Batch modernization on z/OS
An overview: Part 1

14 December 2011
GSE Belgium z/OS workgroup

De Smet Hendrik
IT architect zEnterprise
Trademarks

- This presentation contains trade-marked IBM products and technologies. Refer to the following Web site:

Agenda

- Batch – Do you still need it?
- Accommodating new functional requirements in z/OS Batch
- Implementing dynamic and agile Batch
- Reducing the Batch window
- Reducing Batch complexity
- Summary and Q&A
What is batch computing?

Batch Processing

Has been around for a very long time. Traditionally it's used to process large amounts of data in a repetitive way. But there's also "compute intensive" as well:

Both types are still very heavily used and under review as part of a broader "modernization" effort.
Is Batch still modern?

- Batch is still widely used and in many cases the best paradigm for data processing. Examples are:
  - generating reports of all daily processed data
  - paying salary for all employees
  - archiving of historical data at the end of month
  - reorganizing data
  - processing files with large amounts of data from a partner

- Advantages of Batch processing are:
  - A batch and an online window allow balanced system usage with almost constantly 100% utilization. This helps to optimize IT resources and save costs. In other words, you use the mainframe during the non-office hours for work that does not need immediate response, freeing up cycles during the office hours for transactions that need immediate response.
  - Batch programs are more efficient performing repetitive logic.
  - Postponing transactions can help to achieve more business security, e.g. intervening into money transfers that could damage your business.
45 years of Batch – From Punch Cards to Java Batch

- 45 years of evolution in Batch processing on the IBM Mainframe have provided the foundation for heavy-duty, reliable and efficient Batch for most large companies in the world:
  - WLM, WLM Batch initiators
  - Batch & Print Subsystem JES2, JES3, PSF
  - Job Control Language (JCL)
  - Batch Management Interfaces (for example, SDSF)
  - Step and Job dependencies by means of Condition Codes and Job Networks
  - Online and Batch in parallel
  - Time-driven Job execution
  - Job / Step Restart functions, Start, Submit, Remote submit, Syntax Scanner
  - Accounting based on Job/USER, Job statistics and RMF reports
  - Pre-loaded address spaces (initiators)
  - All Mainframe programming languages can be used in Batch
"Modernization"
A fancy term for the reality of constantly reviewing existing processes in light of the pressures facing you and your business objectives:

- Cost Pressures
- Competitive Pressures
- Time Pressures

Response
- Change
- Improve
- Update
- Modernize
- Work smarter

What is batch modernization
What is batch modernization

**Not a "Rip-and-Replace"**

We wish to emphasize that we're speaking of a reasoned evolution here, where the needs of the business are key:

- **Existing Batch Processes**
- **Java Batch Processes**
- **Enterprise Scheduling Solution**

Some are reengineered

Some are integrated with existing into a broader process

Some are net-new
IBM Batch Modernization initiative

- Ensure that IT architects and the industry recognize
  - Almost all large scale IT projects include batch
  - The web focus of the last decade has focused on online work, response time, transactions
  - Backend processing is often best implemented as batch

- Provide an IBM Batch solution for any platform
  - Including z/OS
  - Websphere Compute Grid
    - Both java programming model and execution environment

- CICS Compute Grid on z/OS

- Maintain leadership in z/OS batch
  - Improvements to existing function
  - Integrate and support the Compute grid programming model
Agenda

- Batch – Do you still need it?
- Accommodating new functional requirements in z/OS Batch
- Implementing dynamic and agile Batch
- Reducing the Batch window
- Reducing Batch complexity
- Summary and Q&A
The z/OS Batch Infrastructure is superior and unique in its kind, but did you ever try any of the following in your z/OS Batch Environment...?

- ... generating PDF files in COBOL, PL/I or ASM?

- ... creating *.xls or *.doc files in traditional Mainframe programming languages?

- ... generating diagrams or graphics?

- ... sending e-mails?

- ... using remote services via TCP/IP, HTTP or Web Services?

- ... processing XML?

