- IBM Developer Kit for OS/390, Java 2 Technology Edition V1.3
  - Currently at SDK1.3.1 Level
  - Product 5655-D35
  - Build Level: April 21, 2005 (PTF UK03478) at SR 25
- IBM SDK for z/OS, Java 2 Technology Edition, Version 1.4
  - Currently at SDK1.4.2 level
  - Product 5655-I56; Supported on z/OS V1.2 and above
  - Build Level: January 26, 2006 (PTF UK11298)
- IBM 64-bit SDK for z/OS, Java 2 Technology Edition, Version 1.4
  - Web available on September 24, 2004 at SDK1.4.2 level
  - Product 5655-M30; Supported on z/OS V1.6 and above
  - Build Level: January 20, 2006 (PTF UK11177)
- IBM 31-bit and 64-bit for z/OS, Java 2 Technology Edition, Version 5
  - Original product delivery
- No Charge Product but it is supported by the normal IBM support channels

NOTE: The EOS date for SDK 1.3.1 is September 2007
The EOS date for SDK 1.4.2 is September 2009

SDK for Linux on zSeries

- IBM Developer Kit for Linux®, Java 2 Technology Edition, Version 1.3
  - SDK1.3.1 Level
  - SuSe SLES 8, Turbo
- IBM 31-bit SDK for Linux® on zSeries, Java 2 Technology Edition, Version 1.4
  - SDK1.4.2 Level
  - SuSe SLES 8, 31 bit mode, RHEL3, RHEL4
- IBM 64-bit SDK for Linux® on zSeries, Java 2 Technology Edition, Version 1.4
  - SDK 1.4.2 Level
  - SuSe United LINUX (aka SLES 8 & SLES 9), RHEL 3 & 4 (NPTL)
- IBM 31-bit and 64-bit SDK for Linux on zSeries, Java 2 Technology Edition, Version 5
  - SLES 9
  - RHEL 3 and RHEL 4
- Delivery and Service
  - On DeveloperWorks at
    - **https://www6.software.ibm.com/dl/lxdk/lxdk-p**
  - Also available from LINUX distributors
  - Level 1, 2 service by IGS contract
  - Same EOS as SDK for z/OS: 9/07 for SDK 1.3.1 and 9/09 for SDK 1.4

See a summary of tested platforms at http://www-128.ibm.com/developerworks/java/jdk/linux/tested.html

IBM SDK V5

- IBM "Value Add" changes
  - Large number of Java Runtime Changes
    - New GC implementation
    - New Just In Time (JIT) Compiler
    - Shared Classes on all Platforms
    - New RAS Functionalities and Tools
  - Number of RAS and Debugging Changes
  - New functionality
- IBM 31-bit and 64-bit SDK for z/OS, Java 2 Technology Edition, Version 5 (Available 12/05)
  - Provides a full-function SDK at Java 2 technology level with Sun SDK 5 APIs
  - Available from the IBM eServer zSeries Java web site and on tape from IBM Software Delivery and Fulfillment (SDF)
- System requirements
  - z/OS V1.6 or z/OS.e V1.6 or later is required
  - z800, z890, z900, z990 and z9-109
- Compatibility
  - 31bit SDK V5 is compatible with SDK V1.4.x with the exception of Persistent Reusable JVM

Note: SDK V5 will be serviced through September 2009

SDK V5 GC New Implementation

- Uses a “Type Accurate” Collector
  - Does not suffer from “retained garbage”
  - Does not suffer from pinned/dosed objects
- Introduces a new Generational GC Policy
  - -Xgcpolicy:gencon
    - “gencon” – controls the behavior of the Garbage Collector
    - To minimized the amount of pause time in GC cycle
  - Two generational collector
    - The new and old areas

SDK V5 JIT New Implementation

- Uses a separate compilation thread
  - Methods queued for compiling
- Can recompiled methods
  - 5 optimization levels
  - Profiling thread helps to determine “hotness”
- Still carries out compilation under –Xdebug
  - Termed “full speed debug”

SDK V5 Shared Classes

- Available on all platforms
- Static class data caches in shared memory
  - Shared between all IBM Java VMs
  - All application and bootstrap classes shared
  - Cache persisted beyond lifetime of any JVM, but lost on shutdown/reboot
- Provides saving to memory footprint and start up time

SDK V5 RAS Enhancement

- Reworked Trace Engine
  - Activated using –Xtrace
- More flexible Dump Initiation
  - Initiated using –Xdump
  - More trigger events
- DTFJ Dump Analysis API
  - **D**iagnostic **T**oolkit **F**ramework for **J**ava
  - Packages
    - Com.ibm.dtfj.image
    - Com.ibm.dtfj.java
- Debugger Extensions

