Analyzing CICS TS SOS Problems in z/VSE

Mike Poil
michaelalanpoil@gmail.com
Last updated: February 2022

z/VSE 5.2

http://www.ibm.com/zVSE
http://twitter.com/IBMzVSE

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Notice Regarding Specialty Engines (e.g., zIIPs, zAAPs and IFLs):

Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT").

No other workload processing is authorized for execution on an SE.

IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

References

- CICS TS for VSE/ESA Performance Guide SC33-1667
- CICS TS for VSE/ESA Problem Determination Guide SC33-1663
- CICS TS for VSE/ESA Trace Entries SC33-6108.
- CICS TS for VSE/ESA Enhancements Guide GC34-5763
- CICS TS for VSE/ESA Operations and Utilities Guide SC33-1654
- z/VSE LVC June 2013 How to handle or avoid CICS storage availability problems

Abstract

This session begins with a short introduction to the design of CICS storage management. It explains what triggers an SOS condition, how it affects CICS processing and introduces a potential workaround. It identifies possible reasons for SOS occurring and actions to resolve them. It explains what is required to be in place to capture the correct diagnostic data for problem determination and shows how to use a DFHPD410, DFHPD420 or DFHPD430 formatted dump output in conjunction with the appropriate CICS manuals to find the root cause and hence pursue the appropriate resolution.

Changes have been made since the LVC was first presented.

Agenda

- CICS DSA overview.
- CICS subpools.
- CICS GETMAIN/FREEMAIN storage requests.
- LE/VSE storage requests.
- What is Short-On-Storage?
- The SIT MXT parameter and SOS.
- Can I do anything to work around SOS?
- Could I have prevented SOS?
- What can cause SOS?
- What is a Storage Leak?
- How do I resolve SOS?
- What diagnostic material do I need?

Agenda

- What signs do I look for in a dump or statistics?
- Analyzing SOS problems.
- Problem #1.
- Problem #2.
- Problem #3.
- Q & A.
- Bonus material, including additional SOS analysis examples.

CICS DSA Overview

- Most of the CICS storage is managed through its DSAs allocated in Partition GETVIS.
- There are 4 types of DSA, with 24-bit (Below 16MB) and "E" 31-bit (Above 16MB) versions.
- The CDSA and ECDSA are for CICS-key storage requirements.
  - CICS control blocks.
  - Non-reentrant CICS nucleus programs.
  - Non-reentrant EXECKEY(CICS) programs.
  - CICS-key task storage.
  - CICS-key is the storage protection key of the Partition.
- The RDSA and ERDSA contain reentrant (SVA-eligible) programs when the SVA copies are not being used by CICS (SIT SVA=NO etc.).
  - CICS nucleus and other programs defined in the CSD.
  - SIT RENTPGM=PROTECT is recommended and uses storage key 0 protection for each phase's code rather than using the less secure CICS-key.
  - Note: Phase DFHSIP31 must be loaded into the SVA to protect it with key 0 storage and SIT SVA=YES is not required for the SVA copy to be used.

CICS DSA Overview

- The SDSA and ESDSA are for shared USER-key storage requirements.
  - EXEC CICS GETMAIN SHARED.
  - Non-reentrant EXECKEY(USER) programs.
  - Other data areas.
  - USER-key access, which is key 9 if SIT STGPROT=YES, and the protection key of the Partition if SIT STGPROT=NO.
  - The use of SIT STGPROT=YES is recommended.
- The UDSA and EUDSA are for non-shared USER-key storage requirements.
  - USER-key program task-related storage.
  - USER-key access, which is key 9 if SIT STGPROT=YES, and the protection key of the Partition if SIT STGPROT=NO.
- Note: Use SIT CMDPROT=YES to avoid USER-key program EXEC CICS commands overlaying CICS-key storage.

