Batch Automation and Job Scheduling – A New Collaboration

Session: 13142

Speaker: Mike Puig

mike@mvssol.com
MVS Solutions is a software company that focuses entirely on batch management in the world of z/OS and JES2

- ThruPut Manager has provided control of batch for 25 years
- TM AE (Automation Edition) provides service driven batch management using our automation engine
- TM AE capacity management allows sites to reduce their software licensing costs with IBM, in the context of sub-capacity pricing
- TM AE+ adds the Production Control Services component (PCS) which integrates our automation engine with CA Workload Automation CA 7 Edition to extend your scheduling goals into the z/OS execution phase

- ThruPut Manager SE has been around more than 25 years.
- ThruPut Manager AE was released about 5 years ago and is managing the batch in many sites today.
- Capacity management is a recent addition to ThruPut Manager AE and manages batch so as to allow the installation to lower their soft cap and reduce their MSU usage and therefore the monthly fee to IBM
- ThruPut Manager AE+ is the newest addition to the AE world. It communicates directly with CA 7 so that jobs can be managed in context of their application, its importance and its due-out requirements.
CA Validation

- MVS Solutions Inc. is CA Technologies Partner
- CA Validation for:
  ThruPut Manager AE+ with CA 7
- CA Validation Program
  - standard process for verifying the integrated solution
  - correctly installs, configures and performs
Agenda

- ThruPut Manager Evolution
- The vision
- The schedule
- Running the schedule
- Viewing the schedule
  - Application view
  - Job Instance view
- Dealing with problems
ThruPut Manager Evolution

- Without ThruPut Manager, all controls are manual and rely on everybody playing by the same rules.

- With ThruPut Manager SE, control is via a programming language (JAL)... IF MY-JOB THEN SET SERVICE_CLASS(VHIPRTY)

- With ThruPut Manager AE, control is via a goal based policy plus some JAL... TARGET=2 ACCEPTABLE=5

- With ThruPut Manager AE+, control is derived from the CA 7 schedule for production jobs and is the same as AE for ad-hoc jobs.
IF MY-JOB THEN SET SERVICE_CLASS(VHIPRTY)
40 years ago an operator would run 50 – 60 jobs per night and know every one. That was becoming impossible 25 years ago when the workload might be a few hundred to 2000 jobs. It is impossible today, when many installations run 20,000 to 100,000 jobs per day through their scheduler.

CA7 is itself an application, outside of z/OS and JES2, and has little or no influence on the processing once a job is submitted.

PCS provides the link between CA7 and the z/OS world so that the entire schedule is automated from schedule generation through to the end of the production cycle.

---

**The vision**

Batch processing has become too complex for users and operators to manage without automation

- Good job scheduling software such as CA 7 is required but does not provide a complete answer
- Automation within z/OS and JES2 is needed to ensure production applications meet the CA 7 schedule goals
- ThruPut Manager AE+ works hand in glove with CA 7 and z/OS & JES2 to deliver your schedule goals automatically and provide that complete answer
The schedule

- The scheduled applications are very simple
- An application is equivalent to a system name in CA 7
- A tree is all the jobs between a header job and all its trailers
  - An application can have multiple trees
- A path is all the jobs between a header job and a single trailer
- An application always has an urgent path – the path with the least amount of slack time
  - The urgent path often changes during processing

- This presentation is based on a small and simple CA7 database used for demos to illustrate what PCS is doing.
- Slack time is an important concept in PCS and is recalculated continually for each path as jobs end.
— This is one example of a simple application, with 1 tree and 2 paths.
— This application has three jobs that require exclusive use of the same dataset, leading to contention situations.
Of course there are setup tasks to be done for PCS. The first time only that PCS comes up, it asks CA7 for the list of system names and then the list of jobs in each system, which it uses to populate its Application Management Database or AMD.

- If necessary you can refresh the list of applications.
- In the AMD dialog you can set due-out times and Production Importance (PI) for any application.
- As jobs are run, PCS captures experience data at the step level, including the elapsed time, CPU usage, and number of tape drives actually used (including dynamic allocation).
- On a daily – or production cycle – basis, you run the batch job to capture the schedule data for that period.
- That data is combined with the AMD data and our experience data to build a “Battle Plan”, in which every application, tree and path in the schedule is calculated for its expected runtime and its ability to meet the due-out time.
PCS manages selection priority based on the “rerun factor” (how much time is there to rerun this job/path/tree in the event of a problem, expressed as a number, with 1 meaning the job/path/tree could be rerun once), the slack time and the Production Importance.

