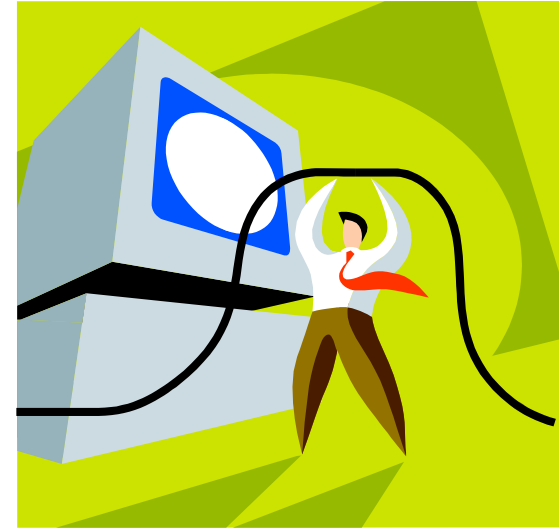


SHARE Anaheim – Aug. 8, 2012



Top 10 CICS Tuning Tips

By Ivan Gelb



Think *Faster* with
Gelb Information



© 2012 Gelb Information Systems Corp.

Any questions? Email to: ivan@gelbis.com; Call: 732-303-1333

Trademarks & Copyright

- The following are trade or service marks of the IBM Corporation: CICS, CICSplex, DB2, IBM, z/OS, Parallel Sysplex. Any omissions are purely unintended.

© 2012, Ivan L. Gelb

Gelb Information Systems Corp.

10 Country Club Lane, Marlboro, NJ 07746-1447

Phone: 732-303-1333 E-mail: ivan@gelbis.com

Permission granted to reproduce presentation only in its entirety and include all copyright notices. All comments, contributions and questions are welcomed and rewarded.

Think *Faster* with
Gelb Information



© 2012 Gelb Information Systems Corp.

Any questions? Email to: ivan@gelbis.com; Call: 732-303-1333

Disclaimer

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 workplace. In fact, some of this advice can be hurtful if it is misused and misunderstood. Gelb Information Systems Corporation, Ivan Gelb and anyone 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. 😊 😊



Agenda

- Your Questions @Anytime
- CICS Performance Analysis
- Best Practices



Think *Faster* with
Gelb Information



© 2012 Gelb Information Systems Corp.

