



A CICS Health Check
Mike Poil
CICS Level 3 Service
IBM Hursley
poilmike@uk.ibm.com

Important Disclaimer

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the operational environment. There is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

Performance data contained in this document was determined in a controlled environment and, therefore, the results which may be obtained in other operating environments may vary significantly.

This presentation is aimed at CICS TS for VSE/ESA. When the word "CICS" is used, it refers to CICS TS for VSE/ESA unless specified otherwise.

The content of this presentation has mostly come from actual experience on customer systems.

Agenda

- What is a Healthy CICS System?
- What can affect the health of CICS?
- LPAR configuration and the use of multiple cpus by z/VSE.
- DASD.
- z/VM configuration.
- z/VSE configuration, shared 24-bit storage, PRTY and SVA-resident code.
- LE Configuration.
- CSI TCP/IP configuration.

Agenda

- CICS DSALIM sizing.
- CICS MRO versus ISC.
- The CICS Dispatcher.
- Things to avoid in CICS and why.
- DFH0STAT versus DFHSTUP.
- CICS monitoring data.
- CICS cpu time accuracy.
- Exception and historical reporting.

What is a Healthy CICS System?

- I would say that a healthy CICS system is one that is performing as well as can be expected or agreed, and is not subject to unnecessary degradation and outages.

- You may have your own definition.

- If you need a Health Check, IBM can help you (for a price ☹)

 - <http://www-01.ibm.com/software/htp/cics/service/>
 - Due to a technical problem, CICS TS for VSE/ESA is not mentioned, but it can be done.

What can affect the Health of CICS?

- **Hardware Environment**
 - Capacity and LPAR definitions.
 - Dasd.
 - Network.

- **Topology and Availability**
 - CICS topology and configuration.
 - Hardware issues.
 - Single points of failure.
 - External service provider problems.
 - Abends.

What can affect the Health of CICS?

■ Applications

- Lack of suitable unit, acceptance and load testing.
- Performance.
- Affinities/design, possibly conflicting with CICS configuration and/or the way CICS is designed to work.

■ Software Levels

- Base IBM software levels
- RSL and RSU - is it better to be proactive and not wait for the problem to occur?
- Check for IBM APARs that are relevant, there is a definitive CICS APAR list at URL:

<http://www-01.ibm.com/support/docview.wss?uid=swg27015142>

What can affect the Health of CICS?

- z/VSE 4.n APAR DY47204/DY42710/DY42715 fixes a problem in CICS where the application DSA return address is overlaid with an odd address, causing the DFHEIP BSM 0,R14 to switch into 64-bit addressing for both the caller *and* the CICS ESTAEX exit; CICS is in a very unhealthy "hung" state after multiple program checks with code 38 ☹. You used to get a program check code 6 for an odd address, but that is how BSM works in z/Architecture mode, BASSM does the same ☹.
- z/VSE 4.3 VSAM HIPER APAR DY47271 - an *occasional* loop between CICS and VSAM trying to resolve an exclusive control error after a split; many consecutive AP 04B7 exception trace entries are produced - the loop does stop.
- Maybe check z/VSE 4 VSAM DY47083, DY47084 and DY47085.
- **OEM software - make sure that a suitable level of the product and the appropriate fixes are installed.**

What can affect the Health of CICS?

■ Performance and Monitoring

- Do you know what your typical daily z/VSE and CICS usage looks like, and when your peaks are? If not, why not?
- Are you aware of any limits? (Probably not yet!)
- Do you get warnings for exception conditions, including invisible limits such as MXT so that you can react?
- Monitoring issues - lack of tools, tools that do not provide the right data, tools that duplicate data, tools that collect and produce too much data.
- Do you know how to correctly interpret the data? ***This is a very important point.***
- Lack of suitable historical data and lack of effective capacity planning - remember to factor in new release deltas and a change in the processor type; there may be changes in options that can have a positive or negative impact, some of these changes may now be the default!

What can affect the Health of CICS?

