Why is it so important to keep the number of service classes to a minimum? Is this to reduce WLM overhead?

• No. There are a couple of reasons why it helps to have no more than 25 to 35 active service class periods at any point in time:

- WLM responsiveness: At each 10 second policy interval, WLM selects the service class period in the most need of help and takes one policy action on its behalf. The more active service class periods you have that need assistance, the more 10-second cycles it will take WLM to take all the needed policy actions for these service classes. This could reduce WLM's responsiveness in managing to goals.

- Better WLM decisions: As part of its decision-making, WLM makes projections for how work will behave when resource changes are made. The accuracy of these projections depends on the quality of the history data gathered for a service class. The more like work that can be put into a single service class, the more statistically valid will be its history data, and the better will be WLM's decisions.

How do I make sure my "loved ones" get the best possible treatment in goal mode? I worry that I will lose control of them under goal mode. Should I put them in SYSSTC?

• Firstly, make your "loved ones" the most important work in the system, typically Importance 1 or 2. This means that the "loved ones" goals will be honored over all other goals. However, use Importance 1 only for workload which needs fast access to CPU, but not for your main production workload.

• Secondly, set velocity or response time goals that reflect your true requirements for this work.

• Thirdly, opt for response time goals if you can, especially for CICS or IMS. When CICS or IMS regions are recognized as transaction servers with response time goals, WLM gives them protective storage targets even before they encounter page delay problems.

• SYSSTC will certainly give the "loved ones" extremely favorable CPU access. However, if you are running storage constrained with CICS or IMS, you may be better off with aggressive response time goals. This way you retain the proactive storage protection afforded to servers with response time goals.

• If you want no limits to the level of service for the "loved ones", you could set very aggressive or even unattainable goals. This further reduces the possibility that the "loved ones" would ever become a resource donor to other work in the system. Keep in mind, though, that the "loved ones" would then be allowed to consume your system when the workload increases.

What workloads are currently exploiting enclaves? How do I know if they are being used on my system?

• The workloads listed below use enclaves. Enclaves represent transactions that do not have address space boundaries. They can span many address spaces or execute concurrently in a single address space. WLM is able to manage and account for these enclave transactions separately from the address space where they are executing. You can expect the list of exploiters to grow as new types of workloads are ported to the System z platform:

WORKLOAD	WLM subsystem type
DB2 distributed queries	DDF
DB2 sysplex query parallelism	DB2
Websphere Application Server for z/OS	CB
Tivoli NetView for z/OS	NETV
IBM Tivoli Directory Server	LDAP

- If you have work running in the service classes associated exclusively with any of the WLM subsystem types listed above, you have enclaves! In goal mode, if you do not classify work that is using enclaves, it defaults to the SYSOTHER service class with a discretionary goal. This service class can be used as an indicator of unclassified work.

- If you have long-running enclaves, you can display the currently active ones using RMF Monitor III Delay report. This report displays the generic category *ENCLAVE. Using cursor sensitive control, you can get a list of individual enclaves and, using cursor-sensitive control again, get a pop-up window showing classification details on the enclave.

How does WLM manage initiators for Unix System Services?

- WLM rather than APPC creates address spaces on demand for Unix System Services fork and spawn processes. WLM does not dynamically manage the number of these spaces as it does for batch initiator management. They are created strictly on demand.

- When a forked child process ends, its address space is returned to a pool to be used by a new forked child. If these address spaces go unused for 30 minutes, they are terminated. If you want to limit the number of address spaces created for forked children, use the MAXPROCSYS parameter in the BPXPARMxx parmlib member.

- These address spaces, sometimes called OE initiators, are classified by WLM under the OMVS subsystem type. The transaction name (that is, jobname) is by default the userid plus a number 0-9. However, if an enclave is active at the time of the fork or spawn, the child process runs under the enclave's service class and managed to the enclave's goal. For example, if an Webserver transaction makes a CGI request, the resulting child process is managed and reported as part of the original Webserver transaction and does not use the initiator's service class. Therefore, you would find service consumed by the CGI processing reported in RMF under the service class of the Webserver transactions.

I set my response time goals based on the observed average response time and response time distributions from RMF, but my online work eats up my system during busy times, and no other work gets done.

- Set response time goals based on the response time during peak periods only. Using a long-term average response time that includes periods of high and low usage gives you a goal that is too aggressive for the peak periods. Select the response time that was achieved during the heaviest peak period to ensure you have a goal that is achievable. A similar method can be used with velocity goals, that is, select the velocity that was achieved during the heaviest peak period. This yields the lowest velocity that still meets the needs of the workload.