- Note slack time and rerun factor are dynamically recalculated as each job completes or fails to start when expected.
- Jobs are prepared to run while in the queue so that they can run immediately and without delays once selected.
- Tape drives are dynamically managed based on the needs of the job and the real availability of drives.
- Dataset contention is dynamically managed when a job is selected. A job may be held back if its datasets are not available, rather than letting it occupy an initiator. Jobs held back are released based on Importance.
- The WLM Service Class is assigned as the job goes into execution, from a set that is dedicated to ThruPut Manager.
This is the ‘dashboard’, the panel that gives a quick overview of your current status, and is always the place to start.

Note the term ‘Job Instance’. This refers to a combination of the jobname and schedule id, as in IAP2001_001.

Note also the term ‘Culprit’. This refers to jobs that are the cause of other jobs being late.

The panel shows how many applications, trees, paths, and job instances are late, are calculated as will be late and were late.

It shows the total number of entities in the schedule, how many are done and how many are still to be run.

It also shows the number of ‘Adhoc’ jobs. These are jobs submitted by CA 7, probably as a result of being DEMANDed in, that were not in the schedule. These jobs do not have a slack time and do not belong to a path or tree, since there is no information to identify where they fit. However they do have a system name so do belong to an application and will be managed with the importance of that application.
One of the choices is to view the schedule by application.

From this panel you can drill down to see the trees, paths and job instances in the application and their status.
You can also view by Job Instance, as in this panel.

This is the left view of a wide panel (I use 132 for my width under ISPF).

The panel contains a lot of information from PCS, CA 7, and z/OS. It provides the ability to get more information from CA 7, through use of its commands, and from ThruPut Manager through use of the I (Info) command.

Any non-dialog CA 7 command, such as LQ, LJOB, LRLOG, can be issued on the command line prefixed by forward slash-blank (’/ ’)
This is the same panel scrolled to the right. Notice the Job Instance id does not scroll.

The fields on the Job Instance panel are explained in the next few slides.
### Fields of Interest

- **Job Instance Id** – Job Name + schedule id
- **CP** – is it on the critical (most urgent) path of the application?
- **Slack Time** – used to calculate urgency
- **Job Instance Progress** - # jobs preceding or waiting on this job
- **PI** – Production Importance
- **Status** - Pending, Active, Done, Ready, Failed

<table>
<thead>
<tr>
<th>Job</th>
<th>App</th>
<th>C -Urgent Path-</th>
<th>Job Inst P</th>
</tr>
</thead>
<tbody>
<tr>
<td>--Instance Id--</td>
<td>--Name--</td>
<td>Rerun</td>
<td>Slack Progress</td>
</tr>
<tr>
<td>IAP2001_001</td>
<td>IAP</td>
<td>0.3</td>
<td>00:07</td>
</tr>
</tbody>
</table>
If the job is in execution, the Details column shows on which LPAR and in which step.

It also shows when a job has been cancelled or skipped in CA 7 and if its triggering job was cancelled or skipped.
This is the status according to CA 7
### Fields of Interest

- **CA 7 Status** (Continued)
  - R:#nnn – step level condition code fail
  - R:Cnnn – job level condition code fail
  - A:Snnn – system abend
  - A:U:nnn – user abend
  - R:JCLERR
  - Not Run
  - N:FAIL/STOP/OFFL/XTRK – for XPJOB
  - W:AGENT

<table>
<thead>
<tr>
<th>Job Instance Id</th>
<th>CA-7</th>
<th>CA-7 # of</th>
<th>Exp/Actual</th>
<th>Job-Elapsed</th>
<th>Step-Elapsed</th>
<th>Exp/Actual Ending-Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAP2001_001</td>
<td>FC1</td>
<td>2</td>
<td>2</td>
<td>00:00:00:00</td>
<td>00:00:00:00</td>
<td>Sep 30 15:17 00:32</td>
</tr>
</tbody>
</table>
### Fields of Interest