New technologies need to be introduced to the z/OS Batch landscape to make all of the above examples possible.
Approaches

- Approach 1: Using new functionalities in traditional programming languages
- Approach 2: Executing Java Batch in traditional containers
- Approach 3: Java stand-alone Batch
- Approach 4A: WebSphere XD Compute Grid
- Approach 4B: WebSphere Application Server Feature Pack for Modern Batch
- Approach 5: WOLA
- Approach 6: z/OS Batch Runtime (new in z/OS 1.13)
- Approach 7: PHP with BPXBATCH
Approach 1
Using new Functionalities in traditional Programming Languages

- XML support
  - Using built-in XML support in COBOL
  - Using built-in XML support in PL/I
  - Using the XML Toolkit for z/OS (more complex XML processing in c/c++)
  - Using z/OS XML System Services (ZAAP, ZIIP enabled. Enterprise COBOL and PL/I, DB2 for z/OS, and the XML Toolkit for z/OS exploit the z/OS XML parser to reduce the application development and to increase performance.)
  - Solving the “XML problem” by combining XML technologies: read redbook!
  - Using pureXML capabilities in DB2 v9 for z/OS

- Interoperability between languages, c/c++ (using Inter Language Communication from LE)

- Unicode support
Approach 2
Executing Java Batch in traditional Containers: Introduction

The major application servers in z/OS support Java:

- CICS
- IMS
- DB2
- WebSphere Application Server
- WebSphere stack solutions running on WebSphere Application Server, such as
  - WebSphere Process Server, and WebSphere Portal
- Java stand-alone under z/OS
IBM uses a common code base for the JVM on all platforms plus additional platform specific extensions. One of those extensions on z/OS is specific Java APIs for batch processing, including:

- MVS data set and VSAM access to interact with z/OS specific data
- Condition code passing for integration of Java batch jobs into z/OS job nets
- z/OS Catalog search
- Interaction with the MVS Console
- Conversion of COBOL/ASM data types to Java types
- Invoking DFSORT to effectively sort data
- Access to z/OS Access Method Services (IDCAMS)
- RACF APIs to integrate Java into the z/OS security model
- Writing of Logstreams (for example as an Appender to LOG4J)
- Submission of Jobs from Java
Approach 2
Executing Java Batch in traditional Containers: DB2

- DB2 Java Stored Procedures
  - Can be called from any application
  - Transactional
  - JVM is persistent

Example: DB2 Java Stored Procedure
Approach 2
Executing Java Batch in traditional Containers: IMS

IMS Java Characteristics

- **Java Programs require a JVM as runtime**
  - JMP and JBP regions in IMS for Java main programs

- **Access IMS DB and/or DB2**
  - Through JDBC Type 2 Drivers
  - Unit of Work and Commit/Backout as in traditional languages
  - DLIModel generates Metadata for IMS DB Access
  - Java APIs for most IMS DL/I functionality is available

![Diagram of Java applications in IMS](image-url)
Approach 2
Executing Java Batch in traditional Containers: CICS

Batch in CICS? See later in this presentation
Approach 3
Java stand-alone Batch

- BPXBATCH or BPXBATSL (very limited functions)

- JZOS
  - Part of IBM Java SDK for z/OS
  - Integration in JES
  - Support for DD Statements
  - Development in Eclipse based tools possible
**JVM Launchers**

Instantiate a JVM and invoke the specified Java program within the JVM. On z/OS the model looks like this:

**Advantages:**
- Simple to use
- Included with z/OS

**Disadvantages:**
- Overhead of repeated instantiation
- Provides limited batch support functionality

JZOS represents a substantial improvement over BPXBATCH, but provides a relatively small set of batch functional support.
Approach 3
Java stand-alone Batch Example
Approach 4A
WebSphere XD Compute Grid

- Java Batch combined with all functionality provided by the WebSphere JEE Container

- Extension on top of WebSphere Application Server for z/OS

- Java stand-alone functionality plus
  - WAS Container management
    - security, transactions and connection management
  - Check pointing
  - Persistent JVM
  - QoS, such as high availability
  - Exploiting WLM

- Reuse of OLTP code in WAS XD Compute Grid Batch Container
Java Batch Platform

Provide the framework and integrates with the underlying middleware and platform:

- Java EE Runtime Platform Services
  - Security, Transaction, Data Access, Management, Logging...

- Dispatch of Job to Batch Container Endpoint
- Specific platform exploitation exercised by the Java EE Platform

**Advantages:**
- More functionality
- Greater integration with platform services
- Creates managed container environment for Java batch

**Disadvantages**
- Requires a Java EE platform on which to operate and integrate
Approach 4A
WebSphere XD Compute Grid
Approach 4A
WebSphere XD Compute Grid

WebSphere XD Compute Grid environment supports today’s batch processing needs, including:

• 24x7 batch processing, where batch can be executed concurrently with online transaction processing (OLTP)

• Sharing business services across batch and OLTP, where a service can be executed in multiple execution environments without sacrificing efficiencies, such as bulk-data processing

• Parallel-processing and caching features, where large problems can be partitioned, governed, and processed in parallel across a collection of server resources while hiding the complexities of multi-threading and management

• Container-managed batch qualities-of-service, such as checkpoint algorithms, restart mechanisms, multi-threading, and threshold policies, so the developer can focus on business logic

• Use application design patterns for building agile applications, where object-oriented design and service-orientation allow emerging middleware technologies, such as persistence and caching, to be adopted easily.

