IBM Modern Java Batch Solutions

From Basic to Advanced

IBM Advanced Technical Skills
Gaithersburg, MD

Batch Processing

Batch processing has been with us since the very beginning of electronic computing. It is alive and well: the need to process large quantities of information in a cost-effective and efficient manner is a cornerstone of information processing.

Java Batch?

Yes, Java batch. Java is a programming language growing in popularity for batch use. We see several factors behind the compelling trend toward Java as and batch processing:

24 x 7	Batch windows are shrinking and there's a business need to intermix batch with online. It is imperative to have capabilities to balance resource priorities
Cost	On z/OS, with zAAP processors the cost profile of executing batch processing in Java and offloading to zAAP engines is attractive and compelling.
SOA	An intriguing concept of "batch as a service" is emerging, and Java batch front-ended with web services is gaining ground.
Skills	Many companies seek to leverage their in-house Java skills across a broader set of business requirements, including batch processing.

Approaches to Java Batch

There are two broad approaches we see used to engineer batch processing with Java:

Custom Construction	JVM launch utilities, programming frameworks and custom code that ties it together.
Integrated Managed Environment	Software that integrates batch processing into a full function Java EE execution platform. This provides a superset of the first two: a JVM environment and a development framework, as well as a ***managed container environment*** for batch applications.

The Custom Middleware Trap

One of the things to be aware of and to avoid if possible is the temptation to repeatedly extend custom functionality to meet tactical needs. Over time it becomes a suite of custom middleware, which is expensive to maintain and draws resources away from the core business objectives.

Many of the simpler Java batch utilities and frameworks provide an environment ripe with temptation to develop more and more custom extensions and features. The entry cost is low; the long term costs very high.

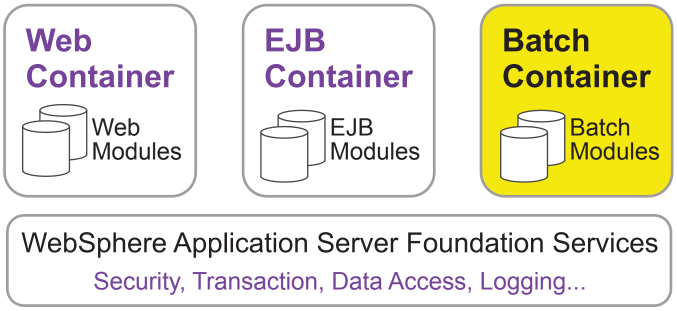
WebSphere Application Server

WebSphere Application Server is IBM's flagship Java EE runtime environment. It provides a rich set of application services within a managed container environment.

IBM offers WebSphere Application Server across hardware and operating system platforms ranging from smaller x86 all the way up to the mainframe running z/OS. In all cases the open standard support is consistent. Platform exploitation is done below that line. On z/OS the platform exploitation is extensive.

Java Batch and WebSphere Compute Grid

WebSphere Application Server is a powerful Java execution platform. It provides a strong foundation for a managed Java batch environment running within it. This is done by extending it to include a managed batch container:



Web Container
Web Modules

EJB Container
EJB Modules

Batch Container
Batch Modules

WebSphere Application Server Foundation Services
Security, Transaction, Data Access, Logging...

IBM's Java batch offering extends the WebSphere Application Server to provide a managed batch execution container environment.

This IBM middleware is fully intended to work in close cooperation with the WebSphere Application Server foundation.

> *"It is a lot easier to manage our batch environment in Compute Grid. Now one application can't take all the resources on the box, and adding capacity is as simple as adding another cluster member or node when necessary."* - Greg Boeschen, IS Java Infrastructure, Principal Financial Group

The Benefits of a Managed batch platform

The goal is the *meaningful integration of online processing and batch within an architected framework across multiple platforms with the ability to manage to business goals.*

The benefits include:

- Better utilization of compute resources
- Greater re-use of business logic
- More effective management of service level agreements
- Avoidance of isolated islands of custom batch middleware

It requires proper consideration of the overall enterprise architecture to fully realize these benefits.

The good news is you can start out simple and incrementally achieve these objectives in phases. IBM's Java batch offerings are designed to support such an evolutionary growth path.

Batch Offerings Built on WebSphere

IBM has two Java batch offerings:

- IBM WebSphere Compute Grid
- WebSphere Application Server Feature Pack for Modern Batch

The Feature Pack is a subset of WebSphere XD Compute Grid, offering key, initial enterprise Java batch functions at no-charge.

Both are available across all the platforms supported by WebSphere Application Server.