■ Change Control

- Bad procedures/execution.
- Where is the documentation, is it up-to-date, and does it cover the environment end-to-end?
- "I never changed anything, well, I did change this, but I don't think that it is related to the problem . . ." (the problem that only happened since that change was made that is!)

LPAR Configuration

- **Is the Hardware up to the job?**
 - Has it got the processing power?
 - Tests have shown that cpu per transaction increases as cpu utilisation % increases, this is a factor of the additional management overheads and the CP slowing down.

- **LPAR processor configuration**
 - Check the relative weights, there may be a "Short Engine" affect when the hardware runs at capacity and PR/SM enforces them.
 - z/VM and/or z/VSE do not know that a physical CP is not attached to a logical CP.
 - A CP appears to be "capped" and may cause queues to form due to the lack of access, this can impact CICS as most of its work is on one z/VSE subtask - QR.
 - Your monitor may show you a difference between LPAR dispatch time and actual LPAR cpu time.

LPAR Configuration

LPAR processor configuration (contd.)

- An example with 2 real CPs available:

LPAR	Weight	Share	#Real CP	#LCP	Real CP/LCP %
ONE	400	44%	<u>0.88</u>	2	44%
TWO	400	44%	<u>0.88</u>	1	88%
THREE	100	11%	<u>0.22</u>	2	11%

- Real CP/LCP is the PR/SM-enforced single CP maximum utilization when the hardware is 100% busy, this is applied to all LCPs.
- When weights are applied, LPAR ONE is only allowed to use 44% of a real CP, TWO is allowed to use 88% of a real CP etc.
- Suppose that there is one very busy CICS in LPAR ONE and it was only able to use 44% of a CP . . .

z/VSE and Multiple CPs

- "In general, a faster single CPU is better than multiple smaller CPUs".
- Every extra CP adds management overhead, an extreme case . . .
- I recorded the cpu time of 3 CICS partitions using a DPL and Function Shipping driver (a worst case scenario), this was z/VSE 4.2 running under z/VM 5.4 on an LPAR with 12 shared CPs:
 - MRO 1 CP total 334 seconds partition 233 z/VSE overhead 101
 - MRO 2 CP total 550 seconds partition 325 z/VSE overhead 225
 - MRO 3 CP total 678 seconds partition 342 z/VSE overhead 336
 - ISC 1 CP total 675 seconds partition 606 z/VSE overhead 69
 - ISC 2 CP total 826 seconds partition 641 z/VSE overhead 185
 - ISC 3 CP total 837 seconds partition 639 z/VSE overhead 198

z/VSE and Multiple CPs

- **YMWV.**
- Investigate z/VSE 4.2 Cpu Balancing (Rexx code can do it on previous releases) and only use the number of CPs that are required based on the actual workload?
- For CPU balancing, each cpu must either be **ACTIVE** or **STOPPED** to activate it.
- If a cpu is in a **STOPPED** state when balancing is activated, it will not be used . . .
- So, if you have 3 cpus, and activate balancing with 2 cpus **ACTIVE** and 1 **STOPPED**, it will only ever use 1 or 2 cpus.

z/VSE and Multiple CPs

Multiple CP limit example:

AR	0015	CPU	STATUS	SPI N_TIME	NP_TIME	TOTAL_TIME	NP/TOT
AR	0015	00	ACTIVE	18852	6100562	13713442	0.444
AR	0015	01	ACTIVE	17906	5424786	12294566	0.441
AR	0015	02	ACTIVE	13405	4685876	10013585	0.467
AR	0015			-----			
AR	0015	TOTAL		50163	16211224	36021593	0.450
AR	0015						
AR	0015			NP/TOT: 0.450	SPI N/(SPI N+TOT): 0.001		
AR	0015	OVERALL UTI LI ZATION:		120%	NP UTI LI ZATION:	54%	

- The NP/TOT (NPS) ratio for 3 CPs is not optimum.
- NP/TOT suggests a maximum of about 200% based on the z/VSE formula:
Max CPUs = 0.9 / NPS.
- 120/54 = 220% - around this point the NP utilization is 100%, which is a limiter.

