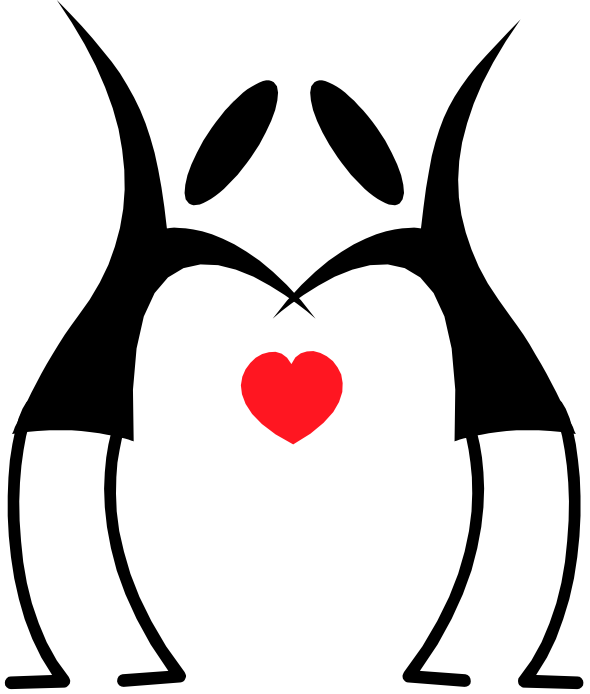
CMG2015 – San Antonio, TX



WLM Caused Pain and Pleasure

By Ivan Gelb

TRADEMARKS

- The following are trade or service marks of the IBM Corporation: CICS, CICS TS , CICSPlex, DB2, IBM, MVS, OS/390, z/OS, Sysplex, Parallel Sysplex. Any omissions are purely unintended.
-
  URL: www.gelbis.com
  Phone: 732-303-1333

MOTHER OF ALL DISCLAIMERS (MOAD)

- All of the information in this document is tried and true. However, this fact alone cannot guarantee that you can get the same results at your place and with your skills. In fact, some of this advice can be hurtful if it is misused and misunderstood. As with all kinds of analysis, anything you may hear or read can be understood and misunderstood in many ways that may seem contradictory to you. Gelb Information Systems Corporation, Ivan Gelb and any one found anywhere assume no responsibility for this information's accuracy, completeness or suitability for any purpose. Anyone attempting to adapt these techniques to their own environments anywhere do so completely at their own risk. ☺

CMG-T Sessions Today + Panel

- Part 1 - Best Practices for Improved z/OS Performance and Lower TCO
- Part 2 - WLM Caused Pain and Pleasure
- Part 3 - Holistic z/OS Performance and Capacity Management
- PANEL – zEnterprize Q&A with Kathy Walsh and Norm Holander

Agenda

- Your Questions… welcome anytime
- WLM Background
- Service Policy Boot Camp
- Case Study

PLUS: Rewards likely for your questions and brain droppings

Best Practices / Recommendations

- ➲ Design WLM service policy to mirror business activity – this enables the most effective Performance Management (PM) & Capacity Planning (CP) activities. You will know when you need to do something, and what you may need to pay closer attention to.

- ➲ Always communicate results to C-management in common business terms: orders, deliveries, effect on total cost...etc. Example:

  - ➲ The average CICS/DB2 response time / trade order decreased by 50% so we can handle twice as many trades now without any additional hardware or software cost.

WLM Planning and Migration

- The Washington Systems Center's "WLM Migration Guide and Checklist, Version 3.1"
- Compatibility to Goal Mode Migration Considerations
- The Washington Systems Center's Sample Service Definition
- Cheryl Watson's Articles, including "Why Go to Goal Mode?" and "Getting to Goal Mode"
- Cheryl Watson's WLM Quickstart Service Definition
- Get the latest versions of the IBM Red Books

Potential Delay Sources

- Wait for CPU so another application can run
- System and application page-in delays
- I/O-s related waits like IOS queue, channel service, device
- Some DASD devices are not providing good performance
- Application “logic” delays

WLM Can help all of the above except…application “logic”

How WLM Protects “Loved Ones”

- Question for everyone: Do you know your “Loved Ones”***
- WLM’s “protection” of an application can be implemented at many levels:
  - WLM (z/OS Workload Manager) and IRD (Intelligent Resource Director) in the SYSPLEX
  - WLM service policy coded relative priorities
  - WLM CPU priority protection (avail. as of OS/390 Version 10)
  - WLM Storage protection (avail. as of OS/390 Version 10)
  - PR/SM LPAR weights
  - WLM I/O priority protection

***“Loved Ones” = If you ask for some $s for care of these workloads, management just asks: “How much?”

What Does WLM Manage?