CICS DSA Overview

- SIT DSALIM (24-bit) and EDSALIM (31-bit) define the limits for DSA storage.
- The amounts are allocated from contiguous Partition GETVIS, and are mapped internally as a series of 256K and 1MB EXTENTs respectively that belong to the "available extent" pool.
- Note: the two GETVIS storage area remain allocated until CICS terminates even if the whole amounts are not allocated for DSA use.
- One or a series of contiguous extents are allocated to a DSA when it needs to expand.
- An extent normally remains allocated to a DSA even if it becomes empty, hence the amount of used (E)DSALIM will grow to a peak value based on concurrent demands.
- A DSA's size will contract if an empty extent is transferred to another DSA as part of SOS avoidance, but this will not affect peak (E)DSALIM usage.
- DSALIM and EDSALIM can be increased by CEMT I DSA, but only if there is contiguous free 24-bit or 31-bit Partition GETVIS storage respectively.
- Note: Over-committing Partition GETVIS use is a potentially fatal condition just like SOS.
- Note: You can't start a new CEMT task when CICS is at SOS.
- Note: Reducing then increasing (E)DSALIM by CEMT I DSA may return empty extents to the "available extent" pool, but YMMV!

CICS DSA Overview

- A DSA contains SUBPOOLs, each of which has a specific purpose.
- CICS System Subpools are documented in Appendix C of the CICS Performance Guide.
- Every task has 4 TASK subpools allocated for its (E)CDSA and (E)UDSA storage, and the subpool name begins with M for CDSA, C for EDSA, B for UDSA, U for EUDSA and followed by a 7-digit task number.
- A subpool is a series of 4K pages, but EUDSA (Unnnnnnn) task subpools use multiples of 64K.
- A storage request is mapped as an ELEMENT into one or more contiguous pages using First Fit logic, which can result in more fragmentation than Best Fit but requires less CPU time.
- Fragmentation will leave free storage elements that cannot be reused, which can cause the DSA to be much larger than the sum of the allocated elements, and even SOS.
- Some CICS system subpools are mapped as Quick-Cells that contain fixed-length elements of a CICS-determined size - Quick-Cells require much less cpu to manage.
- DSA usage is also a function of the way that programs are written and on configuration options being used for CICS and its resource definitions.

CICS Subpools

- Examples of 24-bit subpools:
  - LD subpool sizes are based on the number and size of programs that are currently *loaded* in storage; a large amount of storage would normally be a function of what you asked CICS to support.
    - LDNRS - 24-bit CICS key programs that are not SVA-eligible.
    - LDNUC - 24-bit CICS nucleus programs that are not SVA-eligible.
    - LDPGM - 24-bit USER key programs that are not SVA-eligible.
    - LDRES - 24-bit USER key RESIDENT programs that are not SVA-eligible.
    - LDNRSRO - 24-bit CICS key programs that are SVA-eligible.
    - LDNUCRO - 24-bit CICS nucleus programs that are SVA-eligible.
    - LDPGMRO - 24-bit USER key programs that are SVA-eligible.
    - LDRESRO - 24-bit USER key RESIDENT programs that are SVA-eligible.
  - SMSHRU24 - 24-bit GETMAIN SHARED.

CICS Subpools

- Examples of 31-bit subpools:
  - LDExxxxx Loader Domain subpools.
  - ARI0OLRM - used for DB2/VSE.
  - DFHTDG31 - Transient Data general storage and control blocks based on SIT TD=.
  - DFHTDIOB - Transient Data buffers based on SIT TD=.
  - JCDYNLOG - CICS Dynamic Log for backout of recoverable resources; the lifetime of this type of storage for a task is syncpoint-related.
  - SMTP - Terminal I/O areas based on the number of terminals and activity.
  - SMSHRU31 - 31-bit GETMAIN SHARED.
  - TSBUFFRS - Temporary Storage buffers based on SIT TS=.
  - TSGENRAL - Temporary Storage general usage based on SIT TS= and the DFHTEMP CISZ.
  - TSMAIN - Temporary Storage main storage areas based on usage.

CICS Subpools

- Special subpools:
  - KESTK24E and KESTK31E - every CICS reentrant module active in the task execution hierarchy needs an amount of "STACK" storage, and CICS may need to expand what was allocated at initialisation time in its "Stack Extension" subpools; they are allocated in Partition GETVIS and *not* DSA storage.
  - ZCTCTUA - TCTUA storage based on the number of terminals, which will exist in 24-bit or 31-bit subpools according to SIT TCTUALOC=BELOW|ANY, and the DSA used will depend on SIT TCTUAKEY=USER|CICS; I fixed one customer SOS Below by telling them to switch to TCTUALOC=ANY.
  - DL/I BBSSPn subpools will be allocated in 24-bit and/or 31-bit according to DLZACT TYPE=CONFIG PSBLOC= and HDMODE= options.
- The control blocks that are used by the CICS Storage Manager (SM) domain are allocated in Partition GETVIS-ANY to reduce the risk of them being accidentally overlaid.
- Other CICS control blocks may be allocated in Partition GETVIS and can be Fetch-Protected to cause 0C4 Program Check abends if the content is accessed in any way by User-key 9 programs.