DASD

- See the WAVV z/VSE presentations for the detail.
- Fast dasd is important for CICS, the faster, the better 😊.
- SIR SMF is a free way to look at dasd I/O performance, perhaps use Rexx to parse and analyze it. An example without PAV active:

```
sir smf,vse,261
AR 0015 TIMINGS FOR 261 BASED ON 589 I/O INSTRUCTION
AR 0015 MAXIMUM I/O QUEUE 3          ← What you need is average not maximum
AR 0015
AR 0015   QUEUED      PENDING    CONNECT    DISCONN    DEV.BUSY    TOTAL
AR 0015   msec/SSCH  msec/SSCH  msec/SSCH  msec/SSCH  msec/SSCH  msec/SSCH
AR 0015     0.000      0.000      0.369      0.000      0.000      0.369
AR 0015 1I40I READY
```

- A device queue must be > 1 to actually represent queued I/O.

z/VM Configuration

- See the WAVV z/VM presentations for the detail.

- z/VSE appears to have been overlooked for some time in the z/VM performance tests, maybe you want to lean on them a bit. I can recommend some guys with Italian connections 😊.

- **Paging**
 - z/VSE normally runs with PAGEX ON to allow z/VSE to run during some z/VM-level page fault processing, but, a z/VM page-in for the CICS QR task still stops it.

- **MDC**
 - Use MDISK MINIOPT to target hot spot z/VSE volumes?

- Use the IPL NOPDS option to reduce some z/VSE page management overhead.

z/VSE Configuration

- Use SIR to look at some common ones; NOPDS is active in this example so it does not show any z/VSE overall paging activity.

```

AR 0015 CPUID VM = 003D23B320640000          VSE = FFC24C2420640000
AR 0015 PROCESSOR = IBM 2064-116 83 (123B383) LPAR = WINVME      No. = 0013
AR 0015          CPUs = 0012 (Ded. =0000 Shr. =0012) Cap. = 75%
AR 0015 VM-SYSTEM = z/VM      5.4.0 (1003)   USERID = WINVSEE   VMCF = ON
AR 0015          CPUs = 0003                      Cap. = 25%
AR 0015 PROC-MODE = z/Arch(64-BIT) IPL(120)   02:25:16 GMT 01/31/2011
AR 0015 SYSTEM   = z/VSE      4.2.0 GA              08/13/2008
. . .
AR 0015 TASKS ATT. = 00044          HIGH-MARK = 00045          MAX = 00163
AR 0015 DYN. PARTS = 00004          HIGH-MARK = 00005          MAX = 00048
AR 0015
AR 0015 COPY-BLKS = 00003          HIGH-MARK = 00074          MAX = 01517
AR 0015 CHANQ USED= 00014          HIGH-MARK = 00021          MAX = 00204
AR 0015 LBL. -SEGM. = 00014        HIGH-MARK = 00016          MAX = 00717

```

z/VSE Configuration

■ LUBS (not shown in SIR)

- You might as well allocate 255 for static partitions in NPGR, the storage is already allocated in the Supervisor phase itself.

■ Partitions limit

- IPL SYS NPARTS=n.

■ Tasks limit

- z/VSE 4.2 SYSDEF SYSTEM,NTASKS=n else fixed at 255.

■ Paging

- A page fault *with a page-in* forces the task to wait for I/O until it is resolved, that could be CICS.
- More real storage and/or bigger z/VM virtual machine size.

z/VSE Configuration

■ Copy Block limit

- EXCP forces the task to wait until they become available after an I/O interrupt, the waiting task could be CICS.
- IPL SYS BUFSIZE=n.
- VTAM IOBUF31=YES may affect the usage.
- Check APARs DY47158, DY47179, DY47181 and DY47222.