WLM managed resources:

- Processor (CPU)
- Storage controls
- Multi-programming levels (MPL)
- I/O priority
- Parallel I/O access volumes
- WLM Controlled JES initiators
- DB2 stored procedure address spaces
- Websphere scalable address spaces

WLM Control of Resources

- Insure that critical business workloads are defined in service policy so they are easy to observe and analyze.
- WLM controls:

What?	How?
CPU access priority	Task dispatch priority guided by importance and service level goal
CPU time limits	Defined resource groups
I/O performance	Priority propagations and PAV-s (parallel access volumes)
Enclaves for DDF, stored procedures, etc…	With coded minimum and maximum service level definitions
Dynamic batch initiators	Goal and resource driven controls
Storage paging	With isolation to protect working set size

What Are The Priorities & Goal Types?

- WLM Priorities:
  - Pre-defined service classes
    - SYSTEM (fixed CPU DP=255)
    - SYSSTC (fixed CPU DP=254)
  - Importance 1 – 5
  - Discretionary – when you DON'T NEED TO CARE!

- WLM's three goal types:
  1. Percentile response time
  2. Average response time
  3. Velocity – guarantees CPU access only, not priority

WLM Performance Index (PI)

- PI < 1 results are better than goal
- PI = 1 results are meting goal
- PI > 1 results are not meeting goal
- PI Calculation formulas:
  - Response time goal PI = Actual response time / Response time goal
  - Velocity goal PI = Velocity goal / Actual velocity

- **WLM activities will yield similar PI values within each importance level unless some workload is constrained in a way that WLM can not “fix” (such as application ENQ)**

WLM Service Class Goal Types

- Pre-defined service classes
  - SYSTEM (fixed CPU DP=255)
  - SYSSTC (fixed CPU DP=254)
- Percentile response time
- Average response time
- Velocity
- Discretionary – when you just DON'T NEED TO CARE! **And** great for best I/O bound batch throughput

WLM Options

- Identify work as CPU Critical
- Identify work as Storage Critical
- CICS regions can be managed based on:
  - Transaction response time goals **or**
  - Region level response time goals **or**
  - Region level velocity goals

WLM CPU Critical Protection

- Problem before WLM R10
  - WLM's slow reaction to sudden increases of CPU demand for the service class

- Since WLM Rel. 10 option CPU Critical = YES
  - Defined for single period service classes with velocity or response time goals
  - Service class' CPU dispatching priority will be kept higher than all less important work even if all of goals are being met and exceeded

WLM Storage Critical Protection

- Problem before WLM R10
  - Fluctuating activity levels exposed application to page stealing that results in demand paging related delays
- Since WLM R10 option Storage Critical = YES
  - A region's working set is kept very close to its high water mark (HWM)

Definitions Do-s and Don’t-s

- Service Definition Coefficients
- SYSSTC
- Percentile Response Time
- Average Response Time
- Velocity Goals
- “Loved Ones” Protected?
- WLM Related Sanity Preservers
- Case Study: DB2
- Case Study: Constant Contention

Service Definition Coefficients

- CPU = 1.0
- SRB = 1.0
- IOC = 1.0 (or consider lower value like 0.1)
- MSO = 0.0
  Any other MSO coefficient will result in systems with unstable performance, and certain workloads with most velocity will run poorly regardless of total load on the system. A workload may underperform most times.

SYSSTC

- THE place for “trusted servers” and
- Usual candidates
  - VTAM,
  - JES but . . .
  - IRLM
  - IMSCTL

Percentile Response Time

- The recommended way to manage loved CICS production work
- Stated as:
  - 90% of transactions with < 1 sec. Resp.
- Can address problems caused by long running or never ending transactions

Average Response Time

- Can work if workload is homogeneous – different units of work require very similar amounts of computer resources and similar service goals

- Stated as:
  - ALL transactions < 1 sec. AVG. Resp.

