ThruPut Manager®

AND DYNAMIC INITIATORS

*Dynamic **Initiators** that give you the **control** you need*

***ThruPut Manager** satisfies the requirements for total z/OS batch workload automation. This paper discusses the initiator requirements, and how **ThruPut Manager** optimizes and automates batch job queuing and initiation.*

Initiators and Batch Automation

When dynamic initiators are mentioned, one thinks immediately of initiators that start and stop automatically. But when seen as part of a complete automation solution for z/OS batch processing, there is much more to the issue.

At MVS Solutions, we have developed an unprecedented automation solution for batch. Our **ThruPut Manager** product is a full service-oriented batch workload management system. It takes the service goals for each job and uses them to manage the workload throughout the batch job lifecycle.

A datacenter needs to balance the performance of every job with the optimization of the overall batch throughput. Therefore our automation algorithms also optimize resource utilization and throughput of the workload as whole.

ThruPut Manager uses its automation engine to optimize the workload, based on service goals, knowledge of the workload, and datacenter configuration and constraints. It responds to unexpected obstacles and problems (e.g., LPAR down). When goals specify that "these applications or jobs are more important than those", then the engine will deliver the most important jobs when it can't deliver them all.

Here are some examples of the **ThruPut Manager** built-in automation behavior:

- a job that has been waiting for 30 minutes takes precedence over another job with the same goals, but which has only been waiting 5 minutes,
- a job that cannot immediately begin execution is passed over for selection until it is ready,
- a job is managed individually to its target (i.e., not 'averaged' with a group of jobs), making it impossible for a single job to languish,
- when two jobs both need HSM recalls, the more important one is serviced first,
- a job needing DB2 to run is only selected by initiators on LPARS that have DB2 available,
- **ThruPut Manager automatically** limits the number of jobs it starts on an "online" image.

A key function of the automation engine is the provision and placement of the optimum number of initiators, as well as the manipulation of the job queue to make sure the "right" job is the next one to be initiated.

ThruPut Manager Dynamic Initiators: HIGHLIGHTS

- Initiators are started and stopped based on system load, datacenter controls and workload performance against service targets.
- Workload is distributed across systems.
- The datacenter can control which systems can process which jobs.
- Queue time is used to reduce initiator delays.
- Execution time delays are reduced to increase initiator availability.
- Initiators always select the 'best' job to execute next.

ThruPut Manager®

THE LEADER IN BATCH AUTOMATION

8300 Woodbine Avenue, 4th Floor
Markham ON Canada L3R 9Y7
Tel: (905) 940-9404 Fax: (905) 940-5308
Email: marketing@mvssol.com www.thruputmanager.com

What are ThruPut Manager Dynamic Initiators?

ThruPut Manager dynamic initiators are JES2 initiators managed by **the automation engine.**

Starting and stopping Initiators

ThruPut Manager decides to add or remove initiators based on current system load and datacenter policies. Firstly it looks at WLM, IRD and PR/SM performance data, as well as Service Class performance – the same information available to Workload Manager. Secondly, it performs its own loading and availability analysis. Thirdly, it verifies that adding an initiator is appropriate within the controls set by the datacenter. Fourthly **ThruPut Manager** considers whether the goals of the workload are being met, i.e., how the overall workload is doing relative to its TARGET threshold.

Datacenter Controls

The datacenter normally has goals or policies from a resource utilization point of view, which they specify as constraints to **ThruPut Manager**.

A datacenter can restrict the total number of jobs allowed to run on each LPAR or across the JESplex; it could also make the restriction for just a certain category of jobs. It could restrict the rate of CPU usage in the same way.

A datacenter can also specify that only a subset of jobs is to run on an LPAR, optionally only when they are missing a target. For example, "No batch on the on-line image except if a production job from the Accounting department reaches the CRITICAL threshold."

The datacenter may specify a constraint to route work based on who "owns" the image, either for security, compliance, or business reasons.

The datacenter may route workload to one or more images where software or DBMSs needed by that workload are available. An important instance of this case is when licensed software is required. Many datacenters save licensing costs by restricting the software to particular images and making sure all the jobs that need that software run on one of these images.

Queues Feed Dynamic Initiators

Jobs wait in the queue until selected by TM-managed initiators. **ThruPut Manager** automates the management of this queue to optimize the overall workload throughput.

The characteristics of the **ThruPut Manager** queue are:

- A single queue is used.
- The queue is dynamically reordered.
- Queue time is used to prepare jobs for execution and reduce the time initiators are tied up.

Why a single queue?

Queues traditionally separate workload by significant job characteristics (e.g., number of tape drives) represented by the job class. **ThruPut Manager** analyses every job and therefore can detect any job characteristic directly without reference to the class. A single queue provides service efficiency similar to the single line to wait for bank tellers.

