



CICS TS MustGather Activities

Mike Poil CICS Service
poilmike@uk.ibm.com



Introduction

- This presentation adds some additional notes to the sessions on debugging CICS TS for VSE/ESA by Bob Sodan of IBM.



Displaying CICS TS Partition Getvis Usage

- The GETVIS operator command shows usage e.g. GETVIS F2:

```
AR 0015 GETVIS USAGE      F2-24      F2-ANY      F2-24      F2-ANY
AR 0015  AREA SIZE:      11,260K    51,196K
AR 0015  USED AREA:       8,660K    37,428K MAX. EVER USED:  11,260K    40,132K
AR 0015  FREE AREA:       2,600K    13,768K LARGEST FREE:    2,572K    13,656K
```

- xx-24 is below 16MB, xx-ANY includes above **and** below 16MB; this is due to the way that GETVIS works (operand LOC=ANY means above or below).
- For CICS TS, the xx-24 MAX. EVER USED is always the same as the AREA SIZE because of the way that DSALIM is allocated; the customer should issue a GETVIS xx,RESET as soon as possible after CICS initialisation to get a representative high-water-mark.



DEBUG Overview

- The DEBUG operator command is used to activate a z/VSE system trace facility, which can cause up to a +20% cpu delta whilst active ("ON").
- It is useful for obscure problems and CICS loops.
- See the z/VSE Supervisor Diagnostic Reference Manual (DRM) for full details of how it works and what the trace entries contain.
- DEBUG ON,999K to start it.
- DEBUG OFF to suspend it (zero overhead), then DEBUG ON to resume.
- DEBUG END to terminate it.
- You actually get 3 trace tables used in cyclic form; a switch to the next trace table normally occurs on a CANCEL so that what led up to it is preserved; a switch also occurs by default when DEBUG OFF is executed.



DEBUG Overview

- For a CICS loop in partition xx, I recommend:
- `DEBUG ON,999K` to start it.
- Let it run for about 2 seconds then enter `DEBUG OFF` so that the loop trace data is frozen while the dump is taken to tape drive cuu as you don't want to trace your z/VSE system creating the dump!
- Dump - `SUSPEND xx, DUMP xx,0-7FFFFFFF,cuu, RESUME xx.`
- `DEBUG END`
- Don't forget to use `SUSPEND` before taking the dump otherwise you will be dumping a moving target, which will not help IBM to debug the problem.
- IBM has an internal product to browse and format a dump.



CICS Loop

```
status gl
AR 0015 S61-G1 EVA10MST 82 WAITING FOR I/O, OR ECB POSTING console subtask (low priority)
AR 0015      TCB=00349BBC TIB=00349B40 SAV=006052A0
AR 0015 S62-G1 DFHEVID2 82 WAITING FOR I/O, OR ECB POSTING auxiliary trace subtask
AR 0015      TCB=0035607C TIB=00356000 SAV=002CFC80
AR 0015 S63-G1 DFHEVID1 82 WAITING FOR I/O, OR ECB POSTING RO subtask
AR 0015      TCB=0035634C TIB=003562D0 SAV=002CFD00
AR 0015 S64-G1 DFHEVID1 83 READY TO RUN                QR subtask
AR 0015      TCB=0035661C TIB=003565A0 SAV=002CFD80
AR 0015 S65-G1 DFHEVID1 82 WAITING FOR I/O, OR ECB POSTING SL subtask
AR 0015      TCB=003568EC TIB=00356870 SAV=002CFE00
AR 0015 S66-G1 DFHEVID1 82 WAITING FOR I/O, OR ECB POSTING SO subtask
AR 0015      TCB=00356BBC TIB=00356B40 SAV=002CFE80
AR 0015 S67-G1 DFHEVID0 82 WAITING FOR I/O, OR ECB POSTING JCP open/close subtask
AR 0015      TCB=0036007C TIB=00360000 SAV=002CFF00
AR 0015 S68-G1 DFHSKTSK 82 WAITING FOR I/O, OR ECB POSTING file open/close subtask (high priority)
AR 0015      TCB=0036034C TIB=003602D0 SAV=002CFF80
AR 0015 M2D G1 COMSZCCA 82 WAITING FOR I/O, OR ECB POSTING main task (lowest priority)
AR 0015      TCB=002DA2E8 TIB=002DA268 SAV=00600000
AR 0015      SCB=002DA000 PCB=002DA088 COM=002DA4F0
```

