



Java and Java Batch applications on the IBM Mainframe

Christian Strauer
Field Technical Sales Support IBM System z9
Christian.Strauer@de.ibm.com

© 2007 IBM Corporation

IBM Systems



Agenda

- **About the mainframe**
- **Java runtime environments under z/OS**
 - ▶ IBM Java SDK for z/OS
 - ▶ z/OS Java APIs
 - ▶ WebSphere application server for z/OS
- **For which applications should I use a mainframe?**
 - ▶ Stand-alone Java applications
 - ▶ J2EE applications
- **Java Development for z/OS**
- **Monitoring Java applications with JMX**
- **Summary and literature**

IBM System z9 – The IBM Mainframe

- 40 years of evolution
- Enterprise server for **highest availability**
- High efficient **workload management**
- Capacity on Demand Offerings
- Outstanding **security** concept:
 - ▶ Key management
 - ▶ Encryption
 - ▶ Data integrity
- **Virtualisation** engine
 - ▶ LPAR concept (EAL5 certified)
 - ▶ z/VM



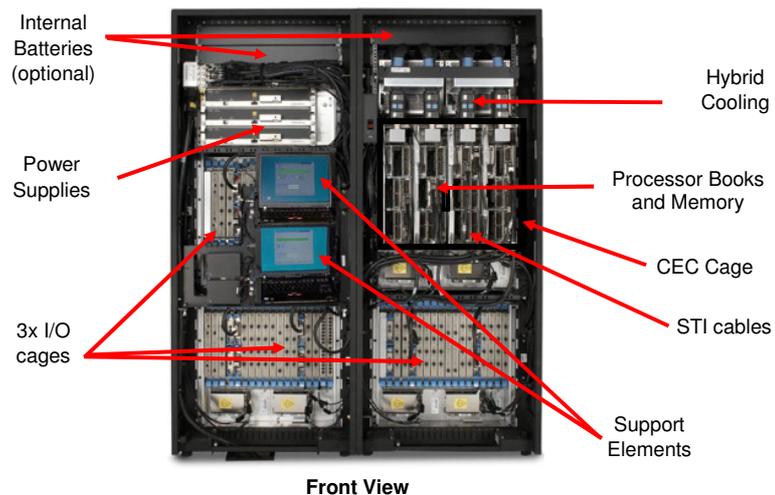
3

Java and Java Batch on IBM System z9

01/06/2007

IBM Systems

Front View of a System z



4

Insert Footer in Master View

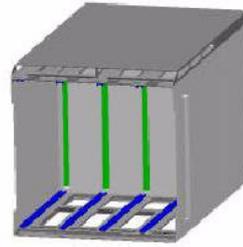
01/06/2007

IBM Systems

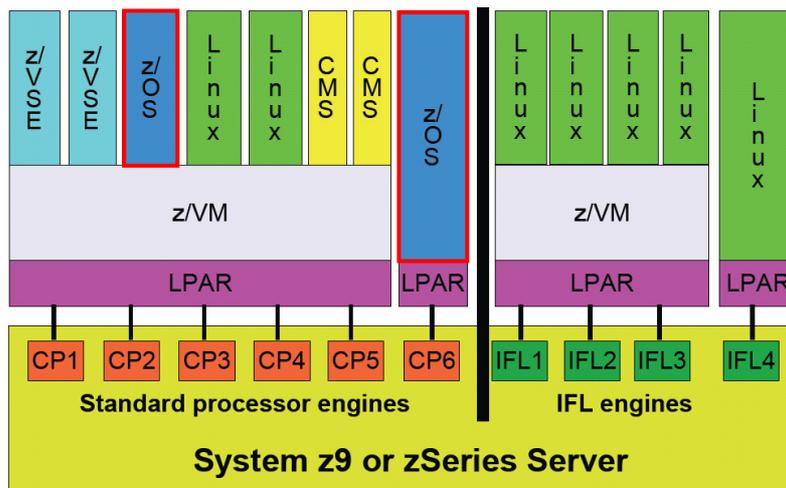
What is a “book”