Any questions? Email to: ivan@gelbis.com; Call: 732-303-1333

CICS Performance Analysis – Holistic View

Performance Hierarchy

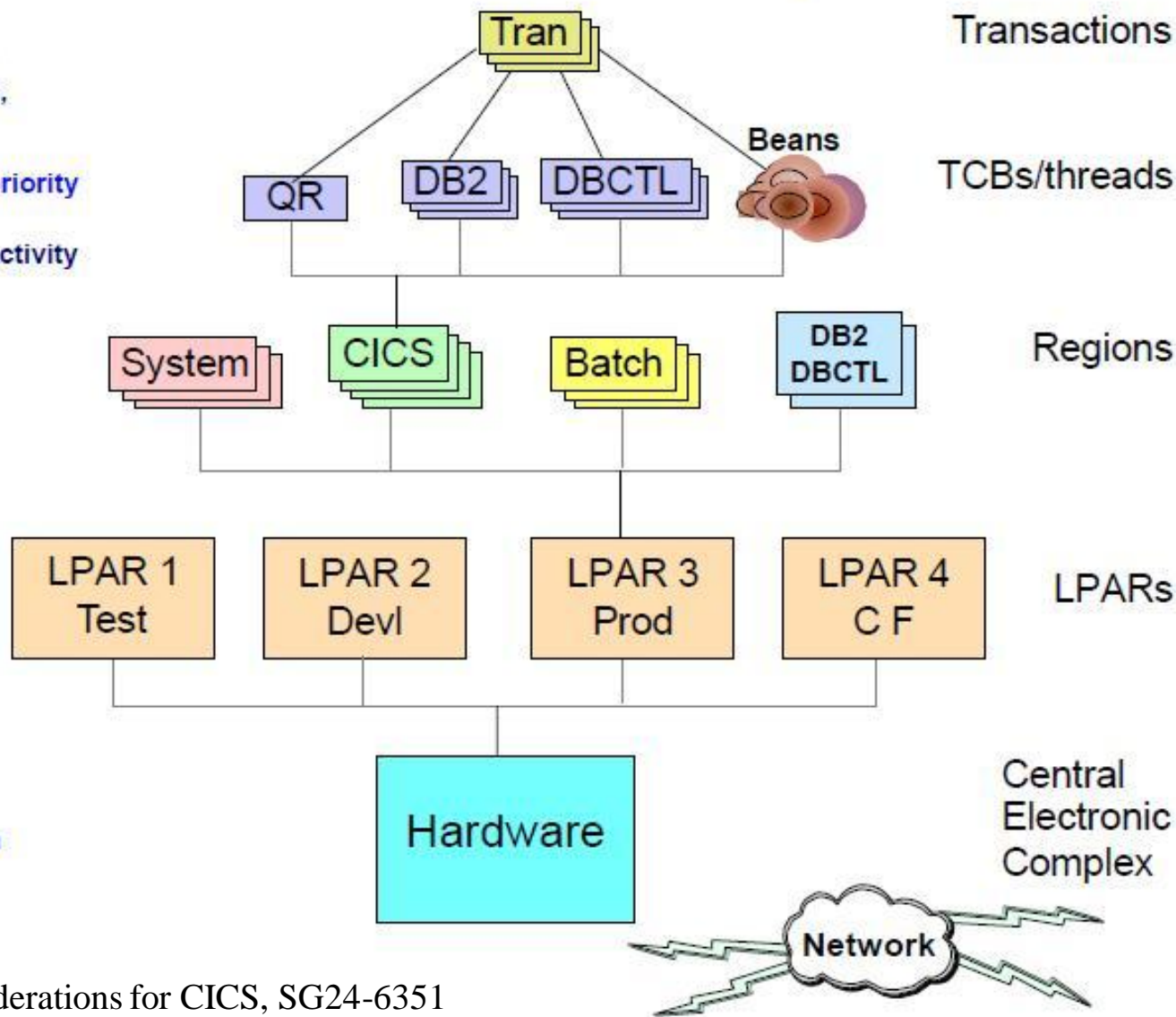
CICS dispatching and tuning; resource usage
Tools: SMF 110 records, statistics

TCB/thread execution priority and contention
Tools: RMF Workload Activity Reports

MVS dispatching and tuning; SRM/WLM
Tools: RMF Reports, SMF Type 30 records

LPAR Mgmt (weights and fair share)
Tools: RMF CPU and Partition data reports

CPU cycles - Engines, Memory, Devices, Hardware configuration
Tools: RMF



Source: Threadsafe Considerations for CICS, SG24-6351

Think *Faster* with
Gelb Information



© 2012 Gelb Information Systems Corp.

Any questions? Email to: ivan@gelbis.com; Call: 732-303-1333

CICS Performance Analysis Top 10 Questions

1. What is the complaint, and are any other older complaints **still** around/being reported and **unresolved**?
2. When did this problem **1st** occur and occur the **last** time?
3. What planned changes occurred during this time period?
4. Were any changes made to z/OS WLM service policy?
5. Did the total system workload composition change?
 - A. Within all the LPARs (not just the affected CICS one)?
 - B. Across all LPARs on this processor complex (CEC)?
6. Did the total load of all LPARs change, and/or is 95-100% physical CEC utilization hit when CICS service degrades?
7. Did any PR/SM LPAR weights or caps change, or switched from uncapped to **hard-capped** LPARs?
8. Was anything moved around (files, clients, regions,..)?
9. What were the most recent changes to the application?
10. How did transaction rates change within affected CICS region?



Best Practices for CICS Performance and Capacity Management

Think *Faster* with
Gelb Information



© 2012 Gelb Information Systems Corp.

Any questions? Email to: ivan@gelbis.com; Call: 732-303-1333

Best Practice (BP) 1 – WLM Service Policy

- **Recommendation:** Use WLM Average or Percentile Response Time goals for best WLM protection available for CICS performance.
 - Percentile response time – likely the easiest and requires the least monitoring and active management
 - Average response time – requires more active monitoring and management to maintain.
 - Let's dissect the WLM service class goal types...



