#SHAREorg

SHARE

Getting Started with WebSphere Batch
(a.k.a. Compute Grid) on WAS V8.5 for z/OS

John Hutchinson
Mike Loos
IBM

March, 2014, Anaheim
Session Number 14692

SHARE in Anaheim

WebSphere Application Server on System Z

Session	Title	Time	Room	Speaker
14618	Getting Started with WebSphere Liberty Profile on z/OS	Monday 9:30	Grand Ballroom Salon C	Loos/Follis
14692	Getting Started with WebSphere Compute Grid	Tuesday 9:30	Grand Ballroom Salon J	Hutchinson/Loos
14693	Using WebSphere Application Server Optimized Local Adapters (WOLA) to Migrate Your COBOL to zAAP-able Java	Wednesday 9:30	Grand Ballroom Salon K	David Follis
14620	WebSphere Liberty Profile on Windows AND z/OS (among other things) Hands-on Lab	Wednesday 1:30	Platinum Ballroom Salon 7	
14949	Tips Learned Implementing Websphere Application Server (WAS) on Linux for IBM System z	Wednesday 3:00	Grand Ballroom Salon G	Eberhard Pasch
14709	Need a Support Assistant? Check Out IBM's! (ISA)	Thursday 8:00	Grand Ballroom Salon A	Mike Stephen
15050	z/OSMF 2.1 Implementation and Configuration	Thursday 8:00	Grand Ballroom Salon G	Greg Daynes
14832	Web Apps using Liberty Profile Technology in CICS	Thursday 11:00	Platinum Ballroom Salon 2	Ian Mitchell
14722	Assimilating WebSphere Application Server into your z/OS WLM Configuration	Thursday 1:30	Orange County Salon 1	David Follis
15017	Using IBM WebSphere Application Server and IBM WebSphere MQ Together [z/OS & Distributed]	Thursday 3:00	Grand Ballroom Salon A	Ralph Bateman

Do you have WebSphere Application Server V8.5?

... Then you already have WebSphere Batch ! ("Compute Grid")

Get started with some basic Batch applications!

- **Java Batch options**
  - JZOS, WAS Feature Packs, z/OS Batch Container, or . . .
  - WebSphere Batch ("Compute Grid") built into WAS V 8.5
- **Important Features in WebSphere Batch**
  - Integration with Schedulers, CICS, COBOL, PJM, WLM, SMF
- **Development Tools**
  - Batch Framework & Supporting Classes
- **Choosing your 1st Application** ("Proof Of Concept")
  - Transform traditional JES batch jobs into WAS batch.

Different Approaches for Java Batch . . .

Standalone Java Program
- Simple programming model
- Launch with java command at the shell
- Programmer responsible for everything

JVM Launcher Solutions
- JZOS, BPXBATCH
- JVM Launcher in a JES Batch Job + JZOS Services for z/OS functions
- Programmer must code functions not provided by Launcher

All are perfectly good approaches, depending on the nature of your batch processing needs.

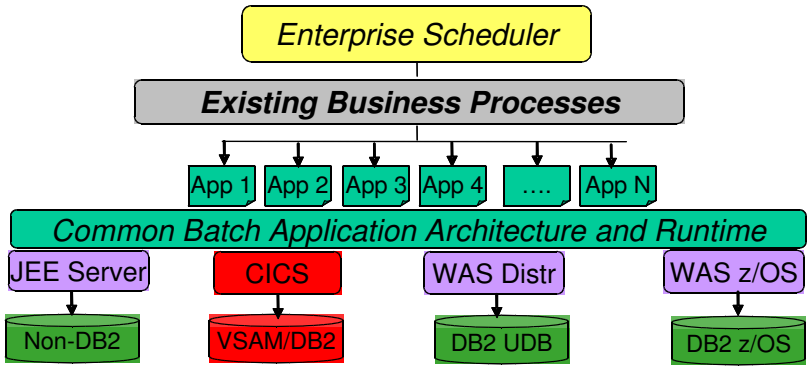
z/OS Batch Execution Runtime
- z/OS 1.13 & V2.1 Batch Container
- Batch Container provides useful functions with JDBC & RRS
- Programmer must code functions not provided by Container

WebSphere Batch Container
- WebSphere Compute Grid or WebSphere 8.5
- Provides batch programming function as services of the platform + Batch Scheduler
- Allow programmer to focus on the business logic, not "middleware" functions

CICS Batch Container
- CICS TS 4.2, 5.1 w/ Batch Feature Pack
- Batch Container provides same functions as WebSphere Batch Container + JCICS
- Jobs must be scheduled by the WebSphere Batch Dispatcher

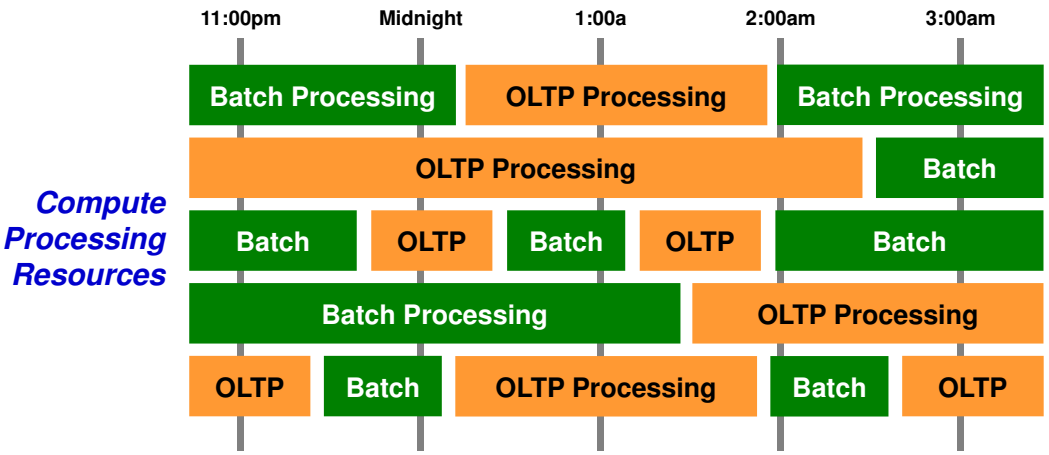
The Batch Vision:

- ✔ Run anywhere, any time.
- ✔ Portable across platforms.
- ✔ Locate near data.
- ✔ Concurrent Access to Data with Online Systems
- ✔ Centrally Managed by Enterprise Scheduler.
- ✔ Integrate with existing Workload Management, Accounting, Disaster Recovery, Auditing, Logging, Archiving.



The Objective -- OLTP and Batch Mixed and Managed:

OLTP and Batch do not need to be "either / or" ... it can be "both":



Possible with IBM WebSphere Batch.

OLTP and Batch processing within a common execution runtime (WebSphere Application Server) allows the WAS platform to mix and manage the two workload types.