▪ A System z9 book consists of

- ▶ Multi-Chip-Module: Processor Units (PUs)
 - 12 PUs (16 for Model S54) per book
 - 2 SAPs per book, standard
 - 2 spares per server
 - 8, 18, 28, 38 or 54 PUs available
 - CPs, IFLs, ICFs, zAAPs, zIPs, optional SAPs
- ▶ Memory
 - Minimum of 16 GB
 - Up to 128 GB per book
 - 16 GB increments
- ▶ I/O
 - Up to 16 STIs per book
 - 2.7 GB/s for each I/O
 - Total system I/O bandwidth capability of 172.8 GB
 - Up to 4 Logical Channel SubSystems (LCSSs) and 60 LPAR (EC)

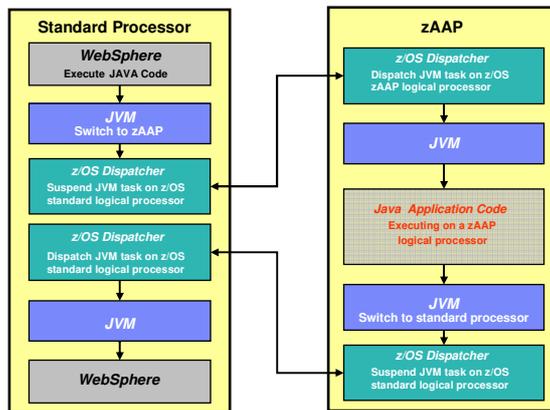


IBM System z9 – Operating systems



System z application assist processor (zAAP)

- A special priced engine for Java workload on the mainframe



7

Java and Java Batch on IBM System z9

01/06/2007

IBM Systems

JVM 5.0 – A complete new JVM for z/OS

- Sun IP-free, but Java 2 (1.3) compliant (J2ME) and J2SE (1.4.2, 5.0)
- Common code base across all platforms
 - ▶ PowerPC, IA32, x86-64, and 390 (Linux or z/OS)
- Flexible and sophisticated technology oriented to:
 - ▶ Performance (throughput and application startup)
 - ▶ Scalability
 - ▶ Reliability and Servicability (RAS)

9

Java and Java Batch on IBM System z9

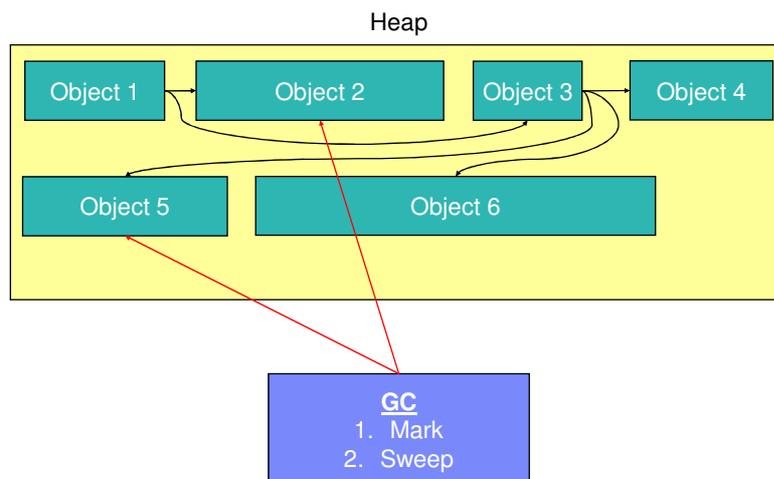
01/06/2007

IBM Systems

JVM 5.0 – Just-in-time compiler

- The just-in-time compiler (JIT) is not really part of the JVM, but is essential for a **high performing Java application**
- Java is Write Once Run Anywhere thus it is interpreted by nature and without the JIT could not compete with native code applications
- Due to different platforms having different JITs there is no standard method for when a method is compiled.
- As your code accesses methods the **JIT determines how frequently specific methods are accessed** and compiles those touched often quickly to optimize performance
- **-Xquickstart** helps to improve JVM startup time for short running Java applications
 - ▶ causes the JIT to run with a subset of optimizations

