

# A synergistic solution for Production Batch workload management

## ThruPut Manager<sup>®</sup> and Job Schedulers

*This white paper clarifies the roles of job schedulers and ThruPut Manager SE (Standard Edition) in a JES2 z/OS datacenter. As you will see, they complement one another rather than compete. They act at different times, based on different input and affect the datacenter workload differently. Virtually every ThruPut Manager installation has a conventional job scheduler. This paper will explain how many z/OS shops benefit from both.*

### ThruPut Manager and Job Schedulers in a WLM JES2 Environment

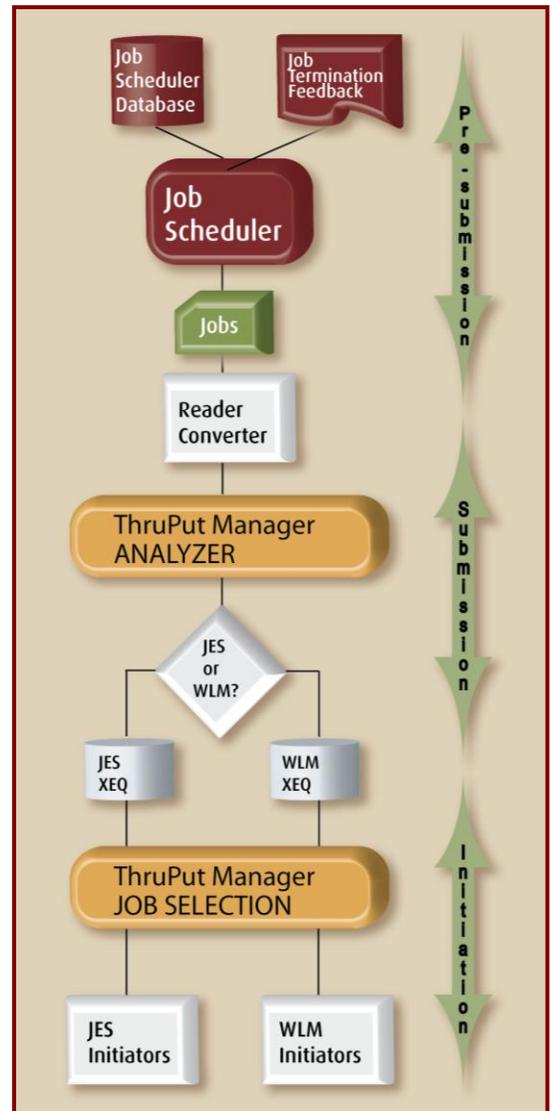
A Job Scheduler determines when and in what order jobs are submitted. ThruPut Manager optimizes the throughput of jobs while in the system. Its rules control the flow of the jobs through the submission and initiation processes, ensuring the availability of resources and directing jobs to the correct classes, queues and initiators.

ThruPut Manager analyses every job when it is submitted, builds a *job profile* containing over 250 *job descriptors* and applies data center-specified rules. Some of these rules can insert or change the Job Class and other job characteristics before the job is queued. Other rules affect the initiator, system or resources used during the Job Selection process.

### What ThruPut Manager does for Production Control

Using ThruPut Manager results in a batch window with more breathing room, "due-out" times which are met, less interference from non-production jobs, and simplified maintenance of your scheduler database.

Production Control uses a Job Scheduler to manage the web of interdependencies among all the production jobs.



### ThruPut Manager SE and Job Schedulers in a WLM JES2 environment:

*ThruPut Manager analyses every job submitted, builds a job profile with over 250 job descriptors and applies datacenter-specified rules. These rules can inspect and change Job Class and other job descriptors before the job is queued. Rules can also affect the initiator, system and resources used.*

8300 Woodbine Avenue  
4<sup>th</sup> Floor  
Markham ON Canada  
L3R 9Y7

905.940.9404  
905.940.5308 (fax)

marketing@mvssol.com  
www.mvssol.com

**MV'S**  
solutions inc.

**ThruPut  
Manager.**

Schedulers know the “due out” targets that need to be met and know when a job is ready or eligible to be submitted, relative to the other production jobs. They launch the jobs with the best intelligence available at that time.

This is where ThruPut Manager takes over. ThruPut Manager manages overall throughput of the entire batch workload, once the job is submitted, through the queuing process, until the correct initiator has been chosen. It acts as a traffic controller by

- Ensuring a job runs where the datacenter wants,
- Pre-loading tapes and other media,

- Pre-staging virtual volumes
- Executing HSM recalls as early as possible,
- Limiting the number of jobs accessing tape drives or other resources at the same time, and,
- Ensuring datasets aren’t allocated to less important jobs.

The time the job waits in the queue is minimized and the right initiator and system is chosen for execution, making all the job scheduler set up work pay off.