Reordering the queue

ThruPut Manager reorders the jobs in its queue so that the right job is always at the top. A job pushes ahead in the queue if it is closer to missing its service goals than other jobs. A job with aggressive targets pushes ahead faster than a job with less aggressive targets. A job might not take the top position if it is waiting for a resource, say an HSM recall. (It would only tie up an initiator waiting for that recall and another job could be finished executing before the HSM recall was completed.)

When jobs are in jeopardy of missing their targets, then the most important jobs are moved to the top of the queue, based on their relative importance level. Initiators select from the top of the queue, mindful of the datacenter goals and constraints.

Taking Advantage of Queue Time

With z/OS, a job waits in the initiator while resources are obtained, one by one. If the resource is not immediately available, there is a delay. This is all the more disconcerting since the system makes no attempt to fetch the resources while the job is idling in the queue.

ThruPut Manager makes effective use of queue time by

Batch Initiator Developments Over the Last 50 Years

fetching resources as soon as the job is submitted; the resources are available from the moment the job is selected. Resource fetching is prioritized to give preference to more important jobs. Further, **ThruPut Manager** fetches all required resources in parallel, not serially as standard z/OS does. These resources include HSM recalls, virtual volume staging, tape drive booking, and so on.

An analogy may be used to explain the essential difference between JES2 and **ThruPut Manager** resource fetching. If an airport ran like JES2, the plane would roll out to the runway, and then load the baggage. After that was done, any food needed would be loaded, and once the final food cart was secured, the passengers would board. Then the plane would be fuelled. Once that plane took off, the next plane would roll onto the runway to repeat the story.

In reality, planes use gates to load baggage and food, board passengers and top up fuel, simultaneously. Only when the plane is ready to go is it eligible to have a runway assigned.

If you think of gates as queues and runways as initiators, **ThruPut Manager** uses queue time to fetch all the resources the job will need. Delays that a job would have experienced to fetch resources during execution are eliminated.

Many installations strive for near-zero batch queue time. **ThruPut Manager's** approach is to use queue time effectively for job preparation and then execute the job as quickly as possible. This has been proven over many years of experience to deliver the best turnaround to users with the most efficient utilization of system resources.

Dynamic Initiators and Selection

All things being equal, **ThruPut Manager** automatically distributes the workload across the available initiators. However, it is important to route workload effectively. For example, if the datacenter has restricted the number of jobs or batch CPU usage on an LPAR and that limit has been

reached, then the workload is routed elsewhere. Or if the job requires one or more software packages, then **ThruPut Manager** will direct it to an LPAR where the correct combination is available.

Dynamic Initiators and Execution

A prioritized set of batch Service Classes is reserved for **ThruPut Manager** to use. It manages workload during execution by assigning the appropriate Service Class to each job as it enters execution. It verifies that the Service Class is receiving service before starting another job in that Service Class.

One of the objectives of **ThruPut Manager** is to minimize delays. Much of this is achieved by the preparation activities while jobs are in the queue, as discussed previously. However run-time delays can still occur due to dataset contention and tape drive shortages. Because **ThruPut Manager** automatically mitigates these delays, the initiator utilization is increased and the turnaround time for a job is improved.

Where do the Service Goals come from?

If all your batch jobs had equal importance, equal demand for external resources and consumed the same CPU, then you wouldn't need goals. However, every datacenter needs to give preference to more important workload, whether they call it "production", or a "mission-critical" application, or a special class of work. Once that's taken care of, other workload can be accommodated. Of course this might have many gradations.

Service Level Agreements might define the service targets and workload priorities. However many datacenters do without formal service level agreements, and are still expected to provide service to certain goals.

In any case, the datacenter defines batch service goals to **ThruPut Manager** in policies. Jobs are analyzed to determine which goals apply to them. Datacenter criteria and constraints regarding resource utilization are also defined in the policy.

The automation engine uses the policy to determine what level of service each job is entitled to, as well as to deploy the available datacenter resources (LPARs, initiators, tape drives) according to the datacenter goals.

When you need new goals, you change the policy. Only one policy is active at a time. **ThruPut Manager** expects the policy to change over time (e.g., MONTH END policy, HOLIDAY policy, DISASTER RECOVERY policy). Whether the policy change is anticipated or spontaneous, the **ThruPut Manager** automation engine transitions from one to another seamlessly.

Occasionally an operator may need to start a job without consideration of the overall workload. An example may be a 'business-critical' job such as an urgent report for a VP. **ThruPut Manager** continues to control the rest of the workload according to the goals in the active policy.

Summary

ThruPut Manager dynamic initiators are designed for batch and have taken into account the unique characteristics of a queued workload with a wide variety of service goals and resource characteristics. With **ThruPut Manager**, its dynamic initiators allow the batch workload to be optimized automatically based on system load, datacenter controls and workload performance against its service targets. They delivers the best possible results.

ThruPut Manager

Only certain highlights of ThruPut Manager have been discussed here. For further information, please contact us as noted on the first page.

006