Changes from SDK 1.4.2 to V5

- JAXP 1.2 (SDK 1.4.2)
  - SAX 2.0.1
  - DOM Level 2 Core
  - Validation coupled to parsing
  - Transformer can serialize XML 1.0 documents
  - Default ErrorListener throws exceptions on errors & fatal errors
  - Experimental support for XPath evaluation provided by DOM Level 3 XPath
  - XML Schema 1.0 1st edition
  - Experimental XInclude 1.0 support
- JAXP 1.3 (SDK 5.0)
  - SAX 2.0.2
  - DOM Level 3 Core & Load/Save
  - Validation decoupled from parsing
  - Transformer can serialize XML 1.0 and XML 1.1 documents
  - Default ErrorListener reports all warnings and errors to System.err
  - Full support for XPath evaluation provided by JAXP 1.3 (javax.xml.xpath)
  - XML Schema 1.0 2nd edition
  - Full XInclude 1.0 support

Migration Considerations for SDK V5

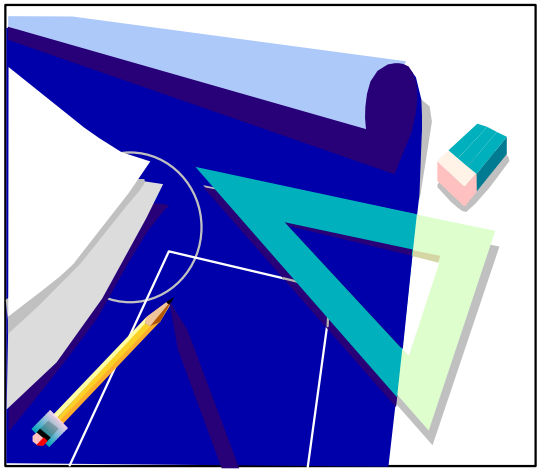
- Incompatibilities
  - http://java.sun.com/j2se/1.5.0/compatibility.html
  - Deprecated APIs
  - http://java.sun.com/j2se/1.5.0/docs/api/deprecated-list.html
- Prerequisites
  - http://www-03.ibm.com/servers/eserver/zseries/software/java/j5prereq31.html
- Restrictions and Other Considerations
  - http://www-03.ibm.com/servers/eserver/zseries/software/java/j5restrict31.html
- SDK V5 APIs
  - http://java.sun.com/j2se/1.5.0/docs/apip

Looking Ahead: What's New and Exciting

- Announcing JZOS Java Batch Toolkit
  - For z/OS SDKs only
  - Runs on z/OS V1.6
  - Currently available on alphaWorks (jzos.zip)
  - http://www.alphaworks.ibm.com
  - Not supported through the official support channel
    - JZOS Forum is available
- To be Integrated into IBM SDK
  - SDK 1.4.2 and SDK V5
  - Target availability 3Q-4Q 2006
  - Supported by IBM support channels
- To learn more about JZOS Batch Toolkit Session 8368 and 8369

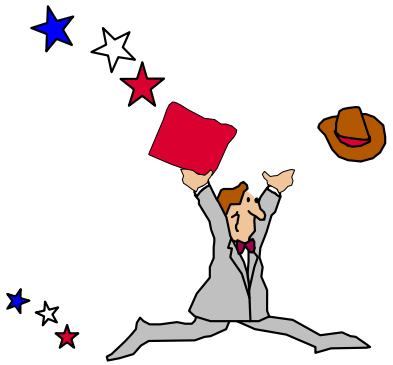
The Re-engineered IBM JVM

- The Re-engineered JVM common code base, plus Platform-specific code for
  - File handling
  - ASCII vs EBCDIC code page
  - JRIO (Java Record I/O)
  - RAS Characteristics
  - Profiling and Security APIs
  - RACF Integration
  - Hardware instruction set



- Integral Part of WebSphere Application Server platform
- Be the market leader in delivering Java technology