Percentile Response Time Goal

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

Note: RMF Workload Activity Report is the easiest and least overhead way of working with and reporting of response time goals.



Average Response Time Goal

- Can work acceptably for homogeneous CICS workloads
- Goal example:
 - ALL transactions < 1 sec. AVG. Resp.
- “Fooled” by long running transactions ending in the interval
- Once “fooled” it will **over-protect** CICS, and **hurt** all the other work in service classes with equal or lower WLM importance



Velocity Goals

- “Execution velocity is an abstract mathematical description with no objectively measurable metric.”
--John Arwe, WLM Developer
- Velocity calculated from sampled states:
$$\text{CPU_Using_Time} / (\text{CPU_Using_Time} + \text{WLM_Managed_Delays})$$
 - WLM Managed delays: CPU, paging, swapping, MPL, IOs,...



Velocity Goals (cont'd)

- Velocity goals NEVER work as tools for relative priority
- Velocity goals do not determine CPU dispatching priority
- CICS TOR velocities fluctuate severely
- CICS TORs performance can be hurt by WLM's efforts to manage them



Velocity Goals – Consider When?

- Recommended for non-transactional work or work that “needs” a limiter
- Low importance + low velocity can control known “loopers”
- Consider use of WLM defined resource group maximum with velocity goals to impose an absolute limit on any work
- CICS test regions



BP 2 – WLM CPU Critical Attribute

- **Recommendation:** Consider use of WLM service class option

CPU Critical = YES

- Defined for single period service classes with velocity or response time goals
- CICS region's CPU dispatching priority will be kept higher than all less important work even if all of CICS' goals are being met.



BP 3 – WLM Storage Critical Attribute

- **Recommendation:** Consider use of WLM service class option
Storage Critical = YES
 - CICS protected from paging because region's working set is kept very close to its high water mark (HWM).
Every other workload could be hurt a lot by paging if not enough main storage!



BP 4 – WLM Batch Initiators Priority

- **Recommendation:** Use **INITIMP** parameter in IEAOPTxx; must be coded to keep initiators' CPU activity below service classes with CPU critical attribute for specified importance levels.
Review current publication and OW55344, OA03581, and OA03870 for the complete story
- **Why?** During job selection, batch initiators run in SYSSTC service class – second highest priority just behind SYSTEM
- Initiators' activity degrades CICS because they will impair work at all importance levels, including importance 1s. Initiators' activity and their related exits often use a lot of CPU at the SYSSTC priority level before the job is classified.



BP 5 – Combine CICS Regions

- **Recommendation:** Combine regions as you move up to faster single CPU speed processors or exploit CICS open TCBs via threadsafe work. Today's z114/z196 series have CPU speeds over 24x greater than those found in the mid-1990s!
- **Why?** Saves 5 – 25%+ CPU if 2-to-1
- Very likely that your production regions were split due to QR TCB saturation or “political” reasons in the region's lifetime.



BP 6 – Customize CICS TRACE

- **Recommendation:** TRACE = NO for all production regions. Use trace in test regions as needed.
- **Why? Saves 10 – 35% CPU!**
 - If **needed** for problem determination in a production region, consider selective tracing by domain or transaction name or user ID, or... – over **30** options available for selective tracing!!!



BP 7 – Monitor Wait for CPU

- **Recommendation:** Always monitor wait-for-CPU-dispatch as a component of the total CICS response time.
Why? Best method of tracking CICS response time degradation caused by wait for CPU.
- CPU wait for dispatch related considerations:
 1. Does the application have delays caused by higher priority work within CICS, and/or within MVS, and/or within the same or other PR/SM LPAR-s?
 2. If answer to any part above is yes, CICS work may lack the proper PR/SM LPAR weight, z/OS WLM importance, CICS transaction priority, and/or not enough processor capacity is available to meet all the service level goals.



