

The Incredible Shrinking *Batch Window*

Is your Batch Window gradually closing down, leaving you with less and less room to breathe? Or does your Batch Window sometimes come crashing down with a thud? If you're a Datacenter Manager, chances are your fingers have already been trapped more than once. Why is this happening more and more frequently? And what, if anything, can be done about it?

Why is the Batch Window *Shrinking?*

First, let's define the Batch Window. For most companies it's the time between the end of the online day and the beginning of the next online day, during which all the regularly scheduled consolidation, balancing, reporting and database maintenance tasks must be completed. Given that definition, a couple of possible reasons for the squeeze on the batch window become obvious, namely the extension of the online day and the growth in the amount and complexity of the data.

Both of these reasons are related to the takeover and merger fever of the past few years. As companies extend the reach of their networks to encompass more offices and provide e-commerce solutions, they often encompass more time zones. And clearly the amount, and probably the variety and complexity, of data will have increased significantly. Throw in the ultra-competitive atmosphere of the nineties and you probably have longer retail or service hours to take care of as well, further limiting your batch window.

Unfortunately these factors don't lend themselves to solutions within the reach of datacenter management. However there are a number of other factors contributing to the squeezing of the batch window that are within reach and are readily resolvable.

Jobs Waiting In The Queue

All too frequently jobs sit in the execution queue waiting for a free initiator of the right class. This is especially vexing when the initiator is in use by a misclassified non-production job.

Unfortunately, adding initiators can compound the problem, causing resource over-allocation and excessive swapping.

Jobs Waiting In The Initiator

Many times a job starts and then has to wait, while operators find and mount tapes, for dataset contention, during allocation recovery, while recalls complete, and so on. This results in jobs taking significantly longer to run than they should and in initiators once again being needlessly tied up.

Jobs Running Slowly

A typical reaction to initiators being tied up is to start more initiators. This results in more contention for CPU, I/O and real memory, resulting in higher paging levels and slower system performance. It also increases the probability of dataset contention and allocation recovery for tape drives, further slowing down work. Too many jobs simultaneously accessing the databases and contending for DB2 threads also makes jobs take longer to run.

Another common problem is misuse, deliberate or otherwise, of performance groups and now, particularly since they're usually based on the job class, Workload Manager service classes.

Reruns Because Jobs Fail

Reruns can play havoc with your schedule. Sometimes they're caused by program or data errors, but all too frequently they're caused by JCL or routing problems, for example when a job needing access to a DB2 or IMS region runs on the wrong processor.

So
what's the
Answer?

The answer is ThruPut Manager, a JES2 batch management product in use at over 250 of the largest installations in North America alone.

ThruPut Manager intercepts all batch jobs on the way into the system, whether submitted automatically by a job scheduler, by a TSO user, or from any other source. ThruPut Manager thoroughly analyzes each job (using IBM's interpreter to decipher the JCL), builds a job profile and provides you with a simple way to dynamically apply your rules to those jobs.

Minimize Queue Wait Time

ThruPut Manager knows the characteristics of each job and also what resources a job needs. It can therefore ensure that all jobs are classed and prioritized correctly. If you have different classing structures for scheduled and unscheduled workload, ThruPut Manager provides you with the tools to fence these competing workloads off from each other and prevent interference from less critical work. Given that most schedulers submit jobs only when their prerequisites are complete and the job is logically ready to run, even ten seconds of unnecessary delay per job amounts to a significant overhead over the whole batch stream.

Minimize Wait Time In The Initiator

ThruPut Manager knows when a job needs manual mounts or HSM recalls and can initiate action before the job is passed to the initiator and hold the job until the resources are ready. Dataset contention problems are avoided and are resolved according to your priorities before the job initiates. Similarly, ThruPut Manager can ensure that required tape drives are available, avoiding allocation recovery situations.

Improve Job Performance

ThruPut Manager speeds up processing by helping you to use your initiators more efficiently and therefore reduce the overall number. It ensures that the correct performance group or Workload Manager service class is selected for the job.

It also provides you with the tools to limit jobs based on various factors, such as the number of concurrent batch jobs using DB2 threads. If there are more jobs than threads available, ThruPut Manager will automatically control the number of those jobs allowed to run, improving the overall efficiency.

Minimize Reruns

In addition, ThruPut Manager provides excellent support for implementing Hyperbatch and/or Batch Pipes, shortening the path length of your critical path.

With ThruPut Manager, JCL errors are caught as soon as the job is submitted, giving you more time to make corrections. Jobs requiring access to a DBMS or CICS server are automatically routed to the system where the server is running when the server is running.

With ThruPut Manager[®]

You Can...

- catch JCL errors at submission time
- class and prioritize your work automatically, whatever its source
- remove interference from less critical work
- initiate HSM recalls early
- avoid allocation recovery problems
- avoid initiators waiting for mounts or robotics entries
- avoid dataset contention problems
- avoid overuse of DB2 threads
- reduce the number of active initiators
- use your initiators more efficiently
- take advantage of parallel techniques

Given even a conservative batch critical path of about 200 jobs, saving thirty seconds per job can shrink the batch window by one and a half hours! ThruPut Manager helps your datacenter to run cleaner and more efficiently, with fewer problems. Once ThruPut Manager is installed and running, the datacenter is free to modify classing structures, initiator setup and operational standards as required to best meet their batch window commitments, without negative impact on the user community.

For further information contact:

MVS Solutions Inc., 8300 Woodbine Avenue, 4th Floor, Markham, ON Canada L3R 9Y7

Phone: 905-940-9404 Fax: 905-940-5308

email: marketing@mvssol.com Web: www.mvssol.com



ThruPut
Manager[®]

ThruPut Manager is a registered trademark of MVS Solutions Inc.