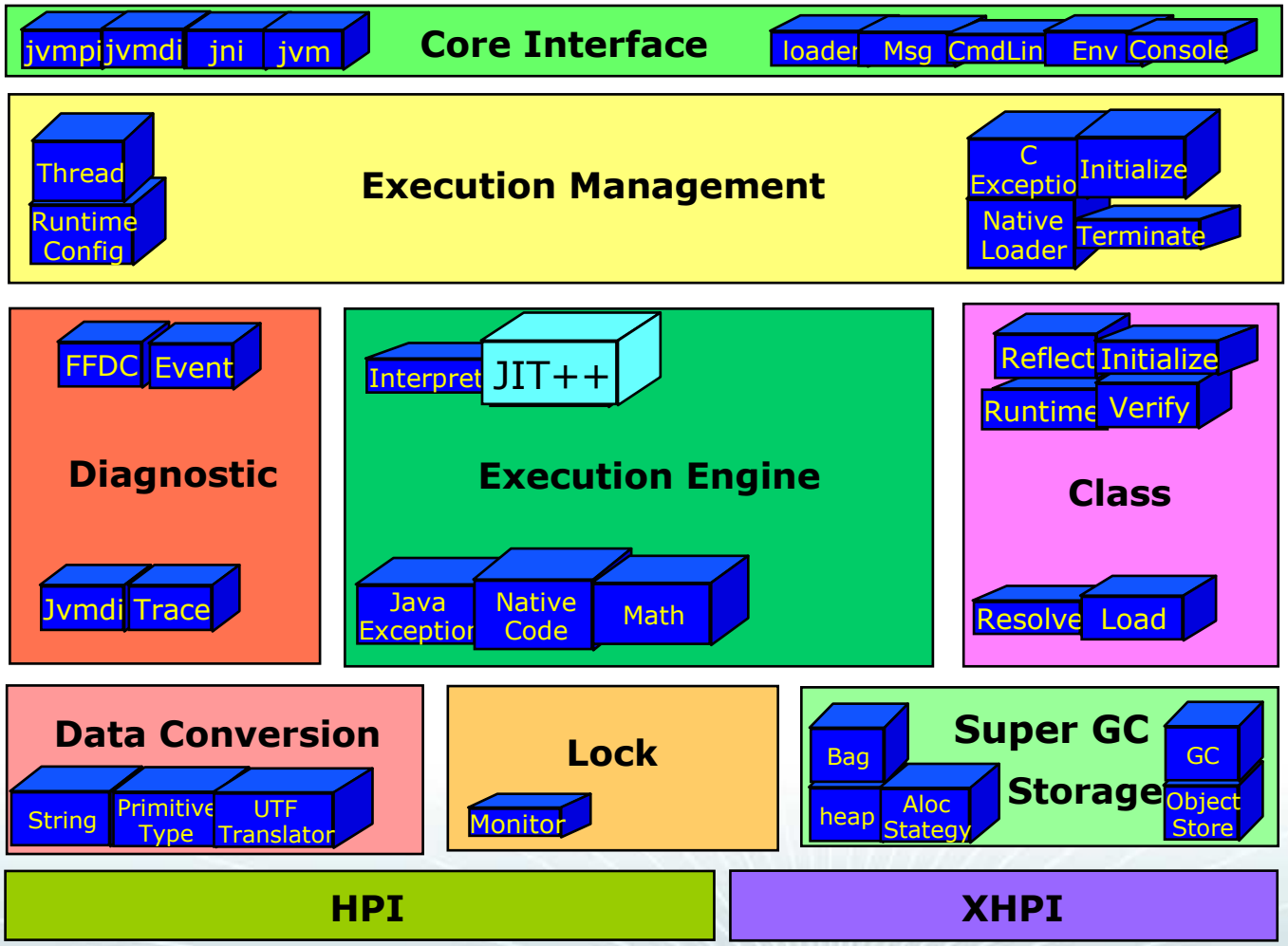
IBM Java™ 2 Objective - Value Add



- IBM Owned SDK Asset Base
- Improved Quality of IBM SDKs
  - Better development processes
  - More consistent functional implementation
  - Performance enhancements across platforms
  - IBM Technology Added Value (JVM and Classes)
  - IBM Just In Time (JIT) compiler
- Reduced cost of porting to new platforms
- Leverage new technology in both IBM hardware and software
- Continually Improving Tools for Application Development and Deployment
- Improved Performance, Scalability, RAS and improved System Exploitation