- Problem:
  - “Fooled” by long running transactions ending in the interval

Velocity Goals -1

- "Execution velocity is an abstract mathematical description with no objectively measurable metric."
  --John Arwe, IBM WLM Developer

- Velocity calculated from *sampled* states: CPU_Using_Time / (CPU_Using_Time + WLM_Managed_Delays)
  - WLM Managed delays: CPU, paging, swapping, MPL, IOs

Velocity Goals -2

- Velocity goals NEVER work as tools for relative priority
- Velocity goals do not determine CPU dispatching priority
- Many application systems velocities **fluctuate** severely due to one or more of the following:
  - Interactions with other work in the system
  - Workload has a low mean time to wait
  - Workload is infrequently ready to run

Velocity Goals – 3

- When velocity goal is reached, your LOVED application will be 1st on the donors list of WLM – this HURTS!
- Turns your LOVED application into “BRAIN DONOR!”
  Lover importance work will run when higher imp. work reaches velocity.

Velocity Goals – When?

- Recommended for non-transactional work (a.k.a. BATCH), or work that “needs” a limiter
- Low importance + low velocity can control known “loopers”
- Velocity goals typically require lower WLM overhead to manage
- Recommendation: Consider use of resource group maximum with velocity goals to impose an absolute limit

“Loved Ones” Protected?

- ➲ New options since OS/390 Release 10:
  - Identify service class as CPU Critical
  - Identify service class as Storage Critical – you WILL HURT less important work and increase system paging rate. You should be concerned about this in 64-bit mode systems that are storage poor! They page to disks!
- ➲ Watch out for I/O priority shifts caused by multi-period service classes

“Loved Ones” Protected?…. NOTES

What is a “loved one?” A workload that your business is willing to spend resources on to maintain and/or improve its quality of service. In other words, work that is most important for your business’ income & reputation.

WLM may do service policy adjustments every 10 seconds and resource adjustments every 2 seconds. This is not fast enough for critical online work. Once a lower priority task is moved above your “loved one’s” CPU dispatch priority, the few seconds required to regain the higher CPU dispatching priority can cause plenty of missed service goals.

- CPU CRITICAL attribute solves this problem. Lower priority work’s priority will not be raised above higher importance work with this attribute set.
- STORAGE CRITICAL solves the problem of paging by protecting the working set of your favorite work.
- I/O priority shifts caused by multi-period service classes may increase / decrease the performance of a loved workload in unexpected ways. Be aware this potential exists by identifying such service classes.

WLM Related Sanity Preservers - 1

- Remove display of actual CPU dispatching priority from any tool used by non-systems staff – display only the importance.
- Always create at least the following two “extra” service classes:
  - Service class with a resource group that has maximum service unit rate of ONE for control of loopers.
  - High importance one (perhaps equal to SYSSTC in priority) for boosting the performance of some important piece of work.
    Note: Operator can reset work into serv. classes

WLM Related Sanity Preservers - 2

- What to do if questioned about the always present CPU delay?
  1. Try to focus questioner on a realistic goal because this delay will never be eliminated.
  2. Try to explain the MVS reduced preemption approach, the cause of always present delay causes:
     - Avoids interrupting running tasks
     - Expects normal work to release CPU rather soon
     - Employs timed preemptions just in case…

WLM Related Sanity Preservers - 3

- Always avoid unrealistic goals or ones very close to what may be possible.
  - WLM will waste time trying to “help” such work until it just gives up.
  - If WLM fails in trying help, it will ignore the affected work for multiple minutes.
- IO priority management
  - Must be enabled for dynamic alias management of Parallel Access Volumes (PAV)
  - Consider use of static PAVs – requires ongoing checking of how well/poorly some volumes perform

WLM Related Sanity Preservers - 4

- Audit your service policy with the SMF 99, subtype 5 records.
  - Do not leave this collection ON! Produces many records.
  - The subtype 1 – 5 are for IBM debugging exercises.

- Insure that the collection of SMF 99s is OFF normally!

Case Study: Unstable DB2 Performance

- If CPU utilization increased to over 95% during any 30 minute period, DB2 response time would begin to wildly fluctuate.
- CICS, DB2 involved
- Significant DB2 activity generated directly from internet as well as CICS regions

Case Study: Unstable DB2 Performance – 2

- Where to look?
  - CPU activity reports from various sources -
    Showed that utilization was at 100% a lot of the time
  - Degradation analysis reports -
    Showed virtually every task within the system as THE cause of the problem
  - IO activity reports –
    Did not show any unusual activity between the good v. the bad response time periods
- So who "done" it? The **Butler** of course!