• Take advantage of the qualities-of-service of WebSphere Application Server z/OS, such as security, thread-pooling, connection-pooling, scalability and z/OS integration.
Physical Deployment – Typical z/OS

TWS

Job Scheduler

PJM

Batch Container

WAS Cell

Online Applications

Jobs

Jobs

Jobs

Jobs
Parallel Job Manager

- Job Repository
- xJCL
- Job Scheduler
- MailerTopJob.xml
- MailerSubJobName
- Batch Container
- Batch App
- SubJob # 1
- SubJob # N
- Parameterizer SPI
- 50 states / 4 jobs
  - Returns Properties[4] Each with STATES_LIST=different states

WebSphere Application Server

- STATES_LIST=AZ,AL,HI,MN...
- STATES_LIST=VA,WA,TX...

Parallel Job Manager

© 2011 IBM Corporation
WebSphere COBOL Container

- Multiple concurrent jobs per servant
- Separate COBOL "container*" per job

* Note COBOL Containers are CEEPIPI LE child environments
Approach 4B
WebSphere Application Server Feature Pack for Modern Batch

What it is
- Java-batch programming model for development and deployment of batch applications

What it offers
- Reduce cost of infrastructure due to concurrent execution of batch and OLTP workloads using shared business logic on a shared infrastructure integrated with WebSphere Application Server
- Reduced operational cost due to integrated administration of OLTP applications and batch job
- High throughput and low resource consumption on z/OS for Java Batch when collocated with data subsystems

WebSphere Application Server Feature Pack for Modern Batch
Approach 4A and 4B – Features compared

WebSphere App Server

WebSphere Batch Feature Pack

- Batch Container
- Job Scheduler
- Batch Toolkit

WebSphere Compute Grid Product

- Parallel Job Manager
- Enterprise Connectors
- Advanced Operations Pack

- Batch Container
- Job Scheduler
- Batch Toolkit

WebSphere App Server
IBM's Java Batch Offerings -- Basic to Advanced

This illustrates how IBM is addressing the Java batch challenge:

- Pacing/Throttling Job Submission
- Parallel Job Processing
- WLM Classification
- Resource Usage Reporting
- Batch Application Quiesce/Update
- System Resource Management
- Job Schedules
- Job Control within JVM
- Record Oriented Stream Handling
- Job Scheduler Web, WS, IIOP
- Job Pause and Resume
- Job Console into JVM
- Checkpoint Processing
- Job Repository
- Job Start Mechanism
- Simple JVM launchers, such as Java CLI
- Conditional Multi-Step
- z/OS SYSOUT
- MVS Datasets and use of DD Cards
- z/OS Only
- Feature Pack for Modern Batch
- Common Batch Container
- WebSphere Compute Grid
- Approach 4A and 4B – Features compared

- JZOS enhancements to the BPXBATCH model
- JVM remains instantiated
- Job Start Mechanism
- Basic Data Access
- System Resource Management
- Job Console into JVM
- Job Start Mechanism
- Common Batch Container
- WebSphere Compute Grid
- Feature Pack for Modern Batch
- Conditional Multi-Step
- z/OS SYSOUT
- MVS Datasets and use of DD Cards
- z/OS Only
Architectural view of how CICS can be a Compute Grid Execution Environment

Jobs are dispatched from the Compute Grid to an Execution Environment running inside a CICS JVM SERVER resource.
How does CICS support for the WebSphere XD Compute Grid help?

- The Compute Grid provides an alternative approach to bulk processing.

- Instead of taking CICS resources offline, bulk processing is run alongside other processing inside CICS and resources are shared.

- The Compute Grid Execution Environment helps with
  - Locking of data
    - Updates are syncpointed in small batches (Checkpoints) to avoid locking contention.
  - Failure/Recovery
    - If a program fails, it can restart from the last Checkpoint
  - Performance
    - Compute Grid jobs can be split and run in parallel across multiple CICS regions
Approach 4A and 4B rational tool support

Available now in Rational v8.0.1

- New Project Type for WebSphere Batch application
- Project and Job Creation Wizards help guide user through batch programming model semantics
- Editors for xJCL Job Control
- Source code wizards for Java code skeleton
- Menu choices for
  - generating required additional files
  - deploying to unit test or remote Compute Grid runtime environment
  - submitting the batch job
- Job output log viewer

Samples / Help

Wizard-driven

xJCL Editor

Unit Test Support
Approach 5: WOLA

The Motivation Behind WOLA

It started out as a way to allow program access into WAS for high transaction rate batch programs. Other solutions existed, but they all had limitations:

Inbound to WAS?
As more and more solutions are built based on Java EE, there is a growing desire to access them by batch, CICS and IMS programs.