■ CHANQ limit

- EXCP forces the task to wait until an entry becomes available after an I/O interrupt, the waiting task could be CICS.
- IPL SYS CHANQ=n.

z/VSE 24-bit Shared Storage Optimization

- Use z/VSE 4.3 😊.
- I have provided a Word document that will be available for download along with this document, and it explains how to do optimize 24-bit shared storage.
- Have a read, if you have any questions, you have my email address.

z/VSE PRTY

- CICS needs a high PRTY setting, be careful setting its relative priority and if you use PRTY SHARE mixing CICS and batch.
- If you have several connected CICS, changing the relative priorities may improve the overall performance.
- Make sure that your CICS external monitor partitions are at a higher priority, failure to do that can occasionally produce *interesting* results!
- Check DMF'S priority is higher than CICS if it is used.
- The default partition balancing MSECs 976 may not be optimum for balancing and PRTY SHARE, reduce and monitor to see if you get better performance.

What about SVA-resident code?

- In the old days it was about saving real storage and placing the code in key zero storage so that the average program could not walk over it, which is still perfectly valid.
- Do not add 24-bit code if it messes up the SVA-24 sizing.
- On the likes of the z10 or z196, it may reduce actual cpu time by enabling the instruction cache to be used more efficiently.
- Better cache usage = more MIPs for free 😊.
- Starting at the z10, there is special instrumentation to measure things like the average number of cycles per instruction - Google "z10 cpu measurement".

LE Configuration

- **LE configuration options can impact CICS performance and 24-bit storage usage, these are generally well-known.**

- **ALL31(OFF)**
 - The default ALL31(ON) may be possible even if you have 24-bit programs.
 - OFF increases C and PL/I 24-bit STACK (DSA) storage (COBOL mostly uses HEAP).
 - IBM or OEM software can be affected, e.g. WebSphere MQ is written in C.
 - Maybe not all of the CICS partitions need ALL31(OFF), use two CEECOPT phases.
 - If you change to ON, make sure that you change HEAP and STACK to ANYWHERE.

- **CHECK(ON)**
 - Activates COBOL subscript range checking, the default is OFF.
 - What is the cost of a Storage Violation versus the cpu time overhead of having it on?

CSI TCP/IP Configuration

- Make sure that you have the latest zaps/PTFs installed that could affect CWS, and all of the relevant CICS PTFs as well.
- Slow CWS (socket close) performance and even other problems may be fixed by ZP15F276 or whatever supersedes it; you need to add `BSDCFG1=$OPTCNFW` in the `SOCKOPT` assembly.
- `SOCKOPT BSDCFG2=$OPTCHKR` can cause some `TCPIPSERVICE` problems due to TCP/IP dead sockets reuse not being correct in all cases, causing `DFHSO0130` abends with return code 1,112 - APAR `PM05556` describes the fix.
- Use `FTPBatch` instead of the normal FTP client.
- Multiple stacks?

CICS General Performance Notes

- **Select "Best Practice" configuration values for the SIT and resource definitions - see other WAVV presentations.**
- **Tune usage - see other WAVV presentations.**
- **Identify and eliminate limit conditions - see other WAVV presentations.**
- **Are you using out-of-date configuration settings that no longer apply?**
- **Maybe you should consider putting CWXN in a TRANCLASS with PURGETHRESH set to a number to avoid a flood killing CICS.**

CICS DSALIM Sizing

- **SOS is definitely not healthy.**
- **All of the CICS partition less 4K should be GETVIS.**
- **DSALIM is allocated in extents of 256K (EDSALIM is 1MB extents).**
- **Can I increase DSALIM?**
 - **Monitor the GETVIS usage over time to find a peak, then use the smaller of (AREA SIZE - MAX. EVER USED) and LARGEST FREE GETVIS.**
 - **Don't use all of it.**
 - **Don't forget GETVIS xx,RESET after CICS initialization to reset the 24-bit MAX. EVER USED from its default of AREA SIZE, Appendix A has sample PLTPI code.**
- **Check CICS statistics if you want to exchange DSALIM for Getvis.**