Case Study: Unstable DB2 Performance – 3

- WLM Service policy did it!
  - All service classes regardless of importance, had velocity goals, and
  - The sum of CPU time consumed by velocity goals of the active service class periods always exceeded processor capacity – **What???**
- The "quick" fix:
  - Introduced response time goals for some service classes
  - Used CPU Critical attribute for importance 1 work
  - Reduced velocity goals of lower importance work

Case Study: Lessons to Learn

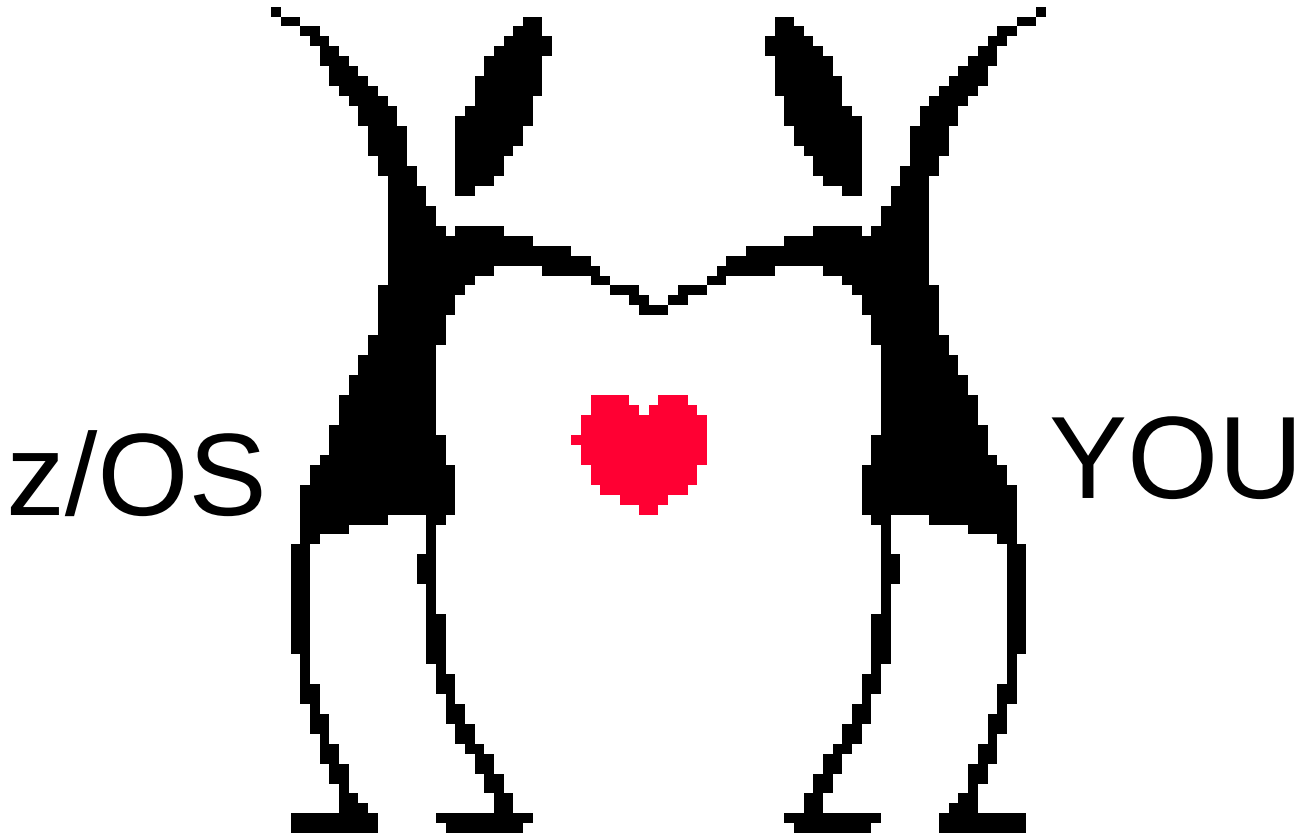
- Total processor time required for all velocity goals of the active service class periods exceeded processor capacity during peaks
- Ounce of prevention:
  - Approximate and monitor expression of a workload's velocity goal in processor capacity terms (MIPS, MSUs, %Utilization)
  - Test and observe performance frequently to insure the system's stability within design limits
  - Consider use of very short RMF collection intervals (1 – 2 minutes) for analysis

Summary

- Design WLM service policy to mirror business activity.
  This enables the most effective Capacity Planning & Performance Management (CP & PM) activities.
  You will know when you need to do something, and what you may need to pay closer attention to.

- Effective CP & PM activities will yield required service level quantity and quality for least total cost. Now go and convince your management of this! Be careful!

Questions?



Speaker Information

Ivan Gelb

E-mail: ivan@gelbis.com

Phone: 732-303-1333

www.gelbis.com

Your questions and comments are always welcome.