- If shown, DFHIRPST is for MRO and a fifth DFHEVID1 is FEPI; the subtask order is normally in low-to-high priority order, except for DFHIRPST, which is lower than QR. A **lot** of activity can occur on DFHIRPST.



CICS Loop

- CICS in a loop - you can do a quick PSW display based on the task(s) that show a READY TO RUN state:

```
status g1
. .
AR 0015      TCB=0035634C TIB=003562D0 SAV=002CFD00      save area address
AR 0015 S64-G1 DFHEVID1 83 READY TO RUN                QR subtask
. . .
show g1,002CFD80.10
AR 0015      DATA  FOUND AT 002CFD80
V002CFD80 C4C6C8C5 E5C9C4F1 07DD1000 82C37EAC 06 *DFHEVID1....bC=.*
```

- The PSW is at +8 into the save area, the address is 02C37EAC.



SDAID for a Partition Loop

- SDAID initialisation JCL procedures are supplied in IJSYSRS.SYSLIB, and are easy to use; an example using the z/VSE console to initialise a branch trace having mounted a scratch tape at 380:

```
r rdr,pausebg
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1R88I  OK : 1 ENTRY PROCESSED BY R RDR,PAUSEBG
BG 0000 // JOB PAUSEBG
          DATE 10/05/2009, CLOCK 12/00/08
BG-0000 // PAUSE
0 // EXEC PROC=SDBRANCH,AREA=G1,OUTPUT=GREG,T=380
. . .
BG 0000 4C44I ENTER 'STARTSD' ATTENTION COMMAND TO ACTIVATE SDAID
BG 0000 EOJ PAUSEBG  MAX.RETURN CODE=0000
```

- Now use STARTSD, run for about 2 seconds, then STOPSD and ENDSO; send the raw file to IBM.
- Use PROC=SDINST to catch every instruction not just successful branches.



SDAID Overview

- Example DOSVSDMP formatted output:

START OF SDAID TRACE

```
BRANCH G1 G1 C00 B 47F0F016 02C39692=>02C396A8 PSW=47DD1000 82C396A8
GR 0-7 00000001 00354138 02B04310 82C376C0 02B04000 02C386BF 00000000 80692849
8-F 016FD000 02B04310 016FE0E4 00000001 02B04310 003540F0 00037F76 82C39692
BRANCH G1 G1 C00 BM 47403052 02C396EC=>02C396FC PSW=47DD1000 82C396FC
GR 0-7 40000000 80000000 016FD000 82C396AA 02B04000 00000038 00000001 02B00000
8-F 00354138 02B04310 016FE0E4 00000001 02B04310 003540F0 00037F76 82C39692
BRANCH G1 G1 C01 BZ 47803092 02C3971C=>02C3973C PSW=47DD0000 82C3973C
GR 0-7 40000000 016FD018 0000FFFF 82C396AA 02B04000 00000038 00000001 02B00000
8-F 00354138 02B04310 016FE0E4 00000001 02B04310 003540F0 00037F76 82C39692
BRANCH G1 G1 C01 BNM 47B030C2 02C39756=>02C3976C PSW=47DD0000 82C3976C
GR 0-7 40000000 016FD018 0000FFFF 82C396AA 00000000 02B00000 00000001 02B00000
8-F 00354138 02B04310 016FE0E4 00000001 02B04310 003540F0 00037F76 82C39692
BRANCH G1 G1 C01 BNP 47D0309A 02C39778=>02C39744 PSW=47DD1000 82C39744
GR 0-7 40000000 016FD018 00000001 82C396AA 00000001 02B00000 0000000A 02B00000
8-F 00354138 02B04310 016FE0E4 00000001 02B04310 003540F0 00037F76 82C39692
```

. . .