CICS Dispatcher Statistics – Time by TCB Mode

DISPATCHER STATISTICS (Note: Columns 2 - 5 deleted to improve legibility)

| TCB Mode | MVS Waits | Total Time in MVS wait | Total Time Dispatched | Total CPU Time / TCB |
|-------------|--------------|---------------------------|--------------------------|-------------------------|
| QR | 13051397 | 000-18:18:33.24 | 000-01:49:46.74 | 000-01:12:02.27 |
| RO | 48658 | 000-20:05:12.28 | 000-00:02:46.27 | 000-00:01:00.80 |
| CO | 0 | 000-00:00:00.00 | 000-00:00:00.00 | 000-00:00:00.00 |
| SZ | 0 | 000-00:00:00.00 | 000-00:00:00.00 | 000-00:00:00.00 |
| RP | 0 | 000-00:00:00.00 | 000-00:00:00.00 | 000-00:00:00.00 |
| FO | 800 | 000-19:00:52.61 | 000-00:00:44.05 | 000-00:00:06.50 |
| SL | 1 | 000-00:00:00.00 | 000-00:00:00.00 | 000-00:00:00.00 |
| SO | 2 | 000-00:00:00.00 | 000-00:00:00.00 | 000-00:00:00.00 |
| S8 | 0 | 000-00:00:00.00 | 000-00:00:00.00 | 000-00:00:00.00 |
| D2 | 2419 | 000-20:18:01.28 | 000-00:00:03.26 | 000-00:00:00.43 |
| L8 | 16952578 | 007-03:07:31.31 | 000-05:36:18.48 | 000-01:13:35.37 |
| H8 | 0 | 000-00:00:00.00 | 000-00:00:00.00 | 000-00:00:00.00 |
| J8 | 0 | 000-00:00:00.00 | 000-00:00:00.00 | 000-00:00:00.00 |

Recommendation: If QR TCB “Total Time Dispatched” is more than 1.25 times “Total CPU Time/TCB,” determine response time degradation and seek increased importance in WLM Service Policy if degradation is significant.

Think *Faster* with
Gelb Information



© 2012 Gelb Information Systems Corp.

Any questions? Email to: ivan@gelbis.com; Call: 732-303-1333

BP 8 – Avoid CICS Limit Conditions

- **Recommendation:** Avoid all CICS limit conditions being hit within CICS. Examples of limit conditions:
 - Max Tasks
 - Class Max Tasks
 - TCB-s
 - Threads
 - Buffers...etc...
 - SOS (short-of-storage anywhere in CICS region!)
- **Why?** The only good reason to hit any limit conditions: You **want** to cause delay and limit performance and capacity of a CICS application or an entire CICS region.



BP 9 – MXT Settings

- **Recommendation:** Keep MXT as low as possible without turning it into a bottleneck unless you intend to cause PAIN!
- **Why?** Minimize z/OS Workload Manager (WLM) CPU overhead by tuning MXT for every CICS region.
 - One PB created for each unit of MXT in every CICS!
 - If transaction management, Performance Blocks (PB) states sampled at 250 millisecond intervals
 - If region management: PBs are sampled every 10 intervals (10 * 250 ms = 2.5 sec)
 - Every PB is sampled every time even if it not in use



BP 10 – RMF Performance Reports

- **Recommendation:** Utilize response time reports from RMF because they are available real-time and use less CPU than the collection of CMF transaction level records for later processing.
- **Why?** 4 – 12% CPU / CICS region can be saved if CMF based response time reporting is replaced with RMF reports of CICS response time. Critical CICS transactions can be isolated in RMF report classes.

Rather than full-time transaction level CMF data collection, selective collection during a few representative periods of activity is recommended.



BP 10 - Performance Reports (cont'd)

- Please Note:
 - If goal = REGION, response times not reported to service class(es)
 - At least TORs must be managed with goal = TRANSACTION to get response time reports from RMF records.
 - **New in z/OS V1.13.** Service classes with velocity goals will also produce response time distribution.
Recommendation: Consider the new reports a tool to assist migration from velocity to response time goals since only these can be associated with business driven CICS transaction specific service level agreements (SLA).