MQ or Web Services?
Both are very good technologies and have their role. But for very high throughput and low overhead, each has their drawbacks.

Something else was needed ... something very fast with as little overhead per exchange as possible.
Approach 5: WOLA

WebSphere Optimized Local Adapters – What is it?

- Providing a highly scalable transactional solution
- New cross-memory communication structure for WAS V7, extension of a WAS “Local Comm” used inside of WAS.
  - Introduced in May of 2009 in WAS 7.0.0.4, and enhanced in WAS 7.0.0.12...

This extension is implemented with a new set of modules that provide an API for programs in external address spaces to access servers using this Daemon shared space mechanism.
## Approach 5: WOLA

### Side-by-Side Comparison

<table>
<thead>
<tr>
<th></th>
<th>WOLA</th>
<th>COBOL Container</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>High-Level Description</strong></td>
<td>Cross-memory byte array message exchange</td>
<td>COBOL module load and invoke</td>
</tr>
<tr>
<td><strong>Intended Purpose</strong></td>
<td>Provide a high-speed, low-latency exchange mechanism between WAS z/OS and external address spaces</td>
<td>Provide a means of utilizing COBOL assets within the context of a Java batch application in Compute Grid</td>
</tr>
<tr>
<td><strong>Applicability</strong></td>
<td>CICS, IMS, Batch, USS, ALCS; C/C++, COBOL, PL/I, High Level Assembler</td>
<td>COBOL invoked within servant region address space of WCG endpoint</td>
</tr>
<tr>
<td><strong>Programming Model</strong></td>
<td>Set of APIs used by external address space to register and exchange</td>
<td>Load of compiled module with direct invocation</td>
</tr>
<tr>
<td><strong>Security and Transaction</strong></td>
<td>Can propagate TX and identity with limits; see WP101490 for specifics on security and TX support</td>
<td>COBOL run under WAS thread identity; shared JDBC T2 may participate in global transaction initiated by Java Batch program</td>
</tr>
<tr>
<td><strong>Server Bit-Mode</strong></td>
<td>64-bit</td>
<td>31-bit</td>
</tr>
</tbody>
</table>
Approach 6: z/OS Batch Runtime (new in z/OS 1.13)

- A new option for running batch work in z/OS 1.13
- Provides a managed environment for integration of Java and COBOL
- Consistent with IBM Websphere based batch
  - A subset of the Websphere programming model
  - Incorporated in the OS
- DB2 resource manager
Batch Execution Runtime Environment
Java COBOL with DB2 Interoperability

- Ability to replace/add functions in current 3GL DB2 (e.g. COBOL DB2) application inventory with new Java DB2 code
  - Requires local attach z/OS DB2 connection sharing for common DB2 access
  - Requires UOW (Transactional) integrity among the application components

- A generalized solution without requiring a specific run-time or middleware, i.e. a pure batch environment

- Implementation requires little or no changes to existing code!
  - Only requires special callbacks for commit/rollback
Our z/OS Topology
JES Single Step based

Submit JCL

JES

Initiator

//STEP EXEC PROC=BCDBATCH
//MAINPROG DD *
PGMNAME

//PARMS DD *
TYPE=JAVA / COBOL
PARM = 'Parm String '
ARG1=
ARG2=
...

ARGIN=
/*

JCL is familiar to operations
code in the batch container is the same as the code one could write anywhere
Approach 7
PHP with BPXBATCH

- Scripting language
- Port for z/OS with full support available
- DB2 z/OS integration
- Straight-forward development in Eclipse
- Many skills available
- Many applications available based on open source
Approach 7
PHP with BPXBATCH

Figure 9-4 Stand-alone PHP batch sample overview
Agenda

- Batch – Do you still need it?
- Accommodating new functional requirements in z/OS Batch
- Implementing dynamic and agile Batch
- Reducing the Batch window
- Reducing Batch complexity
- Summary and Q&A
Questions

http://www.redbooks.ibm.com/
More information on zEnterprise

Thank You

Merci

Bedankt

Obrigado

Danke

多謝

감사합니다