Why run Batch with Java on Z?

Extend Development Skill Sets & Programming Resources

- Java programming skills are more prevalent.
- Leverage your OLTP infrastructure.
- Modern development tools increase application agility.

Integration with other Technology Solutions

- Extend Enterprise Scheduling with WebSphere Batch
- Java rules execution engines (Ilog, WODM, Drools)

Leverage Specialty Engine Processors on System Z

- Offload Java work to System Z Application Assist Processors (zAAP)
- Lower the overall cost-profile of running batch on the mainframe.

Compress your Batch Window

- Run batch during Online

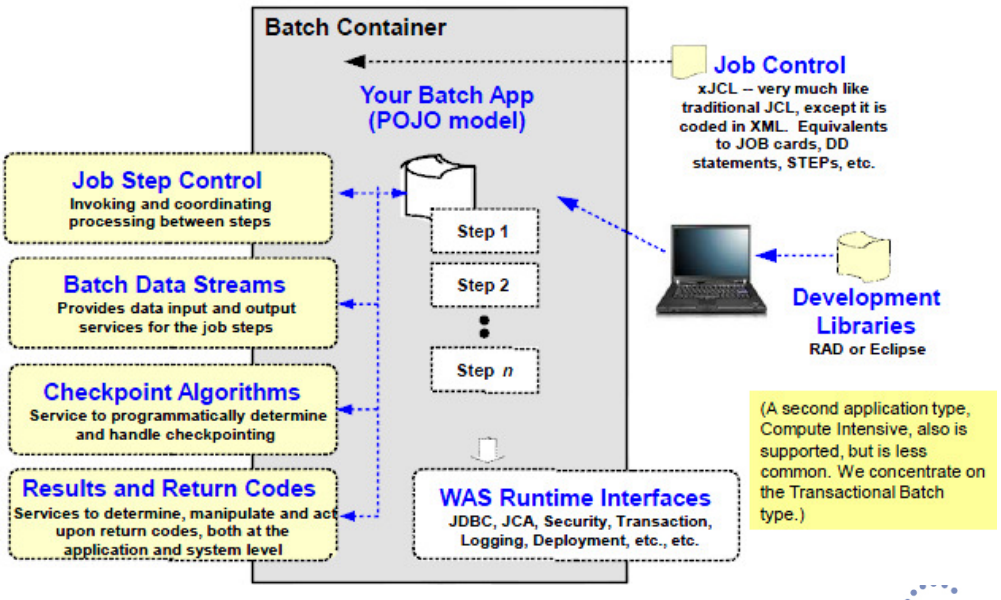
Why Use *WebSphere* Batch?

Benefits beyond a J2SE-based batch implementation:

- ***Long-running re-usable JVM*** to amortize out the cost of initialization and tear-down and reap the benefits of JIT
- Built on Java EE platform security, logging, integration, high availability, resource management.
- Programming model for batch in a structured, re-usable, rich manner.
- Application support functions specific to batch processing:
  - Checkpoint / Restart
  - Service Classification & Workload throttling
  - Log management and aggregation
  - Job Parallelism & Distribution across LPARs
  - Scheduler integration (TWS, CtrlM, Zeke, etc)

The Batch Programming Model

Functions & class libraries supplied with WebSphere Batch



Function-Rich Programming Framework

Batch Data Streams (BDS)

- Supplied "**Patterns**" for JDBC, JPA, J2C, Files, JZOS classes
- **BDS** maps data fields to Data Objects for Java
- Applications focus on Business Logic, Not data handling & Recovery

Checkpoint Processing

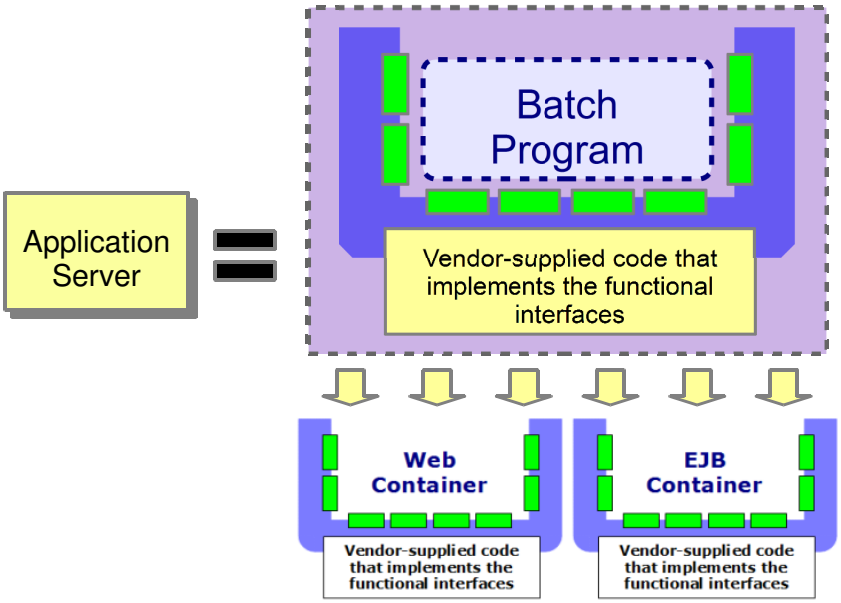
- **Interval** for Commit processing based on Time / Record numbers
- **Restart** failed jobs from Checkpoints

Extended Programming Functions

- **Skip-Record** Processing to tolerate data read/write errors
- **Retry-Step** Processing – Allow job to continue with errors (customizable)
- Configurable **Transaction Modes** (Local/Global at Step level)
- Batch Data Stream **Timeout** configurable at BDS level
- Record **Metrics** available through JobStepContext object
- **Parallel Job** Manager, Parallel Steps & Multi-threading or Multi-JVM
- **COBOL** Container – Share JDBC Type 2 Connectors

WebSphere Batch Runtime Container

The IBM Batch Container is an extension to the existing container structure of WebSphere Application Server

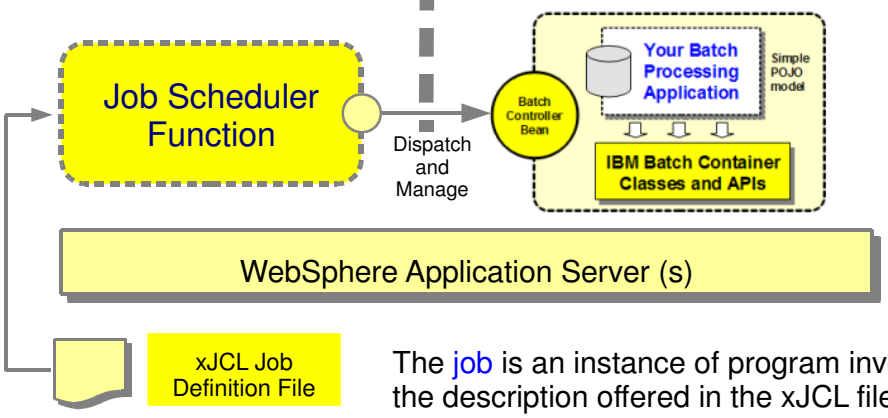


Separation of Job Scheduler & Batch Endpoint

The job scheduler gives you control to submit & manage jobs

Batch applications deployed like other WAS applications in the **Endpoint** server.