JVM 5.0 – Garbage collection



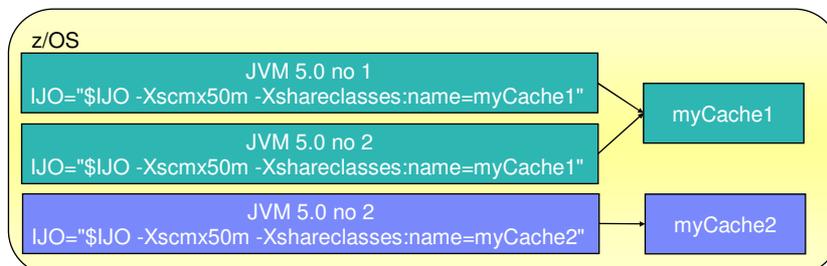
JVM 5.0 – Garbage collection

- **Memory management is configurable using four different policies with varying characteristics**
 1. **Optimize for Throughput** – flat heap collector focused on maximum throughput
 - `-Xgcpolicy:optthruput`
 2. **Optimize for Pause Time** – flat heap collector with concurrent mark and sweep to minimize GC pause time
 - `-Xgcpolicy:optavgpause`
 3. **Generational Concurrent** – divides heap into “nursery” and “tenured” segments providing fast collection for short lived objects. Can provide maximum throughput with minimal pause times
 - `-Xgcpolicy:gencon`
 4. **Subpool** – a flat heap technique to help increase performance on multiprocessor systems , commonly greater than 8. Available on IBM pSeries™ and zSeries™
 - `-Xgcpolicy:subpool`

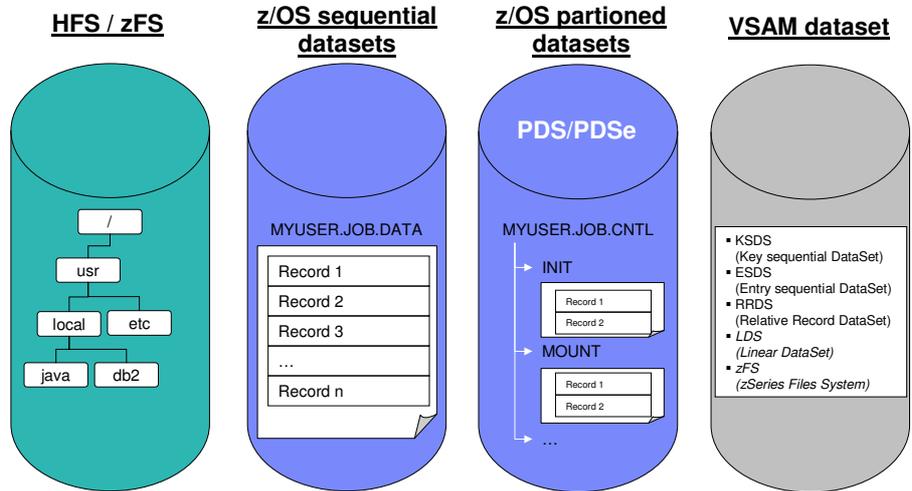


JVM 5.0 – Shared classes

- **A shared class area for one or more JVMs**
- **Improves startup time**
 - ▶ Lots of classes are already preloaded



A short z/OS data overview



15

Java and Java Batch on IBM System z9

01/06/2007

IBM Systems

Special z/OS APIs in the IBM Java SDK for z/OS

- **MVS dataset access**
 - ▶ stream mode/record mode
 - ▶ DD statements possible
 - ▶ APIs: ZFile/JRIO

```

ZFile zFile = new ZFile("//DD:INPUT", "rb,type=record,noseek");
try {
    byte[] recBuf = new byte[zFile.getLrecl()];
    String enc = Zutil.getDefaultPlatformEncoding();
    int nRead;
    while((nRead = zFile.read(recBuf)) > 0) {
        String line = new String(recBuf, 0, nRead, enc);
        System.out.println(line);
    };
} finally {
    zFile.close();
}
    
```

Based on JNI

- **USS file access**
 - ▶ Normal file access like in Unix or Linux with java.io

16

Java and Java Batch on IBM System z9

01/06/2007

IBM Systems

MVS console communication

▪ MVS console commands

- ▶ Start: `/s jobName,commands`
- ▶ Modify: `/f jobName,commands`
- ▶ Stop: `/p jobName`