Using DEBUG for a CICS Task Abend

- PMR 14467,077,724 reported a CICS AKEB/020 abend on a z/OS DETACH (SVC 3E – see GA22-7588 for z/OS SVC numbers) for the Auxiliary Trace Subtask; it resulted in z/VSE APAR DY47081.
- DETACH should not produce an 020 abend, something like 23E was expected (see SA22-7606 for z/OS DETACH), so DEBUG should provide better diagnostics.
- The first slide after this one is for the problem, and shows how the IBM VSEDUMP tool was used to display the previous trace table as the CANCEL had forced a trace switch, then paged forward to show that the last SVC 3E was issued at address 07A18CA2 (PSWA 87A18CA4 - 2).
- The second slide shows the trace table switch due to the CANCEL, hence looking at the active trace table would have been of no use.
- The third slide shows the storage display at 07A18CA2; looking backwards you do not see CICS code, so I passed the problem on to VSE!

Using DEBUG for a CICS Task Abend

```
Oct 05          DEBUG Area Format - Address: 76C5000 (active - 1)          DEBUG01

NextFree: 076CAC60  NextCncl: 076D5000  AreaEnd: 076D4F60  Former: 076E5000
                                LCK
-Address-  ---- Type ----  TID RID CPU ----- Line 289 of 879 --
V.76CA240 Dispatch Exit  066 USR 0080 Task psw ..... 070D200087A18C1C *
V.76CA2A0 SVC 003E      066 USR 0080 SVC old psw ..... 070D200087A18C68 *
V.76CA2D0 SVC 003E      066 USR 0080 SVC old psw ..... 070D200087A18C68 *
V.76CA300 Dispatch Entry 066 DSP 0080 TIB: 003E6B40 Flgs: 30000010000000000002 *
V.76CA340 Dispatch Exit  066 USR 0080 Task psw ..... 070D200087A18C68 *
V.76CA3A0 SVC 003E      066 USR 0080 SVC old psw ..... 070D000087A18CA4 *
V.76CA3D0 SVC 003E      066 USR 0080 SVC old psw ..... 070D000087A18CA4 *
V.76CA400 Dispatch Entry 066 DSP 0080 TIB: 003E6B40 Flgs: 30000010000000000002 *
V.76CA440 Dispatch Exit  066 USR 0080 Task psw ..... 070D000087A18CA4 *
V.76CA4A0 SVC 0021      066 USR 0080 SVC old psw ..... 070D200087A18CE2 *
V.76CA4D0 Dispatch Entry 066 DSP 0080 TIB: 003E6B40 Flgs: 30000010000000000002 *
V.76CA510 Dispatch Exit  066 USR 0080 Task psw ..... 070D200087A18CE6 *
V.76CA570 SVC 0021      066 USR 0080 SVC old psw ..... 070D000087A1C416 *
V.76CA5A0 SVC 0027      066 USR 0080 SVC old psw ..... 070D200000108136 *
V.76CA5D0 User Data      066 REE 0080 USERDATA: 00000000000000000000C6E5C9E200000004 *
V.76CA610 Disp 64bit Reg 066 REE 0080 Reg 14: 8004C3E4      Reg 15: 8006F9E0 *
V.76CA6B0 Disp 64bit Reg 066 REE 0080 Reg 14: 000020D0      Reg 15: 8006FDB2 *
V.76CA750 Display Regs   066 REE 0080 Reg 14: 00085D40      Reg 15: 00000000 *

CMS ==
PFs: 1= Help      2= DataWind 3= Return  4= prevArea 5= nextArea 6= Latest
     7= Backward 8= Forward  9= Print  10= DeltaT  11= DumpArea 12= Exit *
```