Stay started within the running JVM.



The job is an instance of program invocation based on the description offered in the xJCL file.

* The two may be in the same server, or separate servers. Or clustered. Your choice. ☺

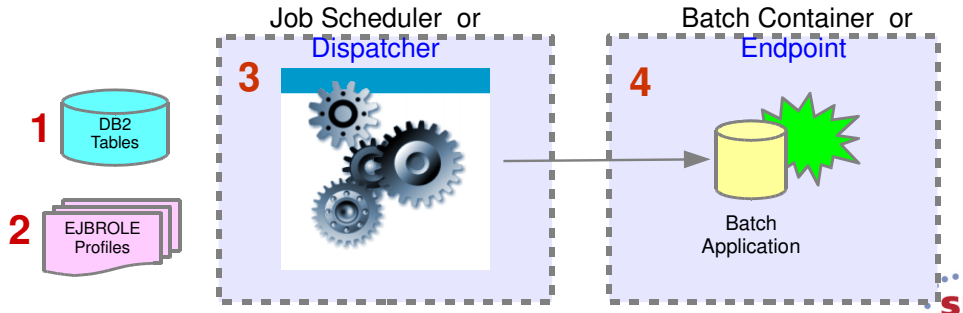
Simple Customization Steps For WebSphere Batch

Batch is Integrated in WAS V 8.5 profiles and "binaries"

(No longer "Augmented" as a "Stacked Product")

1. **Create Database tables & Data sources**
2. **Create Security (EJBROLE) Profiles**
3. **Configure the Job Scheduler in a Server (ISC)**
4. **Deploy a Sample Batch Application in an Endpoint Server (ISC)**
   - *Mapping to a server/cluster defines the Endpoint cluster.*

(See "***Installing and Configuring Compute Grid***" Techdoc WP101783.)



xJCL: Declaring the Structure of a Job

Concepts same as traditional // JCL ... syntax different

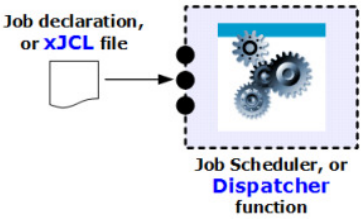
Analogous to the //JOB card

```
<?xml version="1.0" encoding="UTF-8" ?>
<job name="Sample" default-application-name="Sample" ... ">
<jndi-name>ejb/com/ibm/ws/batch/SampleBatchController</jndi-name>
      :
      <substitution-props>
         <prop name="ABC" value="1000" />
         <prop name="XYZ" value="/tmp/Sample.txt" />
      </substitution-props>

   <job-step name="SampleStep1">
   <jndi-name>ejb/SampleModule1</jndi-name>
      :
      <props>
         <prop name="first.name" value="${ABC}"/>
      </props>
   </job-step>

   <job-step name="SampleStep2">
   <jndi-name>ejb/SampleModule2</jndi-name>
      :
      <props>
         <prop name="number" value="${XYZ}"/>
      </props>
   </job-step>

</job>
```



Job STEP ... JNDI name analagous to // EXEC PGM= in traditional JCL

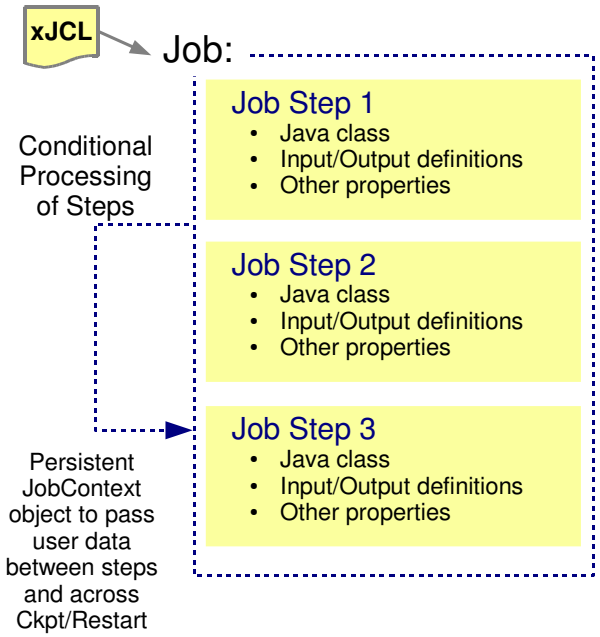
Carry substitution properties down to variable declarations in XML

Not shown:
- Input/Output declarations
- Checkpoint declaration
- Conditional processing
- Much more

xJCL "describes" the elements of a "job"

Batch Job and Job Steps

A batch job consists of one or more steps executed in order...



- Dispatcher interprets xJCL & determines which endpoint has batch application deployed
- Dispatcher passes job (xJCL) to Endpoint server
- Steps executed in order, with conditional step processing if declared
- Dispatcher maintains awareness of the job state
- When the job ends, Joblog accessible through JMC or WSGRID Proxy job
- Step-Retry; Skip-Records; Restartable

Ways to Submit Jobs

Several different interfaces to the job dispatcher:

Interfaces provided to submit jobs:

Browser The "Job Management Console" (JMC) is a simple-to-use browser application

USS Command Line Using the supplied lrcmd.sh client

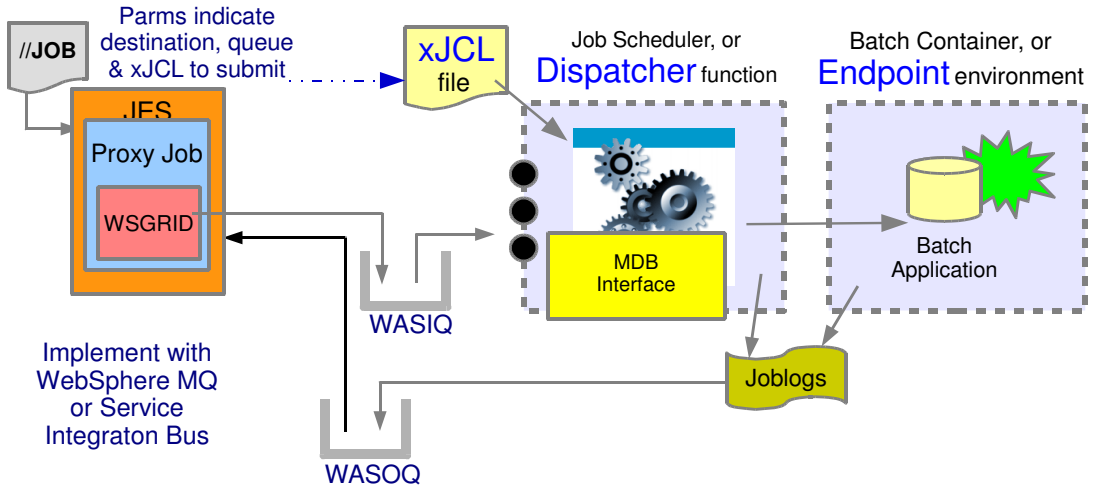
Web Services Web Services clients may use the JobScheduler interface

EJB Java programs may submit jobs using the JobScheduler EJB interface

MDB The Message Drive Bean interface allows job submission across a messaging queue. Heart of the integration with enterprise schedulers

Integration with Workload Schedulers

It's all about the MDB interface to the Dispatcher...