▪ Code:

```
MvsConsole.registerMvsCommandCallback(new MvsCommandCallback() {  
    public void handleModify(String s) {  
    }  
  
    public void handleStart(String s) {  
    }  
  
    public boolean handleStop() {  
        return true;  
    }  
});
```

17

Java and Java Batch on IBM System z9

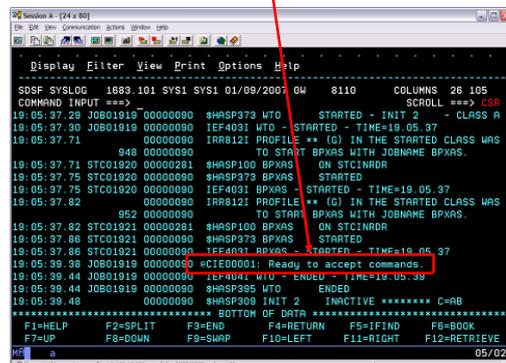
01/06/2007

IBM Systems

MVS console communication

▪ Write To Operator (WTO) API available

```
{  
    MvsConsole.wto("CIE00001: Ready to accept commands.",0x0020, 0x4000);  
}
```



The screenshot shows a terminal window with the following text:

```
SDSF SYSLOG 1683.101 SYS1 SYS1 01/09/2007 GW 8110 COLUMNS 26 105  
COMMAND INPUT ==> SCROLL ==> CSR  
19:05:37.28 JOB01919 00000090 $HASP373 WTO STARTED - INIT 2 - CLASS A -  
19:05:37.30 JOB01919 00000090 IEF4091 WTO STARTED - TIME=19.05.37  
19:05:37.71 00000090 IRR8121 PROFILE ** (G) IN THE STARTED CLASS WAS U  
948 00000090 TO START BPXAS WITH JOBNAME BPXAS.  
19:05:37.71 STC01920 00000281 $HASP100 BPXAS ON STCINRDR  
19:05:37.72 STC01920 00000090 $HASP373 BPXAS STARTED  
19:05:37.75 STC01920 00000090 IEF4081 BPXAS - STARTED - TIME=19.05.37  
19:05:37.82 00000090 IRR8121 PROFILE ** (G) IN THE STARTED CLASS WAS U  
952 00000090 TO START BPXAS WITH JOBNAME BPXAS.  
19:05:37.82 STC01921 00000281 $HASP100 BPXAS ON STCINRDR  
19:05:37.82 STC01921 00000090 $HASP373 BPXAS STARTED  
19:05:37.86 STC01921 00000090 IEF4091 BPXAS - STARTED - TIME=19.05.37  
19:05:39.38 JOB01919 00000090 #CIE00001: Ready to accept commands.  
19:05:39.44 JOB01919 00000090 IEF4041 WTO ENDED - TIME=19.05.39  
19:05:39.44 JOB01919 00000090 $HASP395 WTO ENDED  
19:05:39.48 00000090 $HASP300 INIT 2 INACTIVE ***** C=AB  
***** BOTTOM OF DATA *****  
F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK  
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE  
05/021
```