GOAL: Deliver Complete, Fully Compliant, Leadership SDKs

IBM JVM Additional Value-Add



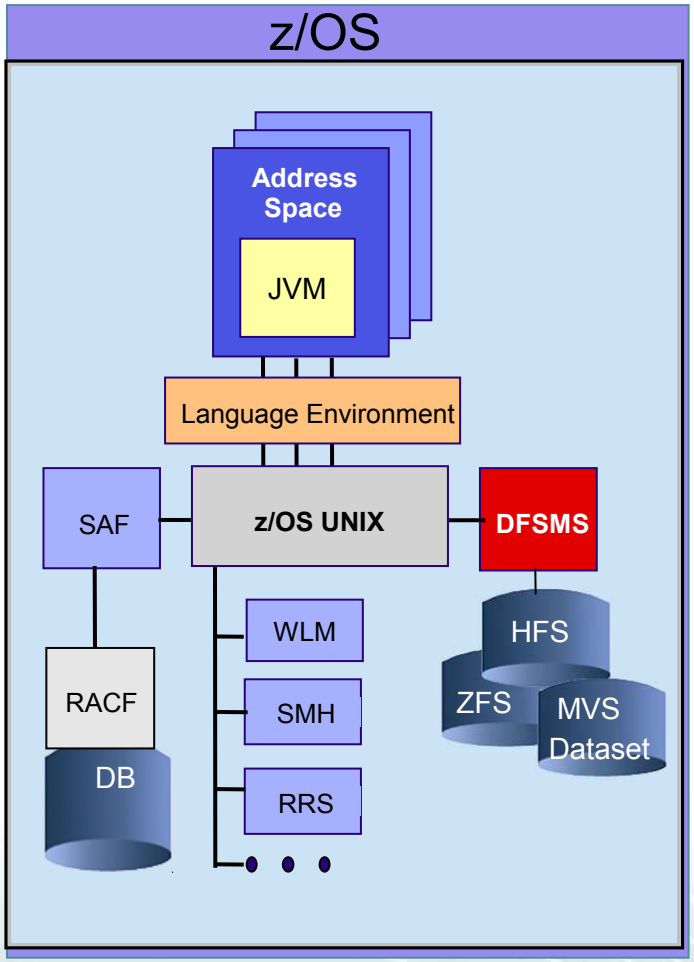
New Entry Points

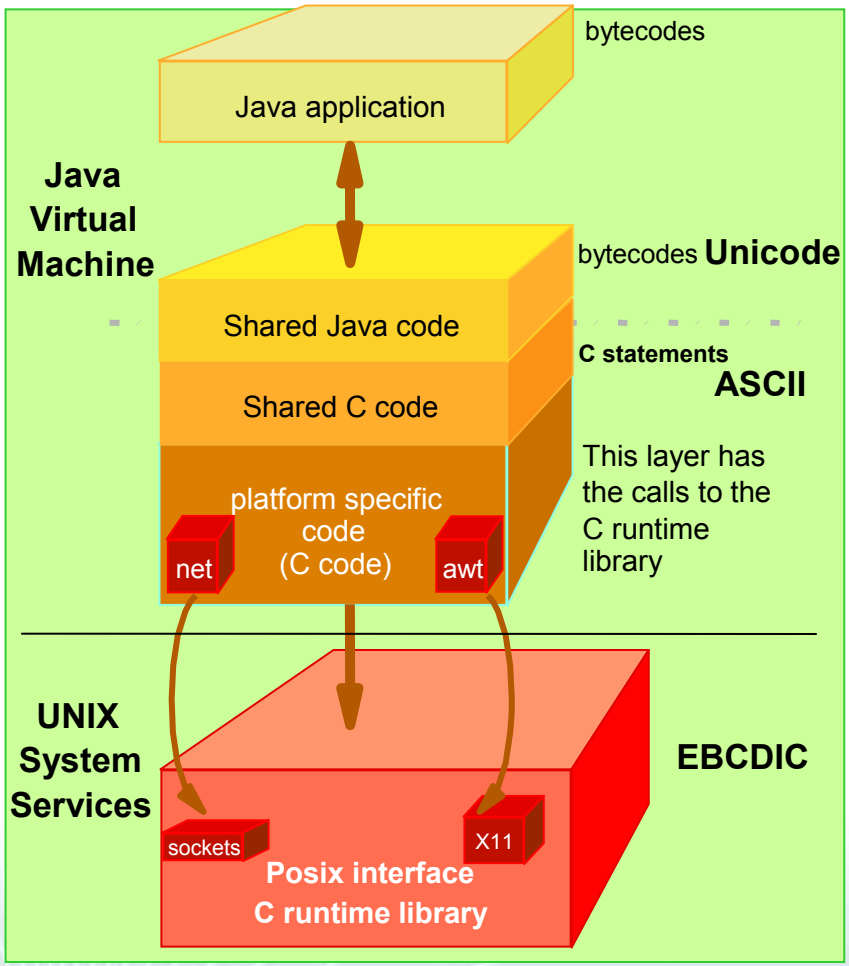
Java Strategy for zSeries Platform

- Establish Java as a "de facto" programming environment on zSeries
  - Deliver J2EE capability in synch with Java industry standards (J2EE Certified)
- Lead with z/OS Qualities of Service
  - Market leader in delivering Java technology
  - z/OS platform for mission critical workload
- Enable all "Application Execution Environments" to support Java2 based applications:
  - WebSphere Application Server
  - Transaction Servers, ie. CICS & IMS
  - DB2 data base (Stored Procedures)
  - Enable connectivity to middleware
  - Messaging queuing
  - Java Batch processing (the inclusion of Dovetailed JZOS)