RMF Workload Activity Report

REPORT BY: POLICY=HPTSPOL1 WORKLOAD=PRODWKLD SERVICE CLASS=CICSHR RESOURCE GROUP=*NONE PERIOD=1
 IMPORTANCE=HIGH

```

-TRANSACTIONS-- TRANSACTION TIME   HHH.MM.SS.TTT ← Response time
AVG             0.00 ACTUAL          000.00.00.114
MPL             0.00 QUEUED          000.00.00.036
ENDED           216 EXECUTION        000.00.00.078
END/SEC         0.24 STANDARD DEVIATION 000.00.00.270
#SWAPS          0
EXECUTD         216
    
```

| -----RESPONSE TIME BREAKDOWN IN PERCENTAGE----- | | | | | | | | | | | | | | | -----STATE----- | | | |
|---|-----|-------|--------|-------|------|-----------------------|-----|------|------|-------|-------|-------|-------|------|-----------------|-------------------|-------|-------|
| SUB | P | TOTAL | ACTIVE | READY | IDLE | -----WAITING FOR----- | | | | | | | | | | SWITCHED TIME (%) | | |
| | | | | | | LOCK | I/O | CONV | DIST | LOCAL | SYSPL | REMOT | TIMER | PROD | MISC | LOCAL | SYSPL | REMOT |
| CICS | BTE | 93.4 | 10.2 | 0.0 | 0.0 | 0.0 | 0.0 | 83.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 83.3 | 0.0 | 0.0 |
| CICS | EXE | 67.0 | 13.2 | 7.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 46.7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

Time in DB2 or
IMS or MQ

This is a sample RMF post processor (ERBRMFPP) output with option SYSRPTS (WLMGL(SCPER))

Source: Chris Baker, IBM Hursley, UK

Think *Faster* with
Gelb Information



© 2012 Gelb Information Systems Corp.

Any questions? Email to: ivan@gelbis.com; Call: 732-303-1333

BP 11 – VSAM Recommendation

- **Recommendations: For VSAM.**
 1. Always focus on physical IO reduction via data in virtual techniques.
This benefits CPU, response time, storage occupancy, IO subsystems performance.
 2. Track CICS VSAM file statistics as 1st indicator of tuning candidates. SMF 64 and 42 records are best sources of file performance data.
 3. Produce ongoing reports of total IO time by file and remember 80/20 rule.
 4. Track and minimize CI and CA split activity rates.
 5. Optimize LSR buffer pools (8 maximum until V4.2's increased to 255).



CICS VSAM File Control Statistics

| File Name | Get Requests | Get Upd Requests | Browse Requests | Update Requests | Add Requests | Delete Requests | Brws Upd Requests | VSAM EXCP Data | Requests Index | RLS req Timeouts |
|-----------|--------------|------------------|-----------------|-----------------|--------------|-----------------|-------------------|----------------|----------------|------------------|
| AAAB2SP | 34238 | 0 | 0 | 0 | 0 | 0 | 0 | 22 | 1 | 0 |
| BBBACTV | 0 | 27 | 0 | 27 | 376636 | 0 | 0 | 382501 | 0 | 0 |
| CCCFNDD | 65928 | 0 | 0 | 0 | 0 | 0 | 0 | 15089 | 6228 | 0 |
| DDDIAFD | 4767 | 0 | 25159 | 0 | 0 | 0 | 0 | 12609 | 148 | 0 |
| EEEINTX | 27088 | 0 | 8124 | 0 | 0 | 0 | 0 | 3 | 2 | 0 ☺ |
| FFFPNDD | 17969 | 5310 | 0 | 5310 | 166 | 0 | 0 | 9905 | 799 | 0 |
| GGGSCRX | 488 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 59 | 0 |
| HHHSEGH | 33043 | 43 | 1712 | 43 | 43 | 0 | 0 | 1597 | 841 | 0 ☺ |
| IIISEG1 | 48931 | 6925 | 531 | 2810 | 6739 | 4115 | 0 | 15537 | 2862 | 0 |
| JJJSEG2 | 23634 | 745 | 0 | 205 | 745 | 540 | 0 | 1291 | 1 | 0 |
| KKKTBLs | 537 | 0 | 75997 | 0 | 0 | 0 | 0 | 525 | 26 | 0 ☺ |
| LLLTEST | 0 | 0 | 0 | 0 | 41741 | 0 | 0 | 43761 | 0 | 0 |
| MMMULHD | 54891 | 43 | 0 | 43 | 0 | 0 | 0 | 806 | 453 | 0 ☺ |
| NNNUNLD | 32679 | 1640 | 0 | 1586 | 53 | 0 | 0 | 7319 | 2670 | 0 |
| OOPCFIL | 37752 | 0 | 0 | 0 | 0 | 0 | 0 | 21 | 1 | 0 ☺ |
| *TOTALS* | 427489 | 18626 | 155690 | 13864 | 459660 | 4655 | 0 | 536868 | 15546 | 0 |