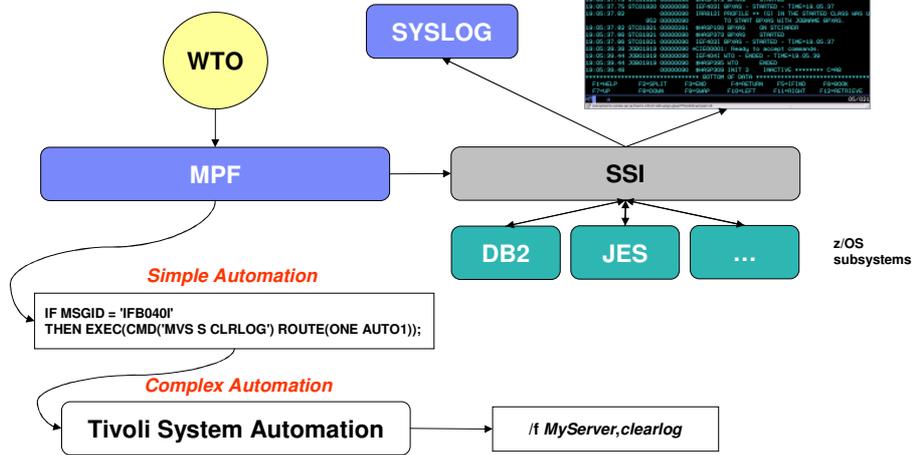
18

Java and Java Batch on IBM System z9

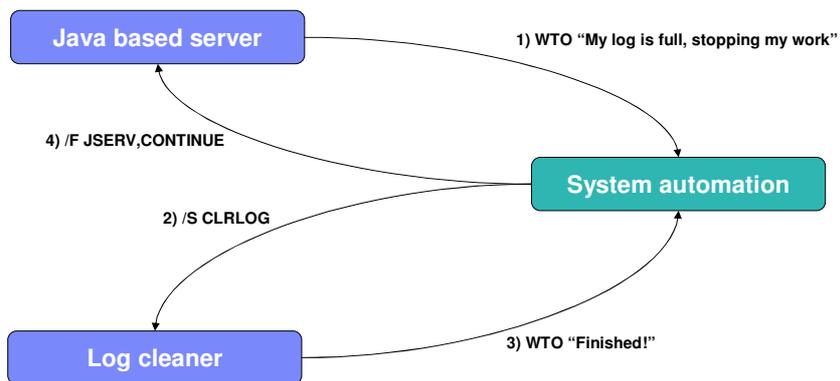
01/06/2007

IBM Systems

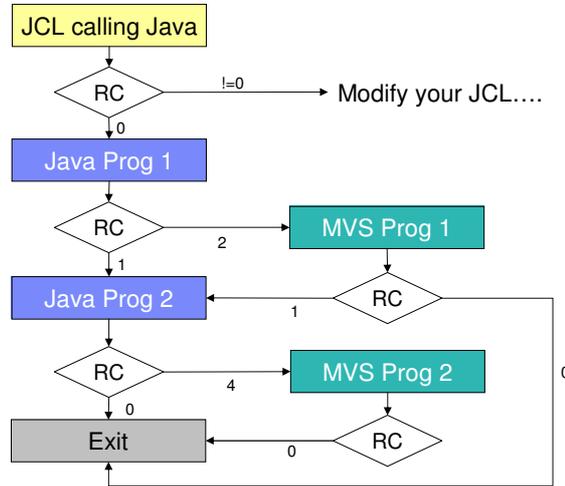
z/OS message processing



Java and z/OS console communication



Condition Code passing for z/OS job nets



21

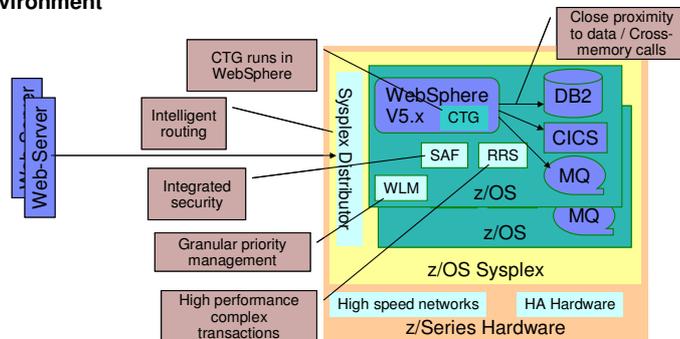
Java and Java Batch on IBM System z9

01/06/2007

IBM Systems

WebSphere Application Server for z/OS

- ... A normal J2EE compliant application server, but with additional features:
 - ▶ WAS Common code base across all platforms
 - ▶ Under z/OS, there is a tight integration into the mainframe specific environment



23

Java and Java Batch on IBM System z9

01/06/2007

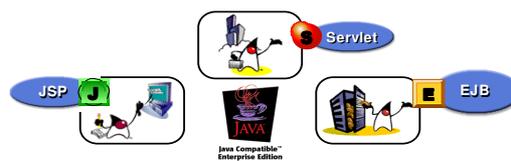
IBM Systems

For which Java applications does a mainframe fit?