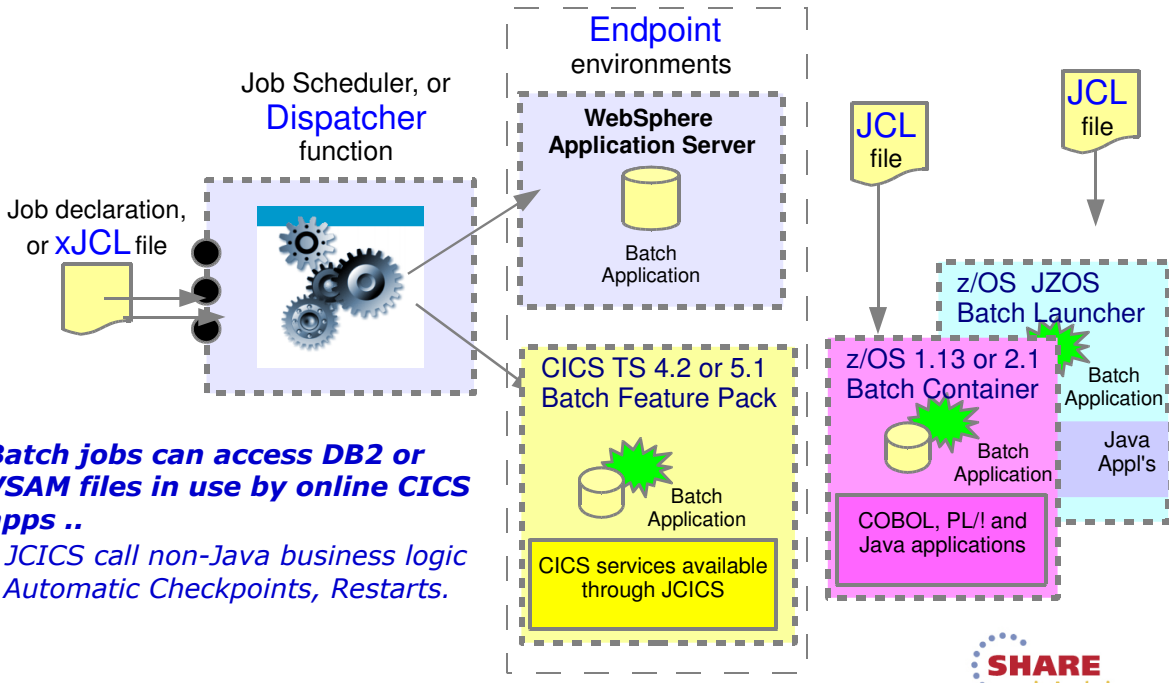


If workload scheduler is capable of submitting JCL or invoking a shell script, it can submit a job into Compute Grid

The WSGRID utility stays up for duration of job in Compute Grid, and feeds Java batch output to STDOUT or JES

"Batch Containers" not Limited to WebSphere

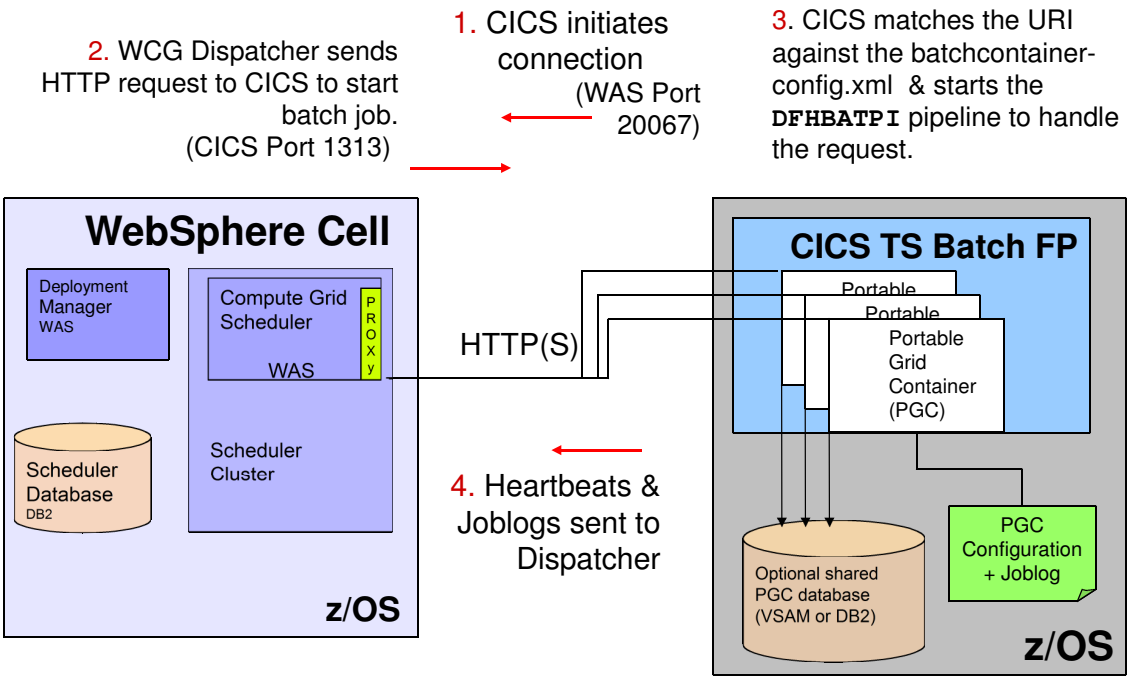
There's also a Java batch container for **CICS** ... and **z/OS**



Batch jobs can access DB2 or VSAM files in use by online CICS apps ..