Notes & Recommendations:

1. Totals are greater than all files shown because many files deleted from sample.
2. Focus your tuning to minimize/eliminate VSAM EXCP Requests.
3. ☺ shown next to files with superior performance (least EXCP-s/Request).
4. BBB tuning options: faster IO service, application changes, file attributes,...
5. CCC, DDD, III, NNN appear to be good candidates for data in memory tuning.

Think *Faster* with
Gelb Information



© 2012 Gelb Information Systems Corp.

Any questions? Email to: ivan@gelbis.com; Call: 732-303-1333

BP 12 - LSR Buffer Pools - 1

Recommendations:

1. Use multiple LSR buffer pools (see next pg.)
2. Separate data from index within buffer pools
3. Use VSAM file statistics to set size of index buffer pools to equal amount of index data.
4. If faced with real or virtual storage constraint, separate business critical files' buffer pools from less important ones.
5. Eliminate or at least **Minimize** string waits.



BP 12 - LSR Buffer Pools - 2

Recommended LSR buffer pool design:

1. CICS TS V4.2 increases pools to 255! Use new pools for benefit of files requiring guaranteed performance level.
2. Business critical high importance files with data buffers separated from index buffers
3. Medium importance files with data buffers separated from index buffers
4. Low importance files with data buffers separated from index buffers
5. Files with poor locality of reference with data buffers separated from index buffers
6. Pools 6, 7, 8 Use to isolate and measure a file's activity then move into one of 1 – 4



BP 13 – CICS Temp. Storage

- **Recommendation:** Tune CICS Temporary Storage to minimize IO-s and activities that can waste CPU capacity.
 1. Minimize auxiliary storage requests by adding enough buffers and modifying applications that force TS activity to AUX.
 2. Avoid causing spanned TS records via proper CI size.
 3. Reduce/eliminate buffer and string waits.
 4. Avoid format writes with properly sized TS file.



Temp. Storage Statistics

| <u>TEMPORARY STORAGE</u> | | |
|--|---|----------|
| Put/Putq main storage requests | : | 78701 < |
| Get/Getq main storage requests | : | 70899 < |
| Peak storage for temp. storage (main) | : | 135916 < |
| Put/Putq auxiliary storage requests | : | 78756 < |
| Get/Getq auxiliary storage requests | : | 135961 < |
| Peak temporary storage names in use | : | 66 |
| Number of entries in longest queue | : | 58 |
| Times queues created | : | 131425 |
| Control interval size | : | 4096 |
| Available bytes per control interval | : | 4032 |
| Segments per control interval | : | 63 |
| Bytes per segment | : | 64 |
| Writes more than control interval | : | 3 |
| Longest auxiliary temp storage record | : | 32080 |
| Number of control intervals available | : | 3599 |
| Peak control intervals in use | : | 13 |
| Times aux. storage exhausted | : | 0 < |
| Number of temp storage compressions | : | 1507 < |
| Temporary storage buffers | : | 6 < |
| Buffer waits | : | 0 < |
| Peak users waiting on buffer | : | 0 < |
| Buffer writes | : | 22 < |
| Forced writes for recovery | : | 0 < |
| Buffer reads | : | 25 < |
| Format writes | : | 0 < |
| Temporary storage strings | : | 6 < |
| Peak number of strings in use | : | 1 < |
| Times string wait occurred | : | 0 < |
| Peak number of users waiting on string | : | 0 < |
| I/O errors on TS dataset | : | 0 < |
| Shared pools defined | : | 0 |
| Shared pools currently connected | : | 0 |
| Shared read requests | : | 0 |
| Shared write requests | : | 0 |

Think *Faster* with
Gelb Information



BP 14 – CICS Transient Data

- **Recommendation:** Tune CICS Transient Data to minimize IO-s and conditions causing waits.