> *"WebSphere Compute Grid and WebSphere application Server provided us the framework to move towards a shared-service infrastructure for Batch and OLTP components. This approach improved time to market of our deliverables by leveraging modern design patterns, tooling, open source libraries and container-managed services as well alleviated our concern about diminishing availability of COBOL skills"* - Reto Estermann, Director and Head of Finance Repository Solutions, Swiss Reinsurance

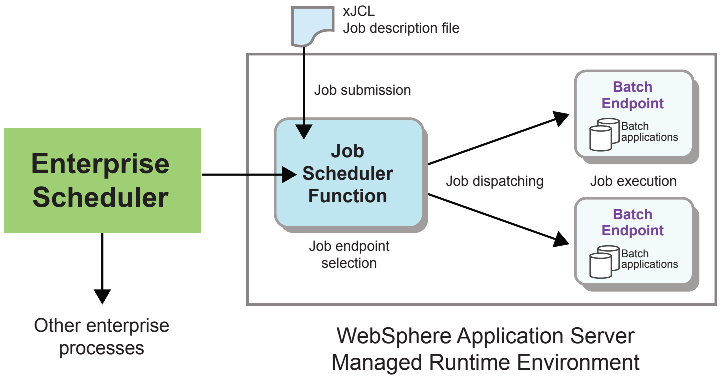
IBM WebSphere Compute Grid Features

Calendar and clock scheduling of jobs

Using database repository of xJCL job control files

Integration with enteprise schedulers

Capability designed into the product and available out of the box.



Usage reporting with SMF 120.20 records

(WAS z/OS only) This allows for accurate and timely accounting statistics for management and chargeback.

WLM transaction classification by job

(WAS z/OS only) Takes advantage of the classification capabilities of WAS z/OS to better manage mixed batch and OLTP to business goals.

Application quiesce and update

To enhance manageability and availability.

Job pacing and job execution throttling

Provides control to mix batch with OLTP within common Java execution environment and manage to batch execution completion goals.

Parallel job management and dispatching

Batch programs are often by their nature able to exploit parallel processing. Compute Grid provides tools and functions to program and split batch processing to run in parallel for faster processing.

> *“IBM has recognized lack of standardized programming model, development tools and infrastructure for JEE batch processing as a fundamental gap and have addressed that requirement through the WebSphere Compute Grid product.”* - Clyde Oldham, Architect: Java & WebSphere Platform, Absa Group

Compute Grid includes all the Batch features described below in the new Modern Batch Feature Pack

Feature Pack for Modern Batch

A "Feature Pack" is an optional function you may add to an existing WebSphere Application runtime environment. Feature packs provide a way to tailor your WAS runtime to have just the function your business needs.

The WebSphere Application Server V7 Feature pack for Modern Batch is available at no charge to customers with current Software Subscription and Support.

It provides:

Batch container environment

Provides the structure and support function your Java batch application uses

Development class libraries

Required for batch applications to run within this batch container

Batch Data Stream (BDS)

Function that abstracts data into an easily accessible record format, taking the stream handling burden off the developer

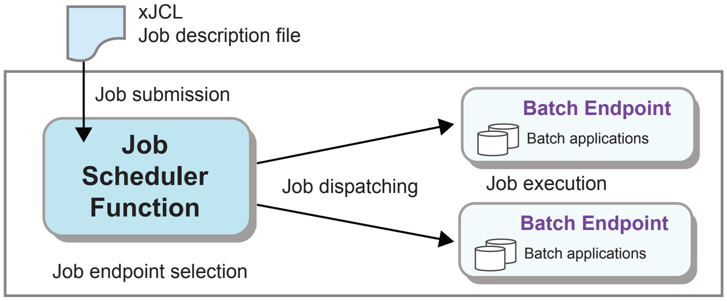
The next two items reveal a key element of the IBM Java batch story: the separation of the batch application (the code) from the batch job (the execution of that code). That's what allows more efficient re-use. It more closely mimics traditional mainframe JES batch processing.

Job scheduler and dispatcher function

Controls the dispatching of the submitted job to a batch container hosting the application

Declarative job control file (xJCL)

An XML file that describes the Java class files that are used in a batch step, and the steps that are included in the batch job.



WebSphere Application Server
Managed Runtime Environment

Finally, the Feature Pack offers a three more key features to enable efficient use of your batch applications running with the WebSphere Application Server environment:

Conditional multi-step job support

A combination of declaration in xJCL with step execution control

Transactional Checkpoint Processing

Record-based or time-based, specified at the job step level.

Results and return code coordination

A consistent results framework within the batch container for each job

In short, WebSphere Compute Grid is an enterprise strength Java batch execution platform while the Feature Pack for Modern Batch provides a subset of the Compute Grid functionality.

Summarizing Points

- Java batch is a rapidly growing option for batch
- Not all Java batch solutions are the same
- Beware of the "custom middleware trap" ... extending simple Java utilities and frameworks to include more and more tactical functionality
- The objective is integration of batch and online processing across platforms with the ability to manage to business goals
- The Feature Pack for Modern Batch is a Java batch platform function that exploits WebSphere Application Server as a foundation for a managed batch container environment
- WebSphere Compute Grid is a robust, enterprise strength Java Batch solution while the Feature Pack provides a subset of Compute Grid's function to get you started with Batch
- On z/OS you get the benefit of collocation with batch data commonly maintained on the mainframe data subsystems
- WebSphere Compute Grid for Hz/OS exploits additional functions of the z/OS platform, such as WLM, RRS and SMF.

Please visit the Compute Grid website, or contact your local IBM sales representative or Raj Suryavanshi, IBM AIM Software Product Manager rajsurya@us.ibm.com

http://www.ibm.com/software/webservers/appserv/extend/computegrid/