- *JCICS call non-Java business logic*
- *Automatic Checkpoints, Restarts.*

WebSphere Batch and CICS with Modern Batch Feature Pack

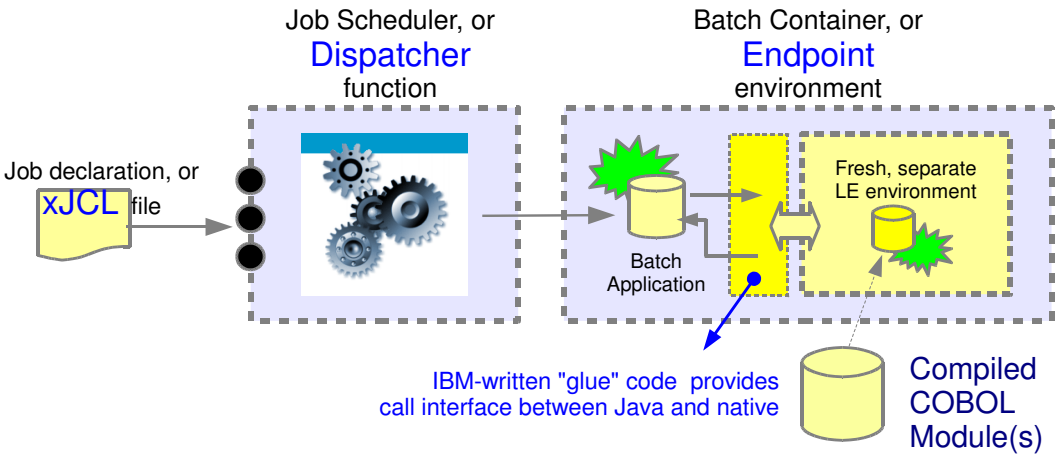


Configuring the CICS Modern Batch feature pack

- **Add resources in the CICS CSD (CICS System Definitions) - `DFHBCCSD`**
  - Use default CICS directory, HLQ, Port#, PIPELINE, DB2NAME
- **Define tables in DB2 - `DFHBCDB2`**
  - Define tables required for Batch Cont. (normal JOBLIB, Program, Runlib)
- **Configure JVM server profile - `BATCHPRF`**
  - Specify JAVA_HOME, WORK_DIR, LIBPATH, CLASSPATH & Java Options
- **Configure the batch container - `batchcontainer-config.xml`**
  - WAS Cell, Node, Server names; Hostname & HTTP Port
  - CICS Host Port, HLQ, HFS home
- **Start the Batch Container with these CICS Trans:**
  - CEDA INSTALL GROUP(DFHBATCH)
  - CBCH

Java Batch + COBOL? Yes ...

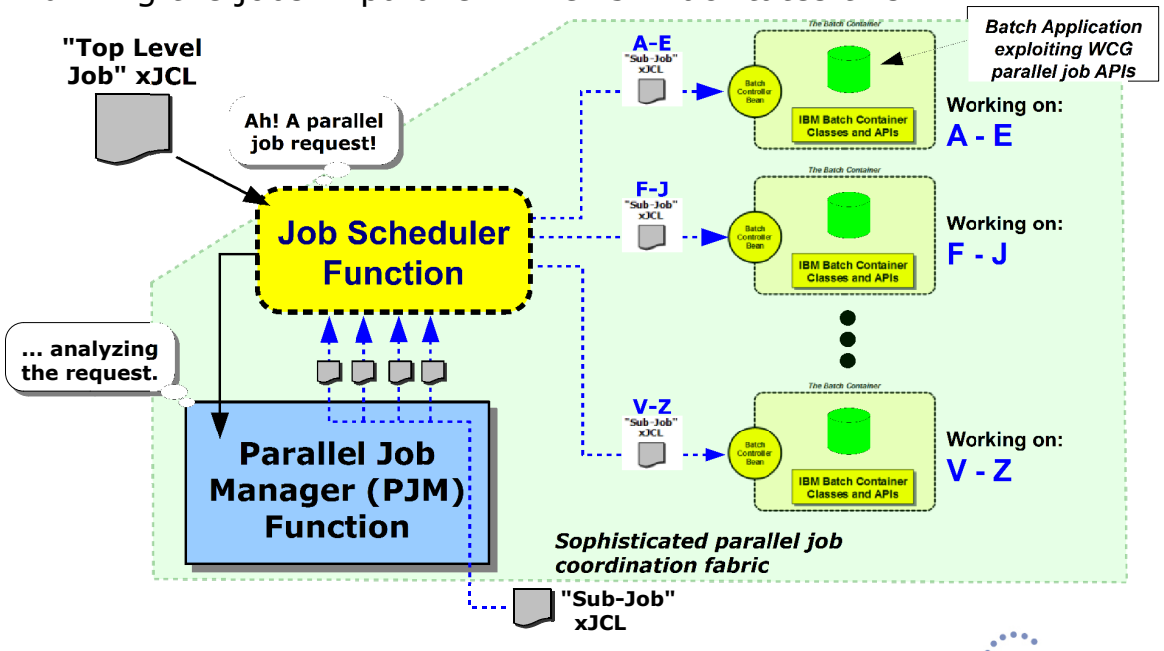
With a new COBOL interoperability function that allows Java batch programs to call COBOL programs directly ...



This provides a way to use (and re-use) COBOL assets as part of a Java batch job within Compute Grid

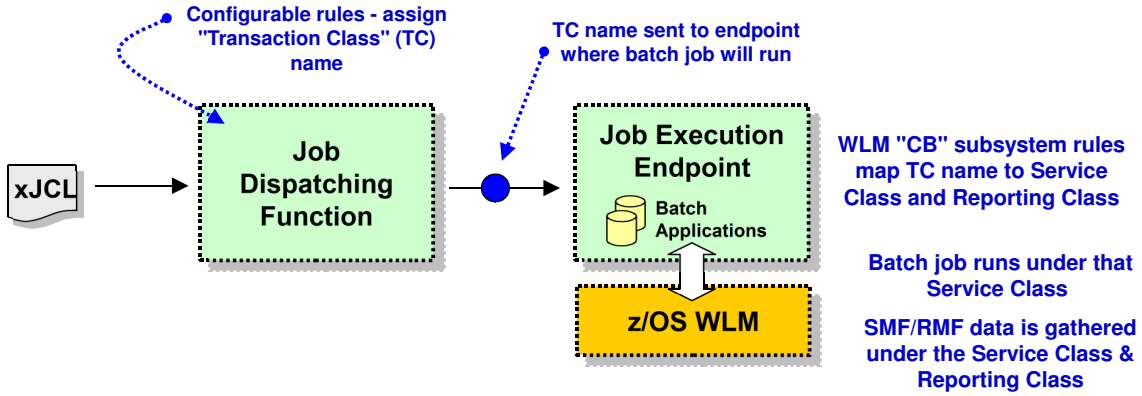
Parallel Job Manager (PJM)

Batch processing often lends itself to data partitioning and running the jobs in parallel. The PJM facilitates this.