Employ same principles as tuning any single VSAM file's activity plus minimize waits for buffers and strings.



BP 15 – Consider THREADSAFE

- Best possible performance if:
 - All applications are written and defined threadsafe
 - All exits within the task-related user exit (TRUE) path are written and defined threadsafe
 - Use on non-threadsafe CICS commands is minimized
 - Intermingling of non-threadsafe CICS commands with SQL or IP Sockets calls is minimized
- Learn more about threadsafe from:
 - Redbook: Threadsafes considerations for CICS, SG24-6351
 - SHARE Proceedings of CICS project presentations



BP 16 - Re-Consider FCQRONLY if. . .

- FCQRONLY={YES|NO}
 - For CICS V4.1 or older, FCQRONLY=YES for best performance of all FORs
 - For CICS V4.2
 - specify FCQRONLY=YES for MRO and ISC connections so FC request uses only QR TCB and avoids locking overhead
 - Specify FCQRONLY=NO for IPIC connections so FC requests stay on L8 TCB which avoids potential for switching overhead and increases parallelism of FC requests



BP 17 – Consider with CICS TS V4.2

- Consider for CICS TS V4.2
 - New TSMMAINLIMIT={64M|amount}
 - Automatic TS cleanup with use of EXPIRYINT specified on a TSMODEL definition. Non-zero value causes system clean up task to run every 30 minutes. Defaults to zero value.



BP 18 - CICS DPMODE for DB2

- **Recommendation:** Choose **DPMODE = Which one??**
- **Important Note:** All 3 will work **OK with little measurable difference** if non-CPU constrained in a multi-CP complex!
- **DPMODE=HIGH**
 - Works best for high volumes with little DB2 use as long as ample CPU capacity is available with multiple CPUs/complex
- **DPMODE=EQUAL (is/was CICS 2.2 default!)**
 - **May(!)** provide better performance for non-SQL transactions
- **DPMODE=LOW**
 - **Can** provide more **consistent** service in CPU constrained situations – otherwise only an experiment will tell which is best for a specific workload.



References

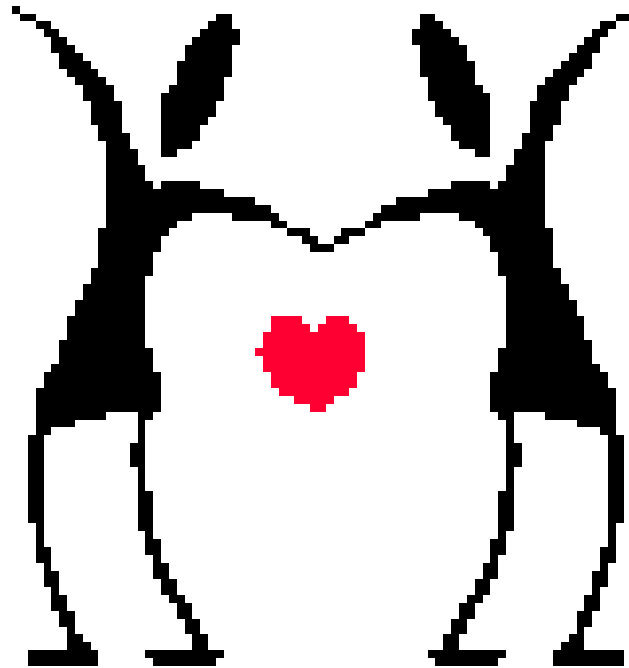
- CICS Information Center – IBM CICS Transaction Server for z/OS
<https://publib.boulder.ibm.com/infocenter/cicsts/v4r2/index.jsp>
- CICS Performance Guide
- CICS Performance Management Guide
- Videos On YouTube (Follow CICSfluff):
 - Performance comparison between CICS TS V3.2 and CICS TS V4.1 and CICS TS V4.1 with WLM
 - Threadsafe Analysis with the CICS Explorer and CICS Tools