Can you mix goal types for service class periods that have the same importance?

• Yes. Service class periods are compared one to another by calculating a performance index (PI) for each period. The PI gives WLM a common way to track how well the work is doing regardless of goal type. The PI is 1 if the work in the period is meeting its goal exactly. The PI is less than 1 if the work is doing better than its goal. The PI is more than 1 if the work is missing its goal. For work at the same importance level, WLM attempts to equalize the PIs.

What are the advantages of an enterprise-wide workload management policy?

• If you have an enterprise-wide service definition, where service class goals, importances, and classification rules are the same across all sysplexes, you can move workloads among systems more easily. The relative business importance of the work is preserved across the move.

How can I manage the individual transactions within the HSM address space?

• Because individual HSM transactions are not known to WLM, HSM can be managed only as an address space with a single velocity goal. It is classified under the STC subsystem type.

I am in goal mode and have been running fine with velocity goals for my CICS regions. Why should I switch to response time goals for CICS transactions?

• As you have found, well-selected velocity goals with an appropriate importance can be as effective as response time goals. However, response time goals have some advantages over velocity goals:

  - Velocity goals need to be reconsidered whenever a hardware upgrade is made to reflect the difference in velocities due to processor speed or number of CPs. Response time goals are based on end user requirements and would not necessarily change across an upgrade.
  - With a response time goal, you are able to use a single reporting product like RMF to obtain meaningful measurement data for all your work, including response time averages and distributions for TSO, CICS, and other interactive work.

When WLM is managing the CICS regions as server address spaces, where is the CICS CPU usage reported?

- The CPU usage is reported in the service class of the CICS regions (either started tasks or jobs), not in the service class of the CICS transactions themselves.

When I migrated my goal mode system to CICS 4.1, the CPU usage of the WLM address space increased noticeably. Why?

- In CICS 4.1, WLM collects CICS delay samples. The sampling overhead is affected by the CICS MAXTASK setting. Check to see if this setting is higher than it needs to be.

Is service accumulating in the SYSOTHER service class a problem?

- Work in subsystems that use enclaves can suffer performance degradation if left unclassified in the service definition. If you do not add classification rules for this work in your service definition, it will be assigned to the SYSOTHER service class, which has a discretionary goal. Using the WLM application, you need to add classification rules to assign the work to service classes with appropriate response time or velocity goals.
- In general, you should keep an eye on the SYSOTHER service class through RMF or another monitor. Any service accumulating in this service class indicates you have unclassified work in the system. You need to find out what it is and get it into a proper service class.

I've heard that achieved velocities can change with processor speed and multiprocessing level. How do I select a single, sysplex-wide velocity goal for a service class if I have a mixed hardware sysplex?

- It's a fact of life with velocity goals that the same work can achieve different velocities on systems of different speeds and different numbers of CPs. The dilemma then is how to select a single velocity goal in a sysplex with mixed machines. Some helpful hints:

  - Use response time goals whenever possible.

  - For work that requires velocity goals, don't try to be too precise in selecting a number. Small differences in velocity goals have little noticeable effect. For example, pick velocities that represent "slow", "medium", and "fast". These might be 20%, 50%, and 80%, respectively. The higher velocity goals are more sensitive to processor differences, so when selecting a fast velocity goal, select one that is attainable on the smaller processors.

  - Revisit velocity goals after a processor change.

Does the z/OS operator command RESET jobname,QUIESCE still allow a non-swappable address space to be dispatched?

- Yes. The RESET QUIESCE command swaps out a swappable address space, and gives a non-swappable address space the lowest dispatching priority. This means non-swappable work can still get dispatched if no other work is ready to be dispatched.

When I switch to a new WLM policy, how long does this disrupt the currently executing workload while WLM accumulates new history?

- The disruption is less than a minute. The impact of a policy switch is reduced by resetting history for only the service classes directly affected by the policy change. For example, if a policy switch only changes the response time goal for service class CICSHIGH, only the history for CICSHIGH is reset. All other service classes are unaffected.

How do I recycle a WLM initiator that is getting ABEND822s?

- There are two ways to do this:

  - If you can determine the ASID of the initiator that is abending, you can stop it by issuing the command P INIT,A=*asid*. The initiator does not need to be idle at the time that you enter the command. If the initiator is busy processing a job, it will stop itself after the job finishes. WLM will automatically replace the initiator with a new one.