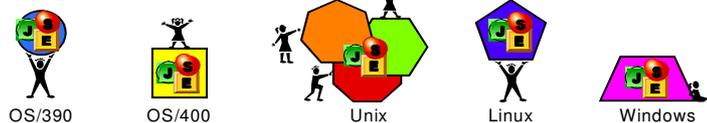
- **Batch** is still one of the mainframes biggest strengths
 - ▶ The mainframe was designed for batch (punch cards)
 - ▶ The mainframe has the longest experience in the batch environment
 - ▶ Special facilities in z/OS allow a huge complex job management for batch jobs (JES, SDSF,...)
 - ▶ Java inherits all these functionalities
- **Business critical** Java based servers that need:
 - ▶ High availability (99,999%)
 - ▶ Best security
- Java applications which use lots of **transactions**
 - ▶ Data proximity



J2EE: Which is the best platform to deploy?



... you can deploy them virtually anywhere ...

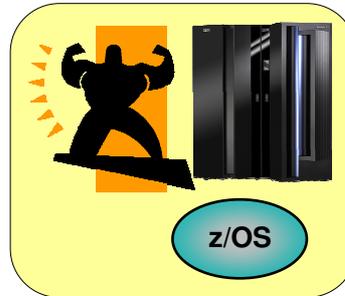


... but which option will best suit your needs??

*During development, J2EE components are not optimized to any deployment platform
The server, operating system and WAS determine how well those components meet
business objectives in production*

The value of the mainframe for J2EE applications

- **Strategic platform for mission-critical business processes, core business applications and data**
 - ▶ **Proximity:** Fewer physical components and network connections result in less complex infrastructure
 - ▶ **Integrity:** System-managed resource recovery and transaction coordination
- **Highest possible Qualities of Service**
 - ▶ **Security:** Most stringent access control and asset protection
 - ▶ **Availability:** zero downtime z/OS brand promise
 - 99.999% availability at the application and data level
 - Managed workload sharing toward achievement of business goals
 - Dynamic application of software changes
 - ▶ **Efficiency:** Full utilization of system capacity with same class of service
 - Less people to configure, monitor and adjust workload
 - Best Total Cost of Computing
 - zSeries Application Assist Processors (zAAPs) to execute Java workloads under z/OS



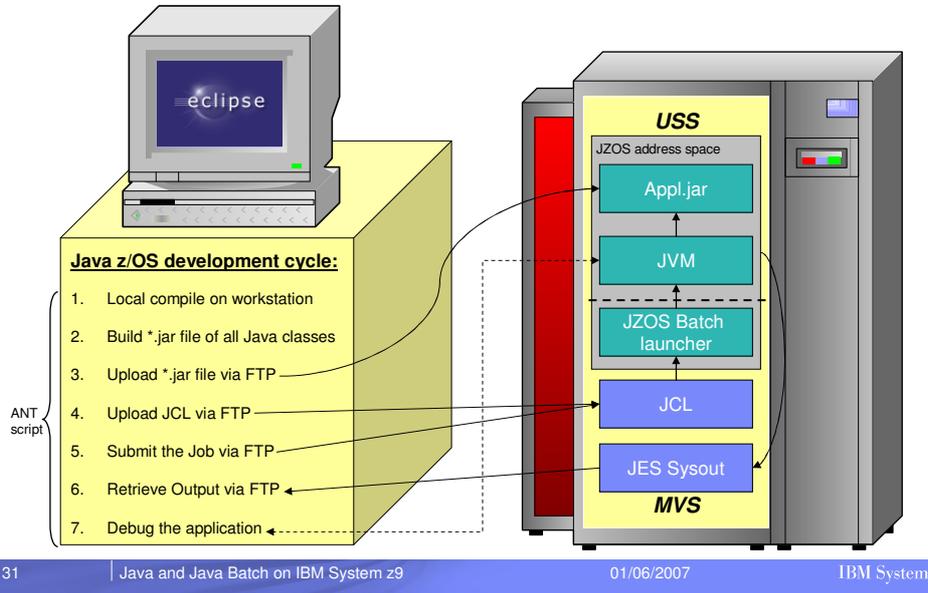
How do I develop z/OS Java applications?