Continue to provide world class Development and Deployment Tooling

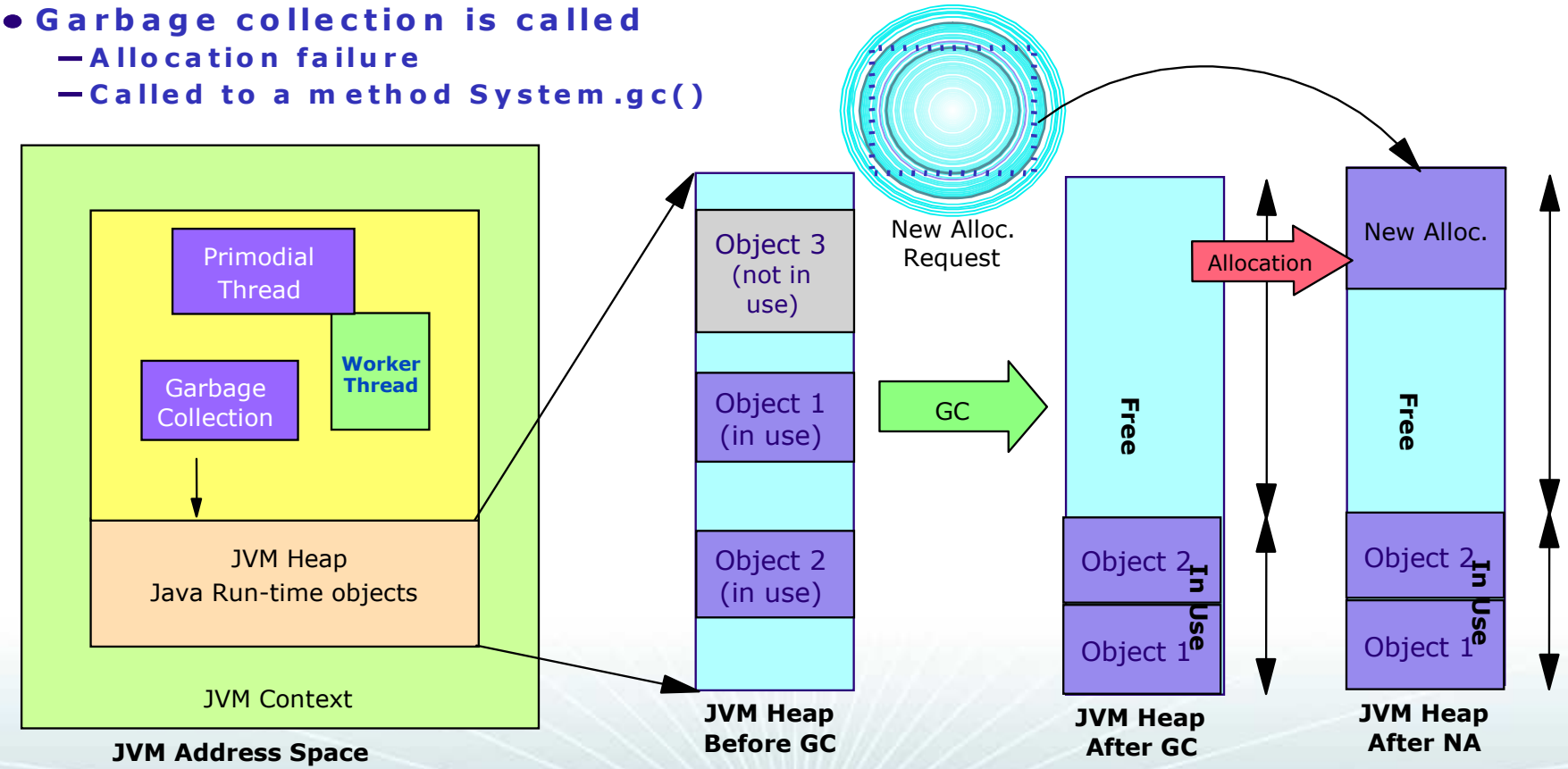
JVM on z/OS


z/OS
Address Space
JVM
Language Environment
SAF
z/OS UNIX
DFSMS
WLM
SMH
RRS
RACF
DB
HFS
ZFS
MVS Dataset


bytecodes
Java application
Java Virtual Machine
bytecodes Unicode
Shared Java code
C statements
ASCII
Shared C code
This layer has the calls to the C runtime library
platform specific code (C code)
net
awt
UNIX System Services
EBCDIC
sockets
X11
Posix interface
C runtime library

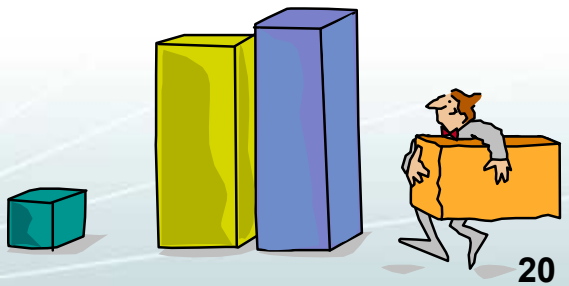
What is Garbage Collection (GC) ?

- Garbage collecton is within the JVM that automatically detects objects that are no longer being used and frees them to reclaim the space
- Garbage collection is called
  - Allocation failure
  - Called to a method System.gc()



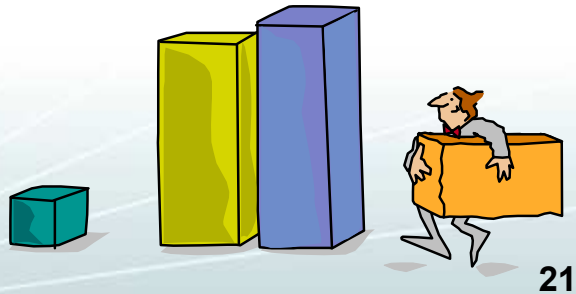
GC Evolution

- Concurrent GC (SDK 1.3) to Mark & Sweep, optional Compaction (SDK 1.3.1)
- Mark and Sweep with optional Compaction
  - Marking live objects
  - Sweeping up the garbage and coalescing it into large areas of free space
  - Compacting the remaining live objects to create a yet larger area of free space
- SDK 1.4.2 Incremental Compaction
  - Compacting a different section of each heap's GC cycle
  - Reduce pause time
- SDK V5 Generational GC
  - Separation of new and old area
  - Reduce pause time



Memory Management

- Java Heap & Garbage Collection
  - Do nothing, use defaults
    - 64MB
    - WebSphere default is 256MB for WAS V5 and V6
- Java heap size and GC frequency
  - Use small heap, frequent, fast GC
  - Use large heap, infrequent, slow GC
- Make sure you have enough physical memory to support the heap size
- Poor performance
  - -verbose:gc and -Xtgc