- CA-7 Job #
- # of Paths
- HTM - Header, Trailer or Middle job
- Expected or Actual Start and Ending Times
- Job Elapsed (and Step Elapsed) Expected and Actual
- Acceptable Time

<table>
<thead>
<tr>
<th>Job</th>
<th>CA-7</th>
<th>CA-7 # of H</th>
<th>Exp/Actual</th>
<th>Job-Elapsed</th>
<th>Step-Elapsed</th>
<th>Exp/Actual</th>
<th>Start</th>
<th>Start Time- Actual Ending-Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAP2001_001</td>
<td>FCAST</td>
<td>2</td>
<td>M</td>
<td>Sep 30 15:17 00:32</td>
<td>00:00 00:00</td>
<td>Sep 30 15:49</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Acceptable Time is the time a Job Instance is expected to be completed by, after which it will be considered to be late.
Let’s look more closely at this panel.

This job shows that it failed with a System abend, S106. An operator should know immediately that this is something that requires technical support or programming to resolve, and would therefore initiate a problem ticket and make a call.
— Here’s another job with an interesting status.

— It’s executing but is late – note the negative values for rerun factor and slack time

— It also shows that there are 11 jobs waiting on this and, by the C under Progress, that it is considered a culprit

— To find out what is happening we put a ‘G’ for proGress, on the line.
The Culprit Details panel shows us in this case that the job has been executing for longer than expected.

There are four commands that are often useful at this point, although not in this case, that can be invoked by choosing 1, 2, 3 or 4.

Even though it’s not really helpful in this situation let’s try 1 to illustrate what happens.
Here is the result of a LQ command issued from the previous panel

The results are displayed in a Browse window, meaning that it is scrollable up and down – very useful for LJOB, LIST=ALL,...
Dealing with problems

Command ==> View Active Battle Plan Job Instances

Cycle Date/Time - From: Sep 30, 2011 14:00 To: Oct 01, 2011 13:59
Current Date/Time: Sep 30, 2011 15:11:21

Line Commands: S-Job Paths P-Detailed Job Paths X-Display Experience
G-View Defaults/Options G-Progress I-Info(TM/UDF) A-Add to R-Remove from

List of All Job Instances in Battle Plan

<table>
<thead>
<tr>
<th>Job Instance Id:</th>
<th>App</th>
<th>Urgent Path</th>
<th>Job Inst P</th>
<th>Rerun Slack</th>
<th>Progress I</th>
<th>Status</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAP001_001</td>
<td>IAP</td>
<td>C</td>
<td>0.3</td>
<td>00:07</td>
<td>1</td>
<td>3</td>
<td>Pending</td>
</tr>
<tr>
<td>IAP001_001</td>
<td>ISI</td>
<td>C</td>
<td>0.1</td>
<td>00:02</td>
<td>1</td>
<td>4</td>
<td>Failed</td>
</tr>
<tr>
<td>IAP001_001</td>
<td>IAP</td>
<td>C</td>
<td>0.3</td>
<td>00:07</td>
<td>0</td>
<td>3</td>
<td>Active</td>
</tr>
<tr>
<td>IAP002_001</td>
<td>IBK</td>
<td>C</td>
<td>0.5</td>
<td>00:07</td>
<td>0</td>
<td>2</td>
<td>Ready</td>
</tr>
<tr>
<td>IAP001_001</td>
<td>IBK</td>
<td>C</td>
<td>0.4</td>
<td>00:06</td>
<td>0</td>
<td>2</td>
<td>Ready</td>
</tr>
<tr>
<td>IMK001_001</td>
<td>IMK</td>
<td></td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>Done</td>
</tr>
<tr>
<td>IMK001_001</td>
<td>IBK</td>
<td></td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>Done</td>
</tr>
<tr>
<td>IMK001_001</td>
<td>IBK</td>
<td></td>
<td>-0.3</td>
<td>-00:06</td>
<td>C</td>
<td>11</td>
<td>Active</td>
</tr>
<tr>
<td>IMK001_001</td>
<td>IMK</td>
<td></td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>Done</td>
</tr>
</tbody>
</table>

© 2013 MVS Solutions Inc.