CICS DSALIM Sizing

getvis g1

AR 0015	GETVIS USAGE	G1-24	G1-ANY		G1-24	G1-ANY
AR 0015	<u>AREA SIZE:</u>	10,236K	39,932K			
AR 0015	<u>USED AREA:</u>	5,596K	28,800K	<u>MAX. EVER USED:</u>	5,628K	28,976K
AR 0015	FREE AREA:	4,640K	11,132K	<u>LARGEST FREE:</u>	4,608K	10,956K
AR 0015	DYNAMIC-SPACE GETVIS USAGE					
AR 0015	<u>AREA SIZE:</u>	1,024K				
AR 0015	USED AREA:	84K		<u>MAX. EVER USED:</u>	84K	
AR 0015	FREE AREA:	940K		LARGEST FREE:	940K	
AR 0015	1140I READY					

- **DSALIM=5M active, and there is up to 4608K extra, which is 38 * 256K DSA extents, you would not use all of it.**
- **Wow, look how much 24-bit Space Getvis is wasted!**

CICS DSALIM Sizing

- As well as for user code, there may be some DSALIM savings by tweaking CICS transactions such as CSMI to use 31-bit storage.

- Make sure you test it fully, CSMI should work for Function Shipping, but abend for DPL to 24-bit code.

- A CICS task has 4 storage subpools:
 - User storage below the line - multiples of 4K if any storage is GETMAINed.
 - CICS storage below the line - multiples of 4K.
 - User storage above the line - multiples of 64K.
 - CICS storage above the line - multiples of 64K.

CICS MRO versus ISC

- Both are expensive in terms of cpu time.
- Both use noticeably more cpu time than CICS/VSE.
- This is W.A.D. - sorry ☹️.
- MRO uses a lot of CICS SVC 150s, two are used for each request, one in CICS to start the request and one in the z/VSE Supervisor to end it.
- My notes say 8 SVCs are used for each CICS MRO request.
- Use SIR MON to see SVC counts, this is possibly more efficient than using OEM software.

The CICS Dispatcher

- Is very much a law unto itself.
- It uses your combined priority value to influence the position of the task on its dispatch list.
- Its aim is to make better use of cache by giving tasks that were recently dispatched the highest priority.
- Using `PRTYAGE=0` causes your task priority setting to have a bit more influence.

Things to avoid in CICS and why

■ System dumps (SDUMP)

- A big impact on the subtask that has the problem, typically QR, but not always.
- Avoid application AP0001/SR0001 system dumps, use a PLTPI program to switch these system dumps off as soon as CICS starts.
- DFHSTUP tells you how many there were.

■ Transaction dumps

- Sequential I/O emulation uses synchronous dump dataset I/O, so the whole of QR waits until I/O completion, this is W.A.D. - sorry ☹️.
- DFHSTUP tells you how many there were.

■ Extrapartition I/O

- A necessary evil - serialized sequential writes ☹️.
- Did you know that any Extrapartition dataset is fully device-independent?
- The CICS DFHDCT does not generate DTFs, it generates DFHDCBs.
- Only CICS journaling uses DTFs.

DFH0STAT versus DFHSTUP

- Why use them when you have an OEM performance monitor?
- They produce useful CICS statistics that cover many of the indicators of the state of health, they have a zero or low overhead and are free!
- DFH0STAT provides less information, and . . .
- DFH0STAT gets information from internal control blocks, and when you run it at the wrong time with the wrong settings, you can lose some important data, this is not the case with DFHSTUP:
 - If you have `STATRCD=ON`, some statistics are reset at the defined interval.
 - Closing a VSAM file causes all statistics to be lost.
- DFHSTUP interval statistics are very useful.

CICS Monitoring Data

- **Very detailed and potentially very useful task-level performance data.**
 - **Internal response time.**
 - **Cpu time.**
 - **Storage.**
 - **Request counts.**
 - **Wait times.**

- **It is a shame that IBM provides nothing except DFH\$MOLS to print it, there goes another forest!**

- **May be used by other products.**

- **Use statistics AND task performance data.**

CICS CPU Time Accuracy

- CICS uses an emulated z/OS STIMER macro to capture cpu time.
- z/VSE records TASK (CICS and user) instruction processing but does not capture much of the z/VSE overhead.
- The reported cpu time is thus subject to a degree of inaccuracy, this will be true for any product that relies on CICS-maintained cpu times.
- CICS statistics only reports CICS Dispatcher-managed TCB cpu time (emulated z/OS subtasks for QR, SO etc.), it does not report on subtasks such as DFHIRPST that are not managed by the dispatcher, nor cpu time for OEM subtasks.