- **This is a picture people often associate with the mainframe:**

 A screenshot of a z/OS terminal window. The window title is 'Session A: [24 x 50]'. The terminal output shows a directory listing for the user 'ZUSER02' in the path '/usr/lpp/'. The listing includes various system components like 'WFS', 'Printsrv', 'booksrv', 'bpa', 'cbclib', 'cicsta', 'cim', 'cmx', 'db2810', and 'db2810_msys'. Below the listing, the user enters the command 'cd /usr/lpp/' and then 'ls'. The output shows a list of files and directories including 'db2_08_01', 'db2ext_08_01_00', 'db2nx', 'internet', 'dce', 'java', 'dfig', 'jccit4', 'dfms', 'load', 'eim', 'ldapclient', 'fu', 'te', 'gskasl', 'local', and 'zWebSphere'. The user then enters 'cd java/java50/j5.0' and 'java HelloWorld_'. The terminal shows '====> java HelloWorld_ RUNNING' and a list of keyboard shortcuts at the bottom.

- **... But it is much easier!**
 - ▶ Eclipse as an IDE can be easily used for Mainframe Java development

Development tools for Java Batch: 1) Eclipse



31

Java and Java Batch on IBM System z9

01/06/2007

IBM Systems

Development tools for Java Batch: 2) WDz

- Based on Eclipse
- Inherits Eclipse functions plus:
 - ▶ JES integration
 - ▶ J2EE and Web programming
 - ▶ UML modeling
 - ▶ Cobol, PL/I and C/C++ development
 - ▶ CICS Web Service Support
 - ▶ Web Service Test Client Generation
 - ▶ IBM Debug Tool integration
 - ▶ ...

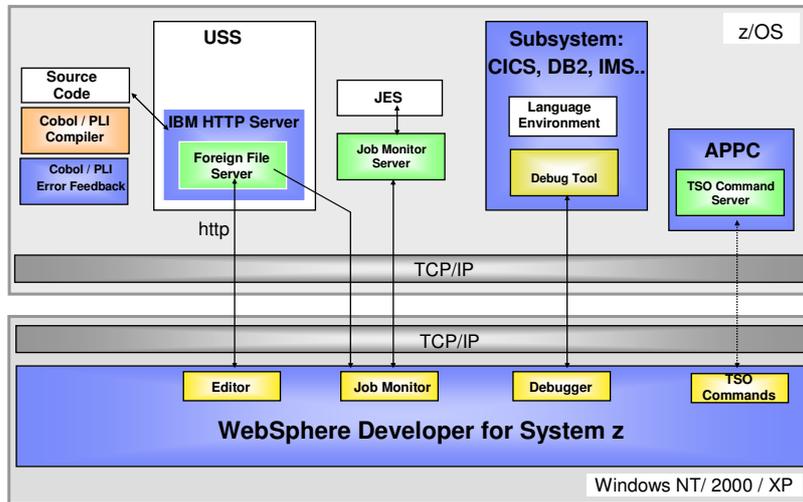
32

Java and Java Batch on IBM System z9

01/06/2007

IBM Systems

Development tools for Java Batch: 2) WDz



33

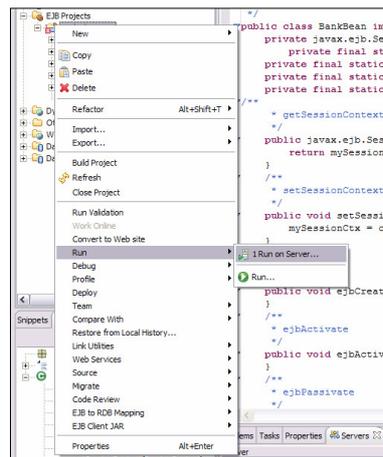
Java and Java Batch on IBM System z9

01/06/2007

IBM Systems

WDz and J2EE development

- WDz is put on top of Rational Application Developer
- Integrated WebSphere Application Server test environment
- Remote deployment of applications
- Wizards for EJB creation
- EJB Test client