Classifying Batch Jobs with WLM

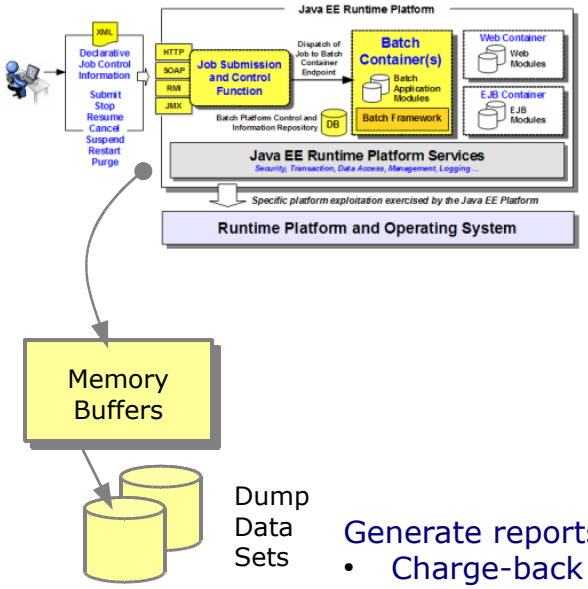
Submitted jobs can be tagged with a WLM "Transaction Class," which may be used to map the batch job to a WLM Service Class and Reporting Class:



Unique to z/OS!

SMF Recording for Charge-Back & Capacity Planning

SMF is a powerful (and fast) activity recording subsystem on z/OS.
Compute Grid z/OS exploits this with its own SMF record:



SMF 120.20 & 120.9 records contain additional info for Batch Jobs:

- Job identifier
- Job submitter
- Final Job state
- Server
- Node
- Accounting information
- Job start time
- Last update time
- General CPU usage
- zAAP or zIIP CPU use

Generate reports and determine usage for

- Charge-back Accounting
- Capacity Planning
- Performance Measurement.

Customer Use Profiles

Development Tools

Batch Programming Model

Framework & Classes

POC Scenarios

Customer Use Profiles - WAS Batch on z/OS

Large Financial Enterprise

Take advantage of newer development tools and skills
Increase agility -- faster time-to-market for changes.

Usage: Month-end, quarter-end & year-end batch jobs

- WebSphere Java batch spread across 6 LPARs
- WSGRID for integration with existing enterprise scheduler
- Parallel Job Manager for reduced overall completion time
- Heavy interaction with MVS datasets and DB2
- Integrated with WODM for business rules engine
- Co-located with batch in same WAS z/OS runtime servers

More Customer Use Profiles

International Insurance Enterprise

Exploit System z Specialty engines.

Usage: Lots of COBOL batch applications with many interdependencies

- Adopting Java batch incrementally based on batch job interdependencies
- COBOL Container lets Java call existing COBOL with low-level interface.
- Share JDBC T2 between Java and COBOL in same transaction scope
- Using WSGRID to integrate with enterprise scheduler.

Utility Company

Improve Critical Path in Batch Cycle.

Usage: Cut Bills for 1/20th of the Customers each night

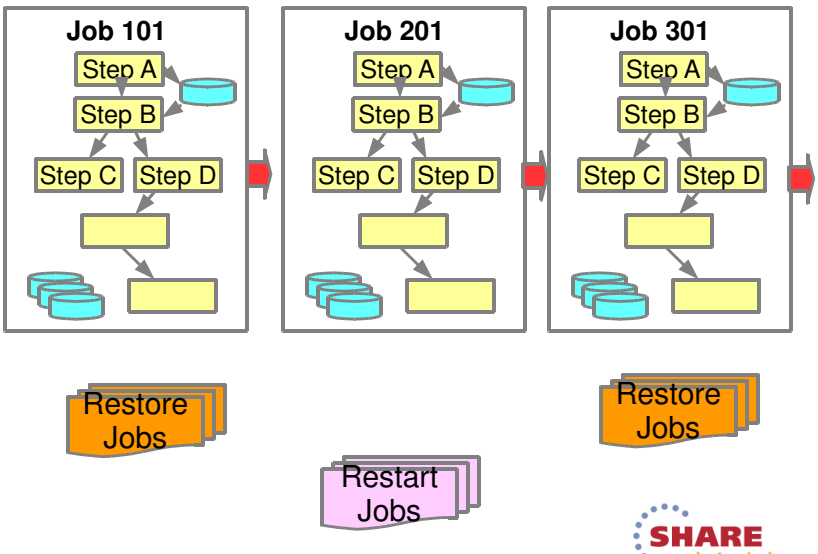
- Allow Batch jobs to be run while CICS online systems are up

  (Previously, online systems owning the database files must be offline so files could be accessed.)

The "Batch Cycle" (Network of batch job-streams)

- Strict Sequence with Restore, Restart & Recovery Jobs.
- Every Night, Weekly, Monthly, Quarterly, Year-End.
- Run when Online Applications are Down.
- Strict Time Schedule... Critical Path = "Batch Cycle"

```
//R71DB2RO JOB (R71DB2,NSBN),
//         'REPORTING ',MSGLEVEL=(1,1),CLASS=P,
//         REGION=3200K,MSGCLASS=X,NOTIFY=R71DB
//*********************************************
//*   Process  NATIONAL_BRANDRPT
//*********************************************
//JOBLIB   DD  DSN=DB2PRDN.DSNLOAD,DISP=SHR
//         DD  DSN=DB2PRDN.RUNLIB.LOAD,DISP=SHR
//*********************************************
//*        IEFBR14 - DELETE DATASETS
//*********************************************
//CLEANUP1 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1      DD  DSN=R71.REORGP.NSBN.SYSREC,DISP=
//         UNIT=SYSDA,SPACE=(TRK,(1,1))
//DD2      DD  DSN=R71.REORGP.NSBN.SYSUT1,DISP=
//         UNIT=SYSDA,SPACE=(TRK,(1,1))
//*********************************************
//*        UNLOAD DB2 FILE
//*********************************************
//GDG1     EXEC PGM=NBSDUNLD
//UNLD     DD  DSN=R71.NSBNUNLD.NSBN00TS(+1),
//         DISP=(NEW,CATLG,DELETE),
//         DCB=(SYSDSCB,RECFM=FB,LRECL=4096,BLK
//*********************************************
//*    -  APPLY NATIONLP.NSBN00TS UPDATES FIR
//*********************************************
//STPRPT   EXEC PGM=NBSDCUPD,SYSTEM=DSNP,
//         COND=(4,LT)
//SORTOUT  DD  UNIT=SYSDA,SPACE=(CYL,(999,99))
//SORTWK01 DD  UNIT=SYSDA,SPACE=(CYL,(999,99))
//SYSRECIN DD  DSN=R71.NSBDUNLD.NSBN00TS(+1),
//         DCB=(RECFM=V,BLKSIZE=16384),
//         UNIT=SYSDA,SPACE=(CYL,(999,99),RLSE)
//REC1     DD  DSN=R71.REORGP.NSBN.UPDATES,DISP=
//         DCB=(RECFM=FB,BLKSIZE=16384),
//         UNIT=SYSDA,SPACE=(CYL,(999,99),RLSE)
//*********************************************
//*    -  NATIONLP.NSBN00TS REPORT
//*********************************************
//STPRPT   EXEC PGM=NBSDCTBE,SYSTEM=DSNP,
//         COND=(4,LT)
//SORTOUT  DD  UNIT=SYSDA,SPACE=(CYL,(999,99))
//SYSRECIN DD  DSN=R71.REORGP.NSBN.UPDATES,DISP=
//         DCB=(RECFM=V,BLKSIZE=16384),
//         UNIT=SYSDA,SPACE=(CYL,(999,99),RLSE)
//RPTOUT   DD SYSOUT=*
//*********************************************
//*        IMAGE COPY NSBN00TS
//*********************************************
```