CICS CPU Time Accuracy

- I have a modified version of DFH0STAT and DFH\$STAS that captures partition + overhead cpu time and shows it as "Job step cpu time" (and now includes Dump Statistics).
- Use this or z/VSE performance data at the partition level to work out the capture ratio to see what the real cpu cost is.
- Ensure that your z/VSE monitor includes overhead and partition time.
- The capture ratio could be something like only 60%.
- Keep tracking it over time as it may change.
- This is could be important for capacity planning.

CICS CPU Time Accuracy

- You need a capture ratio for z/VSE as well, so remember to factor in the overhead of higher levels of software and hardware such as z/VM and LPAR management.
- Are you monitoring CICS cpu time usage at time intervals?
- Are you close to the practical limit of one CP for any CICS?

Exception and Historical Reporting

- How many of you have time to scan 100's or 1,000's of pages to look for performance problems?

- **Parse DFH0STAT or DFHSTUP output?**
 - I have VM/VSE Rexx parser code (PARSTAT EXEC) for DFH0STAT exception reporting.
 - Even better, it creates a series of CSV files that can be input into a relational database for simple, but powerful, historical analysis and trend reporting.
 - No response times though ☹.
 - I haven't found time to convert it to handle DFHSTUP output.

- **CICS TS for z/OS has the DFH0XSTR exit for DFHSTUP to be able to analyze each statistics record.**

Exception and Historical Reporting

- Use a modified DFH\$MOLS to output CSV data?
- I did this once, and it is better than using a Rexx parser.
- Historical detailed transaction performance history will require a lot of disk space.

Any Questions ?

Appendix A: GETVIS RESET PLTPI Program Sample Code

- * Issue GETVIS xx,RESET command, where xx is the partition that CICS is in.
- * Do **not** run from a transaction unless you add code to check the way it is started.
- * This is a CICS program, and must be translated before compiling.

```

GVRESET  CSECT
         MVC   SVC30MFE, SVC30MFI
         LA    R0, SVC30MSG
         STCM  R0, B' 1111' , SVC30ADR
         L     R1, 20
         USI NG COMREG, R1
         SLR   R0, R0
         ICM  R0, B' 0011' , PID
         DROP  R1
         GETFLD FIELD=LOGID, PART=(0)
         LTR  R15, R15
         BNZ  RETURN
         STCM  R1, B' 0011' , SVC30PID
         LA   R5, 5           Number of times to try the command
RETRY   DS   OH
         SR   R0, R0
         LA   R1, SVC30MFE
         SVC  30
         LTR  R15, R15      OK?
         BZ   RETURN       Yes
         CH  R15, =H' 4'   Error?
         BH  SVC30ERR      Yes
* RC=4 means AR is busy, wait then retry
         EXEC CICS DELAY FOR SECONDS(1)
         BCT  R5, RETRY
RETURN  EXEC CICS RETURN
SVC30ERR DS   OH
         EXEC CICS WRITE OPERATOR TEXT(SVC30M)
         B    RETURN
         LTORG
SVC30MFI DS   OCL21
         DC   AL2(L' RESET)
         DC   AL4(0)
RESET  DC   CL15' GETVIS xx, RESET'
SVC30M DC   C' PLTPI program GVRESET was unable to issue GETVIS xx, RE*
         SET'

```

Appendix A: GETVIS RESET PLTPI Program Sample Code

```
DFHEI STG DSECT
SVC30MFE DS    OCL21
SVC30LEN DS    HL2
SVC30ADR DS    AL4
SVC30MSG DS    OCL15
           DS    CL7
SVC30PI D DS    CL2
           DS    CL6
           MAPCOMR
R0        EQU  0
R1        EQU  1
R5        EQU  5
R15       EQU 15
           END
```