- Back on the Job Instance display, let’s find out how this job usually runs
- By placing an X on the line command area we can see the experience data for this job and schedule id combination and determine how excessive its current run time is
The experience data shows that although the job usually runs for around 15 minutes it has recently run as long as 20 minutes, so perhaps 22 minutes is not yet something to worry about.

By the way, this experience data is what PCS uses to calculate the expected path lengths.
— So what would a good operator do? Not much for now
— Depending on how important and time-critical this job is, the operator will probably leave it to run for another 5 or 10 minutes before calling anyone
— However, if that call is necessary operations can report the usual runtime, this runtime, and the step the job is in
There is one more job of particular interest on this job instance panel, job IBK2001.

This job shows as Ready and Awaiting Execution.

Why is it not running? It’s clearly not a CA 7 situation so the cause is somewhere in the world of z/OS, JES2 and ThruPut Manager. There could be a number of reasons so let’s find out by using the I (Info) command.
Info pops up a window giving you the status as understood by ThruPut Manager.

It says ‘The Job was not selected because it was held by ThruPut Manager’. Why would that be?

In the top line of that pop-up it shows SLM in green and DC in red.

SLM stands for Service Level Manager, the automation engine that, under the covers, is getting the jobs selected and run according to the directions of PCS. Since it’s green it is not the reason the job is held back

DC stands for Dataset Contention and is the reason for the Hold

To find out more, put the cursor on DC and hit Enter
Another pop-up appears with Dataset Contention information

It shows that the job is waiting for one dataset, RSI01.GDEMO.HOLDDS SN, and needs it exclusive (DISP=OLD or DISP=MOD)

The next thing to find out is who or what is holding this dataset. Put the cursor on the dataset and hit Enter
And here’s the answer. The dataset in this case is in use by another batch job, IBK1001, which also has it exclusive.
So what would a good operator do about this job? Nothing, because ThruPut Manager is already doing what needs to be done if it is possible.

In this case the operator – and ThruPut Manager - must simply wait until the holding job releases the dataset.

If the dataset had been held by a TSO user the userid would have been shown.

The TSO user would have received nagging messages stating that the dataset was required by job IBK2001.

If the job was importance level 1 or 2, and the user did not have the dataset open for output or update, the user would have been warned that the dataset would be taken away. If after that the dataset was not freed it would be removed from the TSO session, a technique known as repossession.

Dataset fencing is a feature of ThruPut Manager and PCS intended to prevent a TSO user or a job outside of the schedule from enqueuing a critical dataset and causing this situation unnecessarily.
Behind the scenes, z/OS and JES2 are being managed to meet the due-out times and importance of the schedule

- If the queues are growing, jobs are moved ahead based on slack time and Production Importance
- If the system is very busy, low importance jobs may be delayed to ensure high importance work gets done
- If you use IBM’s soft capping with sub-capacity pricing, batch is managed to keep within your limits
- The Service Class is set appropriate to the Importance and urgency of each job

- Because of its connection to CA 7, PCS knows each job in the context of its application, tree and path and can calculate when it needs to complete and therefore when it needs to start
- It adds some padding — a Target time allowing for a rerun if possible — as well as the Acceptable time, and tells SLM the selection priority, the Production Importance and whether the job is late or on the most urgent path for the application
- SLM uses that information to manage the job in its PCS queue, move it as quickly as need be to meet its selection goal, prioritize the preparation and set the Service Class
- SLM is cognizant of soft capping and, with its capacity management capability, will slow down selection of batch according to your directions. For example, you might stop selecting your lowest importance batch when at 90% of your desired cap, stop selecting the next level at 92% and so on
The SDSF DA panel shows the current Service Class in use by each executing job.

This display shows that job IBK3002, a job from the same application and with the same Production importance, is running in a Service Class called SLMHIGH, whereas our late job, IBK1001, is running in Service Class SLMVHI (very high), as set by SLM.
PCS provides intelligent use of your system resources to meet the goals of your schedule

- It automatically adjusts the selection priority and Service Class for a job based on progress of the schedule
- It removes your dependence on the knowledge of a very few veteran operators

- PCS has been part of an Early Availability program and is now generally available.
- One large company has been managing all their production under PCS control since July of 2011
- Another managed all the work in their development CA 7 and used Monitor mode (all the panels, messages, alerts and so on but without actually managing the work) in production during the BETA phase but now are using PCS to manage almost all their production as well.