Java Heap Tuning & Performance Tips

- Collect verboseGC data under peak load
- Verify that your free space in the heap does not 'shrink' from GC to GC
  - Possible memory leak?
  - Look at the %Free data field
- Performance Tips
  - Make sure the JIT is enabled
  - Do not use debug version of Java
  - Make sure you only point to classes you need
  - Make your CLASSPATH efficient
    - Put frequently referenced classes first
  - Turn off JRAS debugging support
  - Set a reasonable Java heap size
  - Review GC overhead % on a regular basis

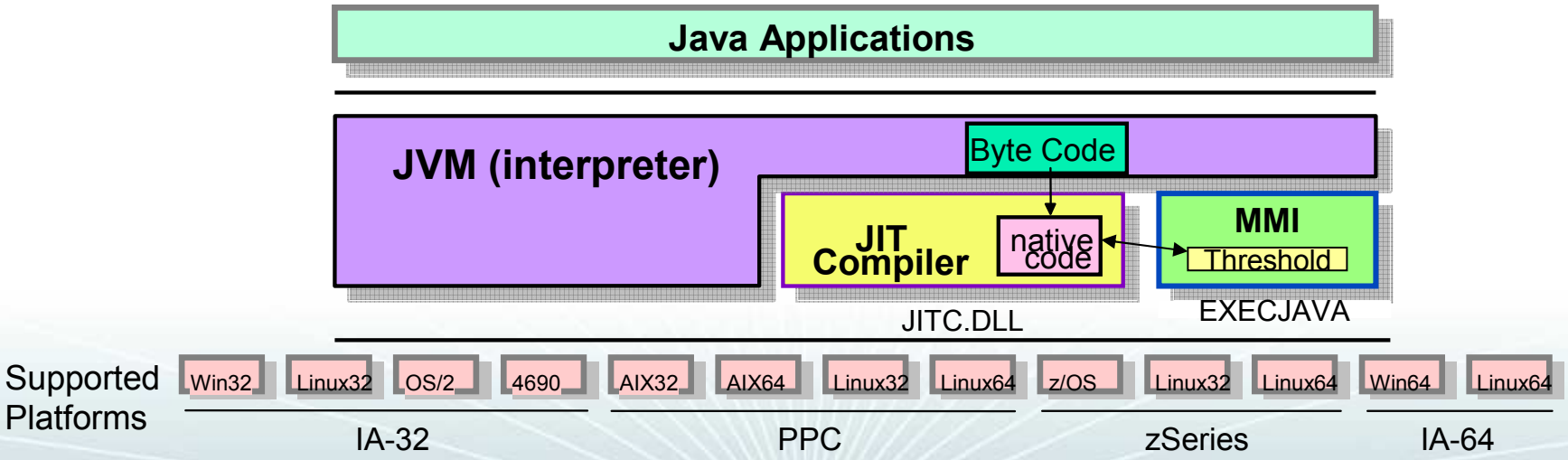
Tuning: The *Right* Java Heapsize

- Collecting -verbose:gc data (output below)
- The desirable GC overhead % is about 5%
  Reference Chapter 3 Tuning Performance Parameter Index of WAS V5.1 Information Center (JVM Storage Tuning Tips for z/OS)
  Chapter 5 Tuning Application Server Environment of WAS V6.02 Information Center
- $$\text{GC Overhead\%} = \frac{\text{elapsed time doing GC}}{\text{elapsed time from GC to GC}} * 100$$
- Example: 118/32225 * 100 = 0.366%

```
 . . .
<AF[21]: Allocation Failure. need 32784 bytes, 32225 ms since last AF>
<AF[21]: managing allocation failure, action=1 (84320/131004928) (3145728/3145728)>
<GC(21): GC cycle started Wed Aug 05 22:46:11 2003
<GC(21): freed 99587928 bytes, 76% free (102817976/134150656), in 118 ms>
 <GC(21): mark: 103 ms, sweep: 15 ms, compact: 0 ms>
 <GC(21): refs: soft 0 (age >= 32), weak 0, final 878, phantom 0>
<AF[21]: completed in 118 ms>  . . . . . .
```

IBM JIT: Just-In-Time Compiler

- What does the JIT compiler do?
  - It dynamically generates machine code for frequently used bytecode sequences in Java applications while they are running
  - To improve performance by optimizing machine code execution
- MMI (Mixed Mode Interpreter) component
  - Designed to optimize the startup time and runtime performance of Java applications
  - Using a fast Assembler byecode interpreter (EXECJAVA)



Java Record I/O

- JRIO is a class library, similar to java.io
- JRIO provides record-oriented access on z/OS
  - Virtual Sequential Access Method (VSAM) data sets (KSDS only)
  - Non-VSAM record-oriented data sets
  - The System Catalog
  - Partitioned data set (PDS) directory
  - DDName and GDGs support
    - GDG for PDS
  - SPACE and DISP parameter support
- zAAP eligibility
- How do I .... and sample programs
  - http://www-1.ibm.com/servers/eserver/zseries/software/java/jrio/recordio/examples/index.html