Using DEBUG for a CICS Task Abend

```
Oct 05          DEBUG Area Format - Address: 76C5000 (active - 1)          DEBUG01

NextFree: 076CAC60  NextCncl: 076D5000  AreaEnd: 076D4F60  Former: 076E5000
                LCK
-Address-  ----  Type  ----  TID  RID  CPU  -----  Line 307 of 879  --
V.76CA7B0  User Data      066  REE  0080  USERDATA: 0000000000000000C6E5C9E200000004*
V.76CA7F0  User Data      066  REE  0080  USERDATA: 0000000000000000C6E5C9E200000004*
V.76CA830  Disp 64bit Reg  066  REE  0080  Reg 14: 8004C3E4      Reg 15: 8006F9E0  *
V.76CA8D0  Disp 64bit Reg  066  REE  0080  Reg 14: 000020D0      Reg 15: 8006FDB2  *
V.76CA970  Dispatch Entry  065  DSP  0080  TIB: 003E6870 Flgs: 00400000000000000002  *
V.76CA9B0  Display Regs    065  DSP  0080  Reg 14: 8000C92E      Reg 15: 0000D700  *
V.76CAA10  Dispatch Entry  064  DSP  0080  TIB: 003E65A0 Flgs: 00400000000000000002  *
V.76CAA50  Display Regs    064  DSP  0080  Reg 14: 8000C92E      Reg 15: 0000D700  *
V.76CAAB0  Dispatch Entry  F7   DSP  0080  TIB: 00057880 Flgs: 00400100000000000002  *
V.76CAAF0  Display Regs    F7   DSP  0080  Reg 14: 8000C92E      Reg 15: 0000D700  *
V.76CAB50  User Data      F7   REE  0080  USERDATA: 0000000000000000C6E5C9E200000004*
V.76CAB90  Display Regs    F7   REE  0080  Reg 14: 00060108      Reg 15: 00000000  *
V.76CABF0  Task Cancel     F7   REE  0080  TIB: 00057880 Flgs: 00000148000000000002  *
V.76CAC30  Debug Swap     F7   REE  0080  LTA: 5B5BC2E2E3C3D5D3  PTA: 5B5BC1C2C5D9E5*
                Next Free
V.76CAC80  Dispatch Entry  067  DSP  0080  TIB: 003F6000 Flgs: 30000010000000000002  *
V.76CACCO  Dispatch Exit   067  USR  0080  Task psw ..... 070D200000108130  *
V.76CAD20  SVC 0027       067  USR  0080  SVC old psw ..... 070D200000108136  *

CMS ==
PFs: 1= Help      2= DataWind 3= Return  4= prevArea 5= nextArea 6= Latest
      7= Backward 8= Forward 9= Print   10= DeltaT  11= DumpArea 12= Exit  *
```