Join Us @ SHARE in 2013

CICS



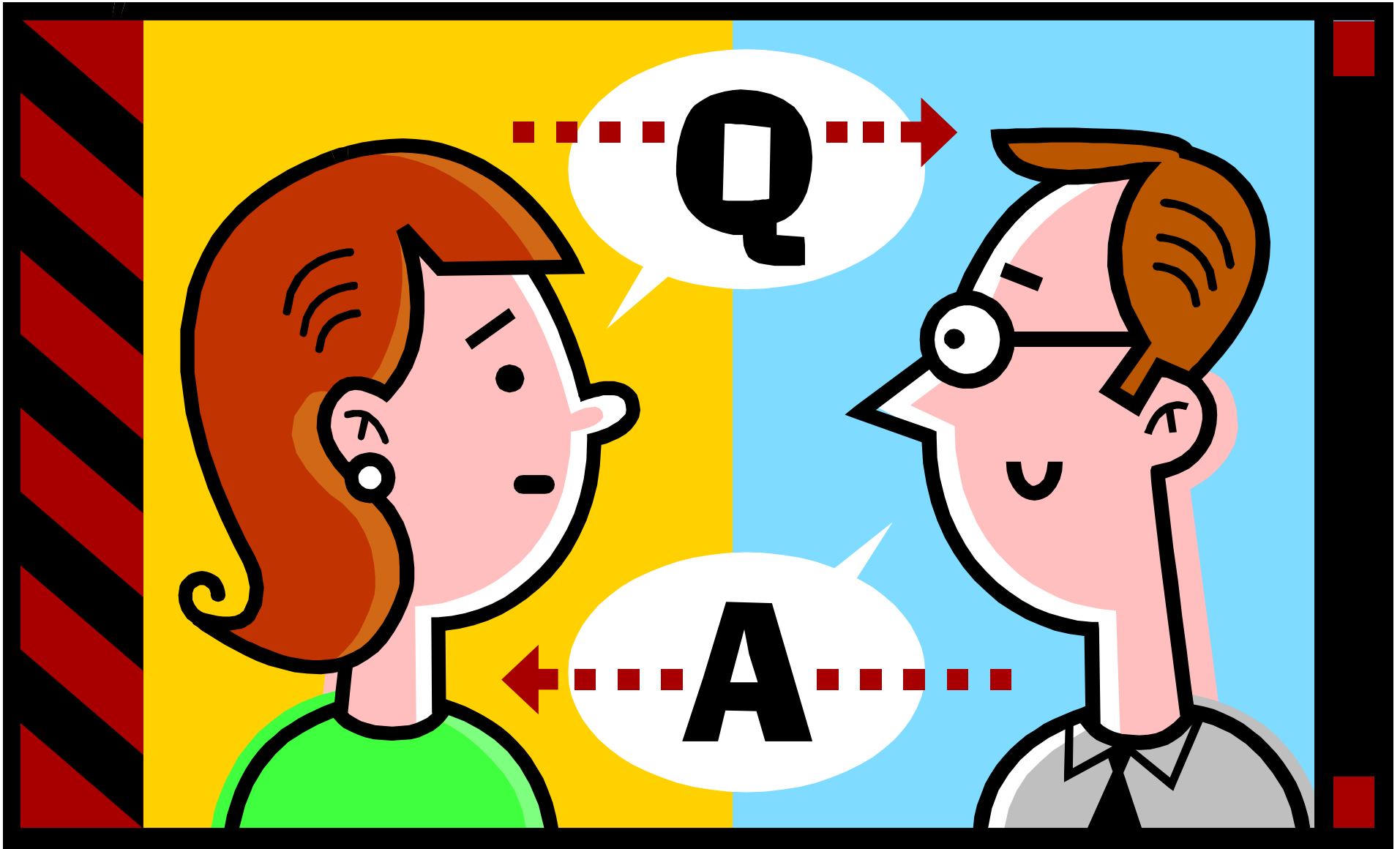
YOU

Think *Faster* with
Gelb Information



© 2012 Gelb Information Systems Corp.

Any questions? Email to: ivan@gelbis.com; Call: 732-303-1333



Think *Faster* with
Gelb Information



© 2012 Gelb Information Systems Corp.

Any questions? Email to: ivan@gelbis.com; Call: 732-303-1333