IBM RAS for Java

- JVM initiated System Dumps
  - Environment variable JAVA_DUMP_OPTS available
    - In the form of LE CEEDUMPs, core dumps, JAVADUMPs Javaheap dump
  - Rewritten Dump Formatter (IPCS JVMDATA)
- Improved tracing and problem determination
  - JVM Exception handler improvements
  - Method Level Trace
  - Application Trace
    - Call Java trace from Java Programs
  - JVM Monitoring Interface JVMMI
    - Lightweight version of JVMPI
  - Robust termination
  - GTF trace for a JITed Java method
  - Hook to interface with MVS dynamic slip trap to collect dumps/traces
  - Java Debugger (JDB) part of JDPA

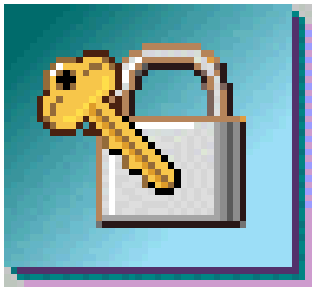
- **IBM JVM Diagnostics Guide**
  - http://www-128.ibm.com/developerworks/java/jdk/diagnosis/

zAAP: New Java Workload Engine on z/OS

- zAAP (zSeries Application Assist Processor)
- New processor type on z890, z990, z9-109 hardware supporting z/OS
- Order using Feature Code 6520
  - You can order up to one zAAP per configured or unassigned standard CP on the processor
- A specialized z/OS and z/OS.e Java execution environment for Java-based applications
  - WAS V5.1
  - CICS/TS V2.3
  - DB2 V7 and V8, IMS V8
  - WebSphere WBI for z/OS
- Newly instrumented IBM JVM
  - With no anticipated modifications to Java application
- Require z/OS V1R6 and SDK 1.4 with PTF UQ88783
- Usage projection
  - z/OS V1.4 or z/OS V1.5 and SDK1.4 with PTF UQ88783 (recommended)
  - SDK1.3.1 with PTF UQ94379
  - z/OS V1.6 RMF report to collect the 'Would Have Been" zAAP usage

Session 8366 – Introduction to IBM Java Workloads Engine zAAP

What is Java Security?