Function-Rich Programming Framework -
Education, including xJCL references & Sample code:

1) Job & Step Overview
   - Batch vs. Compute-Intensive
2) Implement the Batch Step
   - Create Java class
3) Batch Data Streams (xJCL)
4) Batch Loop & Checkpoints
5) Compute-Intensive Steps
6) Job Step Context
   - Object providing user data area
7) Setting/Using Step Return Codes
8) Job States
9) Exceptions & Failures
10) Batch Data Streams
11) Transaction Mode
12) Database Cursors
13) Batch Framework
14) Step Retry
15) Skip Record Processing
16) Application Packaging

References:

- www.ibm.com/developerworks/websphere/techjournal/0801_vignola/0801_vignola.html
- www.ibm.com/developerworks/websphere/techjournal/1109_alderman/1109_alderman.html
- WebSphere V8.5 InfoCenter

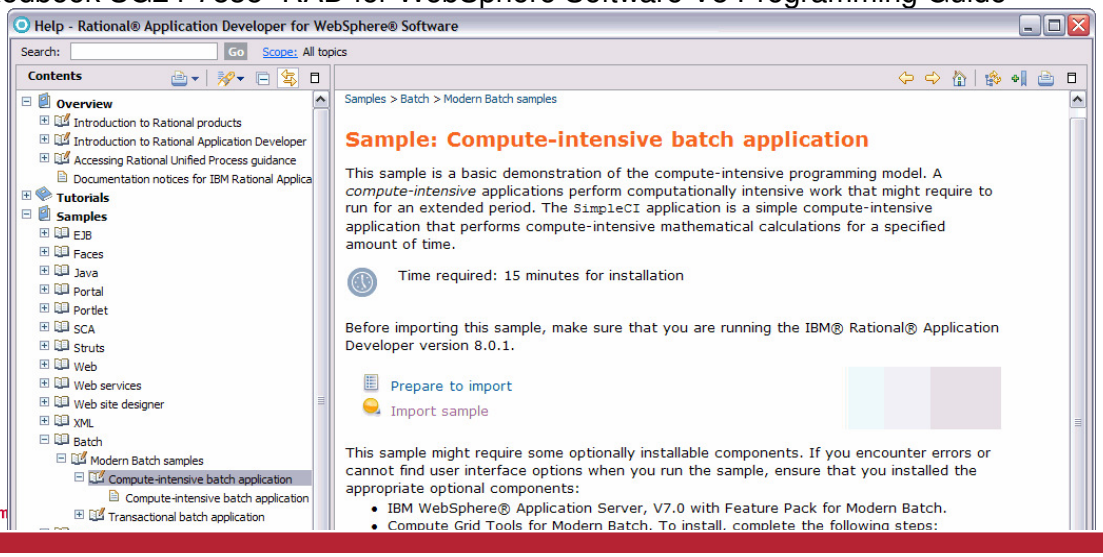
Some JZOS Classes to help Batch Apps:

JZOS has functions that make Java on z/OS much easier & more useful.
Use them in your batch application development:

- **DfSort** - Invoke DFSORT to perform high-volume sort and merge operations
- **Exec** - Run external process that buffers output, provides timeout control and stdout/stderr character encoding.
- **File Factory** - Build a BufferedReader, BufferedWriter, InputStream, or OutputStream on a text file or MVS dataset.
- **JzosPermission** - Simple Permission class to allow JZOS to operate with a SecurityManager (such as RACF)
- **MvsConsole** - Class with static methods to interface with the MVS console.
- **WtoMessage** - Data object/bean for holding a WTO message and parameters.
- **MvsJobSubmitter** - Submit batch jobs to JES2 or JES3 from a Java program
- **PdsDirectory** - Opening a PDS directory and iterating over its members.
- **Zfile** - JNI Wrapper for z/OS C-Library IO routines.
- **Zutil** - Static interface to various z/OS native library calls - getCurrentJobId(), getCurrentUser(), getCpuTimeMicros()

Application Development Environments

- Rational Application Developer tooling for WebSphere Batch (**RAD**)
  - Wizard based dialogs for creating batch application artifacts
  - Designed around the “Batch Data Stream Framework API”
- Techdoc WP101788 “Beginners guide to developing Java Batch Apps”
- Redbook SG24-7835 “RAD for WebSphere Software V8 Programming Guide”



Other Appl'n Development Environments

Options for Developers to test Locally on workstations:

- WebSphere Application Server Unit Test Environment (UTE)
  - No charge: WebSphere Application Server Developer Tools and WebSphere Application Server for Developers

ibm.com/developerworks/downloads/ws/wasdevelopers/index.html

- Batch Simulator
  - Stand-alone J2SE batch simulator to unit-test batch jobs:

ibm.com/developerworks/websphere/techjournal/0801_vignola/0801_vignola.html

- Batch model simplifies “swapping” platform-dependent components
  - BDS implementation for flat-file for z/OS data-set inputs.

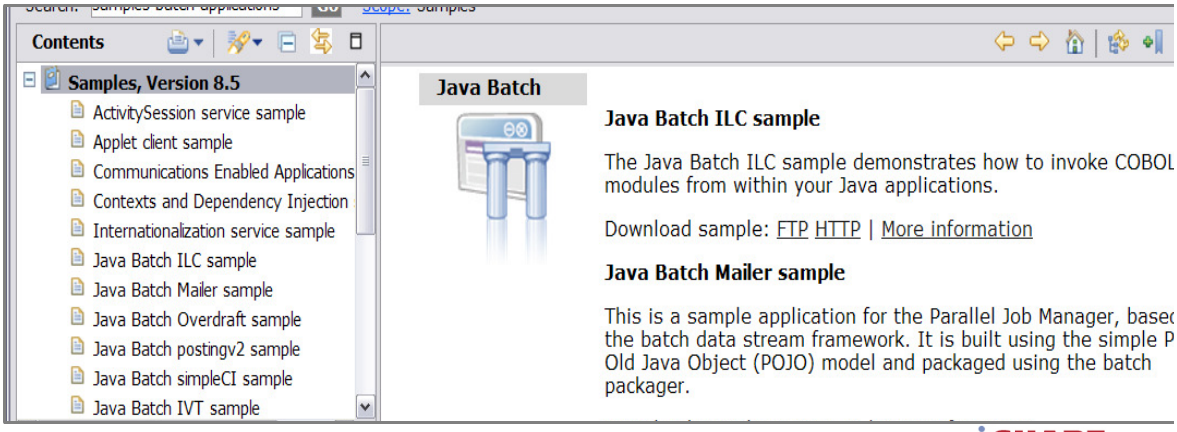
Rational Application Developer tooling for WebSphere Batch still the best!