### Comparing Job Schedulers and ThruPut Manager SE

<i>Criteria</i>	<i>Job Schedulers</i>	<i>ThruPut Manager</i>
<b>Place in overall batch process</b>	<ul style="list-style-type: none"> <li>■ Pre-submission</li> <li>■ Post-execution feedback</li> </ul>	<ul style="list-style-type: none"> <li>■ Job submission to job initiation</li> <li>■ Optimizes execution by eliminating wait time</li> </ul>
<b>Target jobs</b>	<ul style="list-style-type: none"> <li>■ Production</li> </ul>	<ul style="list-style-type: none"> <li>■ Production</li> <li>■ Test</li> <li>■ Ad hoc</li> <li>■ Development</li> </ul>
<b>Rules maintenance effort</b>	<ul style="list-style-type: none"> <li>■ Rules are specified for each job, job-by-job, parameter-by-parameter</li> <li>■ Rules need updating whenever jobs change</li> </ul>	<ul style="list-style-type: none"> <li>■ Rules are specified for each datacenter standard</li> <li>■ Rules automatically apply to all jobs that are subject to that standard</li> <li>■ Rules only need updating when the standards change</li> </ul>
<b>Underlying Architecture</b>	<ul style="list-style-type: none"> <li>■ Looks like an ‘application’ to the operating system</li> </ul>	<ul style="list-style-type: none"> <li>■ Looks like a peer component of JES2 and WLM</li> </ul>
<b>Controls date/time dependencies</b>	<ul style="list-style-type: none"> <li>■ Yes</li> </ul>	<ul style="list-style-type: none"> <li>■ Provides facilities for non-production users</li> </ul>
<b>Controls predecessor dependencies</b>	<ul style="list-style-type: none"> <li>■ Yes</li> </ul>	<ul style="list-style-type: none"> <li>■ Provides facilities for non-production users.</li> </ul>
<b>Manages resource dependencies</b>	Sometimes and STATIC ... <ul style="list-style-type: none"> <li>■ Assumes no change for different submissions of the same job</li> <li>■ Dependent on correct information being entered in the job scheduler database</li> <li>■ Limited to resources held by production jobs</li> </ul>	Yes and DYNAMIC ... <ul style="list-style-type: none"> <li>■ Determined for every submission of every job</li> <li>■ Correct information is determined automatically at TM Analysis step</li> <li>■ Applies to resources used by total workload</li> </ul>
<b>Reassigns Job Class based on resources needed</b>	<ul style="list-style-type: none"> <li>■ No</li> </ul>	<ul style="list-style-type: none"> <li>■ Yes</li> </ul>
<b>Determines availability &amp; location of resources prior to job execution</b>	<ul style="list-style-type: none"> <li>■ Determined at submission time</li> </ul>	<ul style="list-style-type: none"> <li>■ Determined right before job execution</li> </ul>
<b>Books tape drives</b>	Yes, but ... <ul style="list-style-type: none"> <li>■ Only for production jobs</li> </ul>	Yes, and ... <ul style="list-style-type: none"> <li>■ Can see whole workload</li> </ul>
<b>Handles dataset contention</b>	<ul style="list-style-type: none"> <li>■ Limited, based on ‘negative dependency’ specifications</li> </ul>	<ul style="list-style-type: none"> <li>■ Yes, dynamically controlled, with no human specification required</li> </ul>

## Using Job Schedulers and ThruPut Manager Together

There may appear to be some functional overlap between schedulers and ThruPut Manager, but there are more differences than similarities. Using both products together gives the best results for overall batch management. When should you deploy each? That depends on what you are trying to accomplish.

### Use a Job Scheduler to manage job submission, for example if

- A job must run at a certain time
- A job can't run before another job

- A job must meet a "due out" time
- You need to build the batch schedule for a shift
- You need to handle cross platform dependencies

### Use ThruPut Manager to optimize throughput of jobs, for example if

- Classing and sysout standards must be enforced
- The number of jobs accessing the same resource needs to be limited
- Jobs must be run wherever a DBMS is currently available.
- Serialized HSM recalls delay job execution
- Dataset contention is affecting throughput.

### *A CASE STUDY: Datacenter Merger*

#### The Scenario

After an aggressive retirement incentive and an unexpected merger, this large utility's datacenter is reeling. The batch window is shrinking and budgets are frozen. The new datacenter director (both previous directors retired) has inherited two job schedulers and ThruPut Manager. He challenges the Production Control and Performance Managers to show him how these products can help. Together they take stock; their presentation to the new director includes these recommendations.

#### The Plan

##### 1. Choose one job scheduler and consolidate the two original databases.

Capture all the dependencies between jobs. Capture time and date, sequencing and cross-platform dependencies. Do not track resource dependencies here.

##### 2. Implement resource management by resource rather than by job.

This makes specification and maintenance of these rules much simpler. Use ThruPut Manager rules to implement this strategy.

##### 3. Map the Job Classes and SYSOUT to new standards.

Use ThruPut Manager rules to enforce the new standards without relying on the users to learn the new standards or to change their JCL.

##### 4. Consolidate usage of licensed software.

This reduces licensing costs. Use ThruPut Manager binding rules to make sure the original JCL can still run and the datacenter can be compliant.

##### 5. Consolidate DBMS deployment.

Use binding rules to ensure the original JCL can be directed to the most appropriate system.

##### 6. Avoid delays for HSM recalls,

Employ ThruPut Manager's HSM recall rules.

#### The Result

The datacenter director has re-established a reasonable batch window and can stay compliant with his newly consolidated software licenses. The staff is not burdened with the training or implementation impact of JCL changes.



This document assumes the reader is familiar with ThruPut Manager SE (Standard Edition). Only certain highlights of the product have been discussed here. For further information, please contact us as noted on the first page.

ThruPut Manager is a registered trademark of MVS Solutions Inc. The names of the optional components of ThruPut Manager used in this document are trademarks of MVS Solutions Inc. Other trademarks and registered trademarks used in this document are the property of their respective owners and are to be regarded as appearing with the appropriate ™ or © symbol.

© MVS Solutions Inc. 2014. All rights reserved.

N01