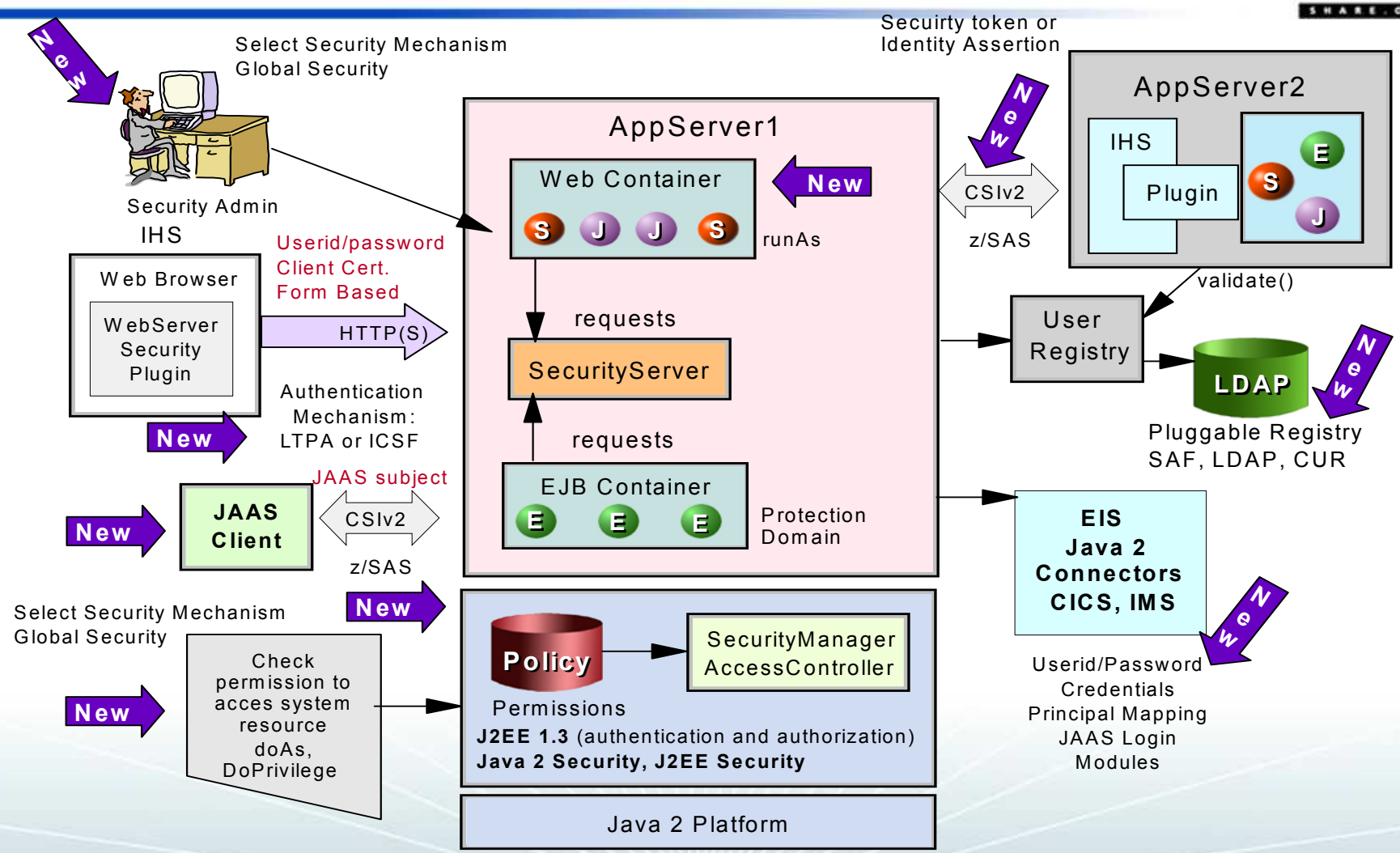


- Part of Java 2 framework
  - A set of common cross platform programming API's
- Java security extensions
  - A set of common API's to extend Java 2 to add security capabilities
- Allow Java applications easy access to complex security capabilities within Java framework
- Java security extensions added to base Java 2 (J2SE) framework with SDK 1.4

Java Security Components on z/OS

- **JAAS**: Java Authentication and Authorization
- **JCE**: Java Cryptographic Extension using CCA (Common Cryptographic Architecture)
- **JCE4758**: IBM zSeries HW Cryptographic Device
- **JSSE**: Java Secure Socket Extension support SSL and TLS
- **SAF Interface**: z/OS Security Services in Java
  - Implement a SAF interface using JNI to call SAF/RACF

J2EE and Java 2 Security



IBM Rational Application Developer (RAD)

- RAD is the new development tool for applications
  - Rational Software Development Platform (Eclipse 3.x)
  - RAD - Rational Application Developer (V6.1)
  - RWD – Rational Web Developer (V6.x)
  - A follow-on products for
    - WSAD (WebSphere Studio Application Developer)
    - WSSD (WebSphere Studio Site Developer)
- Maintain standards and middleware currency
  - Full support for building J2EE 1.4 applications
    - SDO, JSF, Portal, Modeling, Deploying, Profiling and Testing
  - Support deployment to WAS V5, V5.1 and V6
  - Supports J2EE 1.2 and 1.3 applications migrated to 1.4
- Extends on the features of WSAD

Profiling Tools

- RAD Profiler
  - Memory analysis features to locate memory leaks in heap dumps
  - New ProbeKit
    - Easy profiling at key points in the application using byte-code instrumentation
    - Don't have to recompiling the classes to be profiled
    - Summary report on % on amount of code is executed
- RAD Application Analysis
  - Applications can be reviewed for coding practices
    - For GC, performance, scalability, portability, serialization and thread usage
  - Allow developers to create their own rules based templates
- WebSphere PMI (Performance Monitoring Infrastructure)
  - JMX MBeans, byte-code implementation
  - It collects statistics on activities in the server
  - Requires the TPV (Tivoli Performance Viewer)

The New JAXP (Java API for XML Processing) 1.3

- Standardized set of Java Platform APIs for parsing and manipulating XML documents
  - XML parsing (javax.xml.parsers)
    - SAX 2.0.2 (org.xml.sax)
    - DOM Level 3 (org.w3c.dom)
  - XML / Java type mappings (javax.xml.datatype)
  - XML validation (javax.xml.validation)
  - XSLT transformation (javax.xml.transform)
  - XPath (javax.xml.xpath)

- Provides all of the XML processing services included in IBM Java SDK 5.0
- Earlier versions of JAXP were included in Java SDK 1.4
- IBM implementation of JAXP 1.3 in Java SDK 5.0
  - parsing / datatype / validation services provided by XML4J
  - transformation / XPath services provided by XSLT4J

http://www.**ibm.com**/servers/eServer/zseries/software/xml/download/
http://www-03.ibm.com/servers/eserver/zseries/software/xml/perform/

Introduction to XML4J and XSLT4J

- Based on Apache Xerces Java & Xalan Java
  - IBM contributes the majority of code to these open source projects
  - Other contributions have been from Sun and individuals
- Java SDK 5.0 contains XML4J 4.4 & XSLT4J 2.7
  - Completely contained in <JAVA_HOME>/jre/lib/xml.jar
- Both development teams are in Toronto and provide L3 support
- Also released as standalone components
- Maintenance releases are cumulative and contain fixes written by IBM employees only

Note: only the IBM versions of XML4J and XSLT4J are supported

Summary

- Continue to provide z/OS SDK technology base for
  - WebSphere, CICS, IMS, and DB2
  - Linux Middleware
  - ISVs
- Continued rollout of Java2 including new IBM architectures to allow better platform integration, function, tailoring and performance
- Recommendation: Stay Current by visiting our web site Frequently
  http://www.ibm.com/servers/eserver/zseries/software/java
- Reporting a problem
  http://www-1.ibm.com/servers/eserver/zseries/software/java/services.html

Thank You