WebSphere V. 8.5 Sample Applications

- Download samples from Samples, V 8.5 info. center:
  http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp
  - Samples no longer packaged with the WebSphere product.

On the Downloads tab, Download compressed file, or individual sample files.

- Samples have an /installableApps directory containing deployable prebuilt archives.
- Contain sample-specific source archives, scripts & instructions for building apps.



Application Selection ??

Choosing the right application for a POC

Workload Profile

- CPU-bound work can demonstrate capabilities of off-load eligibility
- IO-Bound applications still yield good results.

Application Dependencies

- Applications that interact with many other apps make poor POC candidates.
- Select an application that stands-alone to minimize the development effort
- Focus can be on developer tooling training, and performance comparisons.
- New Application or Existing COBOL application?

External System Integration

- WebSphere batch applications can integrate any external system you need.
- Choose an application that accesses DB2 or MVS data to demonstrate common integration patterns.

Success Criteria – Setting Expectations

Functional:

- Demonstrate equivalent functional capabilities
- MVS and DB2 integration
- Checkpoint / Restart capabilities

Operational:

- Integration with z/OS Job Scheduling (TWS, ZEKE, CA7, etc)
- Test Fail-over scenarios
- SMF 120.9 record generation & reporting

Measurement Criteria:

- CPU time
- Elapsed Time
- % Offload to specialty engines (zIIPs & zAAPs)
- Initial performance profiling

Transforming traditional JES job-streams

Keep Enterprise Scheduler as your Central Controller

- Replace individual job steps with WSGRID Batch Steps
- Keep traditional utilities as-is, or replace with Java / JZOS apps.
  (e.g., DFSORT can be invoked via JZOS services.)
- Use COBOL container to integrate Java-written functions while retaining existing COBOL code.
- Use CICS container to use CICS apps & access VSAM or DB2

Use Advanced Functions as you go . . .

- Parallel Job Manager can provide more horizontal parallelism.
- Automated Recovery and Restart processing
- Optimized Local Adapters (WOLA) for very efficient adapters to CICS, IMS, Batch, USS, ALC
- Parallel Sysplex, Reliability, Availability, and Fail-over scenarios.
- Performance tuning & measurement with WLM & SMF.

Where to go for more Help and Education?

DeveloperWorks **ibm.com/developerworks/websphere**

- Intro to batch programming - techjournal/0801_vignola/0801_vignola.html
- Skip-record processing - techjournal/1109_alderman/1109_alderman.html
- Simple Compute Grid Parallel Batch Application – tutorials/1203_usha
- Enterprise batch processing techjournal/1210_narain/1210_narain.html
- Integration w/ ent.schedulers techjournal/1303_narain/1303_narain.html

Techdocs - **ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs**

- WP101783 - WebSphere Modern Batch
- WP101788 - Beginners Guide to Coding Java Batch Jobs
- WP101909 - WebSphere Compute Grid COBOL Container
- WP102231 - WebSphere Compute Grid z/OS Capacity Planning
- PRS4644 - WebSphere Compute Grid for z/OS Wildfire class mat'ls
- PRS4467 - WebSphere ATS - YouTube Video Flyer with Hiperlinks

RedBooks www.redbooks.ibm.com

- SG24-7779 Batch Modernization on z/OS
- SG24-7343 Best Practices for Implementing WebSphere Extended Deployment
- REDP-4566 Batch Processing with WebSphere Compute Grid: Delivering Business Value

InfoCenter – http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp

More Help and Education (Cont'd)

Wildfire Workshop: **WebSphere Compute Grid for z/OS (WCG01)**

- Hands-on workshop available "no charge" to qualified customers. Learn how to configure and use WebSphere Compute Grid to develop and deploy batch programs in WebSphere Version 8.5.
- See Techdoc PRS1778 for more information.

IBM Education Assistant

- Collection of multimedia educational modules designed to help you gain a better understanding of WebSphere, Compute Grid and other IBM software products.

www.websphereusergroup.org/zos -

- Virtual WebSphere for z/OS User Group formed as a communication vehicle between customers, IBM technical support, and product development. Meetings are held via Web Conference and Phone

Summary

1. Quick Start with Simple Configuration in WAS 8.5.
2. Experiment with Sample Applications
   (IVT, Mailer, Posting).
3. Integrate with Enterprise Scheduler.
4. Create stand-alone Java applications,
   then inter-operate with COBOL and CICS applications.
5. Incorporate into your Batch Production Cycles.

WebSphere Application Server on System Z

Session	Title	Time	Room	Speaker
14618	Getting Started with WebSphere Liberty Profile on z/OS	Monday 9:30	Grand Ballroom Salon C	Loos/Follis
14692	Getting Started with WebSphere Compute Grid	Tuesday 9:30	Grand Ballroom Salon J	Hutchinson/Loos
14693	Using WebSphere Application Server Optimized Local Adapters (WOLA) to Migrate Your COBOL to zAAP-able Java	Wednesday 9:30	Grand Ballroom Salon K	David Follis
14620	WebSphere Liberty Profile on Windows AND z/OS (among other things) Hands-on Lab	Wednesday 1:30	Platinum Ballroom Salon 7	
14949	Tips Learned Implementing Websphere Application Server (WAS) on Linux for IBM System z	Wednesday 3:00	Grand Ballroom Salon G	Eberhard Pasch
14709	Need a Support Assistant? Check Out IBM's! (ISA)	Thursday 8:00	Grand Ballroom Salon A	Mike Stephen
15050	z/OSMF 2.1 Implementation and Configuration	Thursday 8:00	Grand Ballroom Salon G	Greg Daynes
14832	Web Apps using Liberty Profile Technology in CICS	Thursday 11:00	Platinum Ballroom Salon 2	Ian Mitchell
14722	Assimilating WebSphere Application Server into your z/OS WLM Configuration	Thursday 1:30	Orange County Salon 1	David Follis
15017	Using IBM WebSphere Application Server and IBM WebSphere MQ Together [z/OS & Distributed]	Thursday 3:00	Grand Ballroom Salon A	Ralph Bateman