- If you cannot determine the ASID, or if you want to recycle all initiators as part of a regular cleanup process, you can enter the commands $P XEQ and $S XEQ. The $P XEQ command causes all WLM initiators on that system to be "flagged" to terminate. The $S XEQ command enables WLM to start new initiators (without needing to wait for the old initiators to end). Beware that the $P XEQ command purges WLM's history which tells it how many initiators are needed for each service class. It may take some time for WLM to build up the same number of initiators that existed before.

The management of the z/OS' high private area can sometimes result in fragmentation and stranded subpools caused by large imbedded free areas known as "holes".
Some fragmentation can also occur in a region when a job initiator starts multiple jobs without being stopped and then started again. If you define the region as having the maximum allowable storage size, it is possible to start and stop the job the first time the initiator is used, but to have an S822 abend (insufficient virtual storage) the second time the job is started. This is because of the fragmentation that occurs.
In this situation, either the region has to be decreased, or the job initiator has to be stopped and restarted.

After I switch some jobclasses to WLM management, how do I know how many initiators WLM has started?

- There is no special display for the number of WLM initiators because this is not something you can control manually. The real test of whether WLM created "enough" initiators is seeing that the work is meeting its goals. The methods suggested below should be enough to assure yourself that the function is indeed working.
- Note that for methods 1 and 2, the counts could include JES initiators if you recently transitioned from JES-managed jobclasses to WLM-managed. If a job is already executing at the time its jobclass is changed to MODE=WLM, it is allowed to complete normally. Once the job completes, the JES initiator is no longer used for that jobclass.
  1. *By jobclass:*
     Use the SDSF JC display to find the jobclasses that are WLM mode and look at the count of executing jobs for each job class. The actual number of WLM initiators may be slightly higher than the sum of the executing jobs in WLM-managed classes. This is due to idle initiators that are kept around for a period of time in case additional jobs become eligible for selection.
  2. *By service class period:*
     Look at the RMF workload activity report to find out the average number of address spaces for a service class period (AVG ADRSP). If you adhere to the recommendation of not mixing JES-managed batch work and WLM-managed

batch work in the same service class, then AVG ADRSP indicates the average number of WLM initiators servicing the period.

3. *By system:*
   If you know how many traditional JES initiators you have, issue the D A,L command and subtract the JES inits out of the total initiator count on this display. The result is the number of WLM-created initiators for the system.
4. *Idle initiators:*
   To see the number of idle WLM batch initiators, use the SDSF DA OINIT command which shows only idle initiators. Those without JOBIDs are WLM initiators. WLM initiators don't have a JOBID because they are started under the master subsystem. You can set up an SDSF filter to show only the WLM idle initiators.
5. You can also use the WLM Work Queue Viewing Tools† (WLMQUE) to view the WLMmanaged initiators.

DB2 Stored Procedures: We can schedule a procedure in DB2, but when we do a display thread the procedure is shown waiting for WLM to schedule it. The procedure never executes, and eventually times out with a return code that doesn't provide a definite reason. How can we determine what happened to the procedure? Does the z/OS operator command RESET jobname,QUIESCE still allow a non-swappable address space to be dispatched?

- Check your SYSLOG for IWM034I messages indicating that WLM has started the procedure. If you find those messages, see if you can determine what happened to the address spaces when they started. If you don't see those messages, you should see IWM035I messages instead, indicating that WLM couldn't start the procedure.
- You can also use the WLM Work Queue Viewing Tools (WLMQUE) to view the queue server address spaces.

– – –

† **WLM Work Queue Viewing Tools**
WLMQUE, the WLM Work Queue Viewer is a small ISPF based tool that may assist you in displaying the application environments that are currently being used on your z/OS system. This can be helpful for using WebSphere Application Servers when you specify minimum and maximum limits for the number of server address spaces that should be started. You can view the number of started and active server address spaces, and the service classes being used as work queues for the application environments with the help of the REXX command list. The tool can be used for any kind of application environment from WebSphere, DB2 or user specified types and applications.

NOTE: WLMOPT, the WLM OPT parameter viewer assists you in displaying the current OPT settings of your z/OS system. The tool is stabilized on the level of z/OS 1.10. That means no OPT parameters which have been introduced after z/OS 1.10 can be displayed with it. Instead you can use the RMF Monitor II Report "OPT Settings" which is a new selection under "Library Lists". All future OPT parameters will only be made available through the new RMF report.