Using DEBUG for a CICS Task Abend

```

Oct 05          CPPLUS - Version 6.05 <-> 91C SDUMP(X) 09/09/29          CPPLUS01

Address: 7A187A2          Base: 7A18CA2

--Address-- --Offset- ----- Hex ----- ----- Char -----
V..7A187A2   -500    00000000 00000000 00000000 00000000 | ..... | 04*
V..7A187B2   -4F0    00000000 00000000 00000000 0000FFFF | ..... | 04*
V..7A187C2   -4E0    07F05BE2 C3C10000 00000000 00000000 | .0$SCA..... | *
V..7A187D2   -4D0    00000000 00000000 00000000 0000C3C1 | .....CA | *
V..7A187E2   -4C0    E2C12100 08180018 C3C1E2C1 D7C7D4C1 | SA.....CASAPGMA | *
V..7A187F2   -4B0    40F640F1 40F1F161 F3F061F9 F5F147F0 | 6 1 11/30/951.0 | *
V..7A18802   -4A0    C0580700 47F0C058 F8F06B40 F1F9F9F5 | .....0..80, 1995 | *
V..7A18812   -490    40C3D6D4 D7E4E3C5 D940C1E2 E2D6C3C9 | COMPUTER ASSOCI | *
V..7A18822   -480    C1E3C5E2 40C9D5E3 C5D9D5C1 E3C9D6D5 | ATES INTERNATION | *
V..7A18832   -470    C1D340C9 D5C34B00 FFFF0000 05FE90EC | AL INC..... | *
V..7A18842   -460    D00C5840 10044100 00580A21 47F00AFF | ... ..0.. | 04*
V..7A18852   -450    18615850 612845B0 C54C45B0 C62045B0 | ././...E<..F.. | *
V..7A18862   -440    C51A5870 50F41E70 58707008 127747D0 | E....4..... | *
V..7A18872   -430    C096D503 C5F87010 4780C16A 1FFF0A21 | .oN.E8...A.... | E5*
V..7A18882   -420    47F00A04 1831D70B 305C305C D74730A4 | .0...P..*.*P..u | *
V..7A18892   -410    30A450D0 30604100 30A45000 30644100 | .u...-...u..... | *

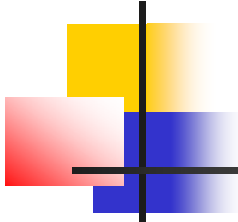
Data: ?          CPU id: 2094000006  Machine: ?  Mode: ?

CMS ==>
PFs: 1= Help    2= DataWind  3= Return   4= Dump Dvc  5= Scan     6= Recall
      7= Backward 8= Forward  9= Print    10= Locates 11= DSECT   12= Exit    *
  
```



Performance Gotchas

- Running the TMON Cross System Monitor (CSM) partition at a lower priority than CICS can cause intermittent poor CICS response times when more than 1 cpu is active. When prioritised incorrectly, the TMON CICS response times showed that using one cpu was actually faster than two!
- Monitoring software must **ALWAYS** be at a higher priority than what it is monitoring, that is unless the monitoring is imbedded like CICS monitoring!
- CICS statistics and monitoring do not account for cpu time spent in the DFHIRPST subtask and any other pure z/VSE subtask. Task IRIOWTT is likely to be "active" wait i.e. cpu time. Using ISC shows a similar profile.
- I have a modification to STAT that shows cpu time consumed by the whole partition, see the next slide.



Performance Gotchas

Dispatcher start time. . . : 02:08:34.93066
Peak tasks : 26
Current tasks. : 16
Current ICV time : 1,000ms
Current ICVR time. : 2,700,000ms
Current ICVTS time. : 100ms
Current PRTYAGING time . . : 500ms

Job step cpu time. : 00:01:55.10329

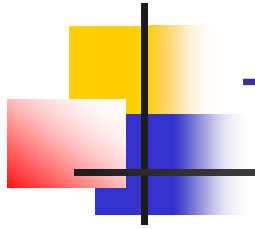
Number of active CICS TCBS : 4

TCB	TCB	TCB	Op. System	Op. System	TCB	TCB	DS TCB
Name	Status	Start Time	Waits	Wait Time	Dispatch Time	CPU Time	CPU Time
QR_SUBD	Active	02:08:34.93066	2,644	00:34:38.27545	00:02:14.07676	00:01:53.66400	00:00:00.0000
RO_SUBD	Active	02:08:35.27984	95	00:36:47.04238	00:00:04.96069	00:00:00.00000	00:00:00.0000
SO_MODE	Active	02:08:38.19514	2	00:36:49.08777	00:00:00.00002	00:00:00.00000	00:00:00.0000
SL_MODE	Active	02:08:38.16320	3	02:08:38.74004	00:00:00.57684	00:00:00.00000	00:00:00.0000



Performance Gotchas

- Function Shipping is horribly expensive, don't be surprised if it multiplies response times by a factor of 2 or more, this is WAD!
- MRO is much more efficient than ISC, although the customer may not notice much improvement in response times, but MRO really **does** use less cpu time.
- Avoid all unnecessary system and transaction dumps as they will stop CICS QR multitasking while they are being taken due to synchronous file I/O causing the z/VSE subtask to wait.
- Extrapartition transient data is the same due to its use of PUT, block data as much as possible.



The end

- Thank you