Product capabilities

- JES2
  - MAS
  - System Affinities
  - Job Classes
  - Initiators
- WLM
  - Service Classes
  - Scheduling Environments
  - WLM managed initiators
- TWS for z/OS
  - Work Stations, destinations, trackers
  - WLM scheduling environment integration
Operational challenges

- Limiting the number of jobs on a z/OS system
  - JOBCLASS(x) XEQCOUNT=(MAXIMUM=N)
  - Starting with z/OS 1.8, you specify this count at the member level JOBCLASS(x),XEQMEMBER(sysname)=MAX=N.
- Forcing immediate initiation
  - $SJ command enables the operator to force immediate initiation of a specific WLM-managed job regardless of goals or the job's position on the job queue
- Stopping initiation
  - $P XEQ announces your intent to perform an orderly shutdown of JES2. JES-managed initiators cease selecting new work; WLM-managed initiators drain and then terminate

(c) Copyright IBM Corporation 2002
Operational challenges

- Prevent work from starting (e.g. to avoid starting long-running jobs when planning a system shutdown)
  - on a particular system
    - `$P XEQ` announces your intent to perform an orderly shutdown of JES2. JES-managed initiators cease selecting new work; WLM-managed initiators drain and then terminate
  - in a particular service class from starting on a particular system
    - define your long running jobs to use a special service class
    - use the `$P SRVCLASS` or `$T SRVCLASS`

Maintaining high availability in TWS for z/OS job submission to a sysplex

- **Using blank work station destination** makes Controller submit to JES2
  - But this can have negative performance impact on Controller?

- **Using multiple TWSz Work Stations** defined with tracker destination to statically distribute job submission across sysplex
  - But how do we handle system maintenance / system failures?
Operational challenges

- Dynamic, goal oriented distribution of 10,000-100,000 jobs daily to multiple z/OS systems in a sysplex
  - WLM managed initiations

TWS for z/OS; JES2 and WLM
WLM performance adjustment

What Happened?
Learning by observation
Goals
Is a change necessary?
Is a change possible?
Will the change have a positive effect?
Go for it!
Will the change hurt some other workload?

Done every 10 Secs
On every system

Meeting your batch deadlines
- Who is in control and when
- TWSz: Addressing TWSz job submission delays
- z/OS: Addressing job execution delays
- z/OS: Addressing initiator queueing delays

(c) Copyright IBM Corporation 2002
Who is in control and when

- TWSz is in control up to the point in time when job is submitted to z/OS
  - waiting for predecessors
  - waiting for resources or time
  - job selected for submission from ready list using
    - urgent flag, Latest Start Time, priority, ...

- z/OS is in control from "reader event" until job has ended
  - wait until job can be scheduled (WLM scheduling environments)
  - wait for initiators
  - execute = use resource and wait for resources

Addressing TWSz job submission delays

- To address TWSz job submission delays, planners can:
  - Remove unnecessary predecessors
  - Split multistep jobs to reduce delays due to special resources
  - Review application and operation deadlines to ensure that OPC decisions based on Latest Start Times are correct
  - Improve deadline calculations by moving to TWS 8.1+ which allows job durations in seconds
  - Consider raising no. of allowed jobs in Started status, by adjusting // Servers and or special resources
    - This of course assumes that the z/OS system(s) can sustain the batch workload increase.
  - Review WLM Importance and Goal controls to make sure any workload increase does not adversely affect performance and throughput of other types of work.
Addressing job execution delays

To address job execution delays, TWSz can exploit the services of WLM, which controls the amount of systems resource available to each work unit, as follows:

- TWSz detects that a critical job is running late.
- TWSz calls a documented WLM interface to move the job to a higher-performance service class.
- The job receives more system resources and completes in a quicker time.
- The delay in the batch workload is therefore reduced.

TWSz job promotion

- AD database - Operation Detail - Automatic Options
  
  - CRITICAL Y/N (default: N)
    - If you specify Y in the CRITICAL field, TWSz automatically sends a request to the Workload Manager (WLM) to promote a job or started task in the high-performance service class, appropriately defined for batch jobs in the WLM environment, whenever the conditions of the specified assist policy are reached (when running job on Current Plan)
  
  - POLICY (default: ' ')
    - L Long duration. The job is assisted if it runs beyond its estimated duration time.
    - D Deadline. The job is assisted if it has not finished when its deadline time is reached.
    - S Latest start time. The job is assisted if it is submitted after the latest start time.
    - C Conditional. An algorithm calculates whether to apply the Deadline or the Latest start time policy.
    - " Default. WLM uses the policy specified in OPCOPTS.
TWSz job delay issues

**JOBA**
- Planned Duration: 10 minutes
- Latest Start Time: "something"
- Deadline: 06:00

**JOBB**
- Planned Duration: 10 minutes
- Latest Start Time: 22:00
- Deadline: 23:00

**JOBC**
- Planned Duration: 10 minutes
- Latest Start Time: 22:00
- Deadline: 22:10

---

Decision table

<table>
<thead>
<tr>
<th>Assist Policy (Intervention Option)</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Duration                            | - Gentle smoothing
   - Low impact                           | - Possibility of intervention when not actually necessary
   - Limited gain                         |                                                   |
| Deadline                            | - Gentle smoothing
   - Low impact
   - 100% certainty that intervention is required | Limited gain                                     |                                                   |
| Latest Start Time                   | - High gain
   - High probability that intervention is required | High risk of overcompensating                     |                                                   |
| Conditional                         | - Best action taken for every situation
   - No decision necessary by you         | Tuning efforts to set the Threshold value         |                                                   |
Addressing job initiation delays

To address JES2 initiator queueing delays, installations can exploit WLM Initiator management as follows:

- Activate one or more JES2 job classes in MODE=WLM
- WLM detects that Service Class goal is not being met
- If reduction of initiator wait time will help service class achieve its goal, WLM starts one or more initiators somewhere in sysplex.
- Service class receives more system resources and completes in a quicker time.
- The delay in the batch workload is therefore reduced.

WLM initiator management

- Dynamic, goal-oriented management of number of batch jobs running
- batch job selection and placement
- using business importance of batch workloads

- Multisystem workload balancing
  - batch workload is dynamically distributed in sysplex based on goals, goal achievement and system resources availability and utilization

- Reduced operational demands

- Improved reporting
  - job response time and pre-execution job delays
WLM Initiator control

- WLM maintains pool of initiators
- Adjust number of initiators for a Service Class
  - Based on goal achievement, observed delays and MPL
  - Decrease, increase and placement of WLM initiators is controlled by WLM algorithms
  - Workload balancing has sysplex wide scope

WLM Batch Management in Action

1. READER
   ① CONVERSION
   ② Classify (assign Service Class)
   ③ Queue for execution
   ④ INITIATION
   ⑤ EXECUTION

JCL

2. Classify request

3. Job select request

4. Batch Workload Service Class

5. Add init

(c) Copyright IBM Corporation 2002
The End