Server	Host name	Status	State
WebSphere v6.0 Server @ 192.168.7.226	192.168.7.226	Started	Synchronized
WebSphere v6.0 Server @ localhost	localhost	Stopped	Synchronized

34

Java and Java Batch on IBM System z9

01/06/2007

IBM Systems

WDz and J2EE / Web development

- **Web Development:**
 - ▶ JSF support
 - ▶ Web site designer
 - ▶ JSP and servlet wizard
 - ▶ Java Visual Editor for JSF Component Layout
 - ▶ Portlet development
- **Web Service development**
 - ▶ Web Services Wizard
 - ▶ Web Services Explorer
 - ▶ Generation of Cobol Web Services

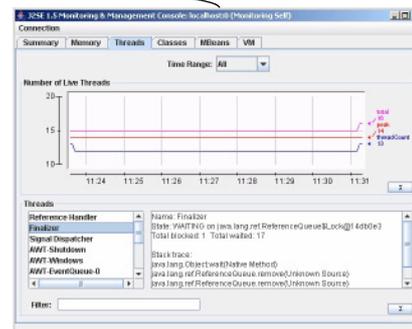
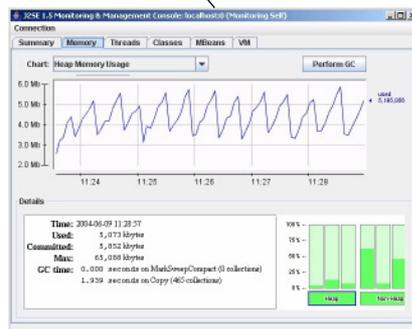
Resource access and backend integration

- System resources like the security facility RACF can be accessed via easy Java APIs
- The mainframe provides special, well approved **transaction monitors** (CICS and IMS) that can easily be integrated into Java applications
- **Local connectors**, that use cross-memory functions allow **high transactional performance computing**
 - ▶ Example: Local JDBC Driver for DB2 z/OS

Monitoring Java applications with JMX



```
IJO="$IJO -Dcom.sun.management.jmxremote.port=9090"  
IJO="$IJO -Dcom.sun.management.jmxremote.authenticate=false"  
IJO="$IJO -Dcom.sun.management.jmxremote.ssl=false"
```



38

Java and Java Batch on IBM System z9

01/06/2007

IBM Systems

Summary

- **The mainframe is the ultimate server for enterprise and mission critical applications**
- **There are more or less no differences in the development of Java applications between distributed servers and mainframes**

40

Java and Java Batch on IBM System z9

01/06/2007

IBM Systems

Literature

- **IBM z/OS Java Homepage**
<http://www-03.ibm.com/servers/eserver/zseries/software/java/>
- **Redbooks:**
 - ▶ **Java Stand-alone Applications on z/OS, Volume I**
<http://www.redbooks.ibm.com/abstracts/sg247177.html>
 - ▶ **Java Stand-alone Applications on z/OS, Volume II**
<http://www.redbooks.ibm.com/abstracts/sg247291.html?Open>
- **IBM JVM 5.0 Diagnostics Guide**
<http://download.boulder.ibm.com/ibmdl/pub/software/dw/jdk/diagnosis/diag50.pdf>
- **IBM z/OS Java SDK 5.0 User Guide**
<http://www-03.ibm.com/servers/eserver/zseries/software/java/pdf/sdkguide.zos.pdf>



Free z/OS Java Workshops in Germany

- **From zero to zHero: System z optimized Java applications**

▶ Target audience: developers

Stuttgart	München	Düsseldorf	Frankfurt	Hamburg
23.04-25.04	07.05-09.05	25.06-27.06	16.07-18.07	20.08-22.08

- **From zero to zHero: System z modernized applications**

▶ Target audience: developers

Stuttgart	München	Düsseldorf	Frankfurt	Hamburg
26.04-27.04	10.05-11.05	28.06-29.06	19.07-20.07	23.08-24.08

- <http://www.ibm.com/de/events/fromzerotozero>

Questions