CICS GETMAIN/FREEMAIN Storage Requests

- DSA storage is managed by CICS GETMAIN/FREEMAIN (DFHSMGF) services.
- Internal requests are used to support the operation of the CICS environment, the tasks and the programs using Control Blocks and data areas, allocating 24-bit or 31-bit based on the CICS design and by SIT parameter options like TCTUALOC=BELOW|ANY and by options like transaction definition TASKDATALOC(BELOW|ANY), and by program definition DATALOCATION(BELOW|ANY).
- Internal requests are used for LE storage allocations for managing tasks and programs, and LE either has fixed requirements or tells CICS how much and whether to allocate 24-bit or 31-bit storage or both for a program (an LE Run-Unit).
- In addition, EXEC CICS GETMAIN and FREEMAIN requests are made by programs and CICS exits.
- A GETMAIN request always needs contiguous storage and if that is not available, CICS must allocate a new DSA extent.
- When free DSA storage is not available for a request, the task might be suspended in an xDSA or ExDSA wait state and an SOS will normally occur.
- CICS or a program may decide to receive a NOSTG response instead and handle it.

CICS GETMAIN/FREEMAIN Storage Requests

- A transaction defined with a non-zero DTIMOUT value (in seconds) and SPURGE=YES will be purged if there is a long task xDSA or ExDSA wait.
- FREEMAIN normally frees the storage but using CSFE storage freeze means that storage is not freed until end-of-task.
- Task-related storage is automatically freed at end-of-task.
- GETMAIN SHARED storage must be explicitly freed.
- You will see most GETMAIN/FREEMAIN activity with SM level 1 trace active, with AP=1 and EI=1 or EI-1-2 showing EXEC CICS requests, including GETMAIN/FREEMAIN, and trace data will contain SM *EXC* trace entries when there is a problem.
- USE AP=3 to look at the content of task storage, which can provide a clue about its use when one task has a large amount of task-related storage GETMAINed.
- The following slides contain an example of an SOS Below problem caused by a CICS COBOL program using very bad embedded CEEUOPT LE run-time options - LE asked CICS for a large amount of 24-bit storage and SM0131 dump was created.

CICS GETMAIN/FREEMAIN Storage Requests

```
SM 0301 SMGF  ENTRY - FUNCTION(GETMAIN) GET_LENGTH(E178) SUSPEND(YES) REMARK(LE_RUWA) STORAGE_CLASS(TASK31)

              TASK-02689 KE_NUM-00EB TCB-00458000 RET-88CD8F98 TIME-17:46:32.4406490041 INTERVAL-00.0000007500    =055009=
                1-0000  00400000 0000000E 00000000 00000000  B6580000 00000000 01000101 00000000  *. ............................*
                  0020  00000000 00000000 0000E178 00000000  00000000 0100D3C5 6DD9E4E6 C1400700  *.....................LE_RUWA ..*

SM 0302 SMGF  EXIT  - FUNCTION(GETMAIN) RESPONSE(OK) ADDRESS(0A7D0008)

              TASK-02689 KE_NUM-00EB TCB-00458000 RET-88CD8F98 TIME-17:46:32.4406495041 INTERVAL-00.0000005000    =055010=
                1-0000  00400000 0000000E 00000000 00000000  B6580000 00000000 01000101 00000000  *. ............................*
                  0020  00000000 0A7D0008 0000E178 0000E180  00000000 0100D3C5 6DD9E4E6 C1400700  *.....'.................LE_RUWA ..*

SM 0301 SMGF  ENTRY - FUNCTION(GETMAIN) GET_LENGTH(2EEA8) SUSPEND(YES) REMARK(LE_RUWA) STORAGE_CLASS(TASK24)

              TASK-02689 KE_NUM-00EB TCB-00458000 RET-88CD8F98 TIME-17:46:32.4406497541 INTERVAL-00.0000002500    =055011=
                1-0000  00400000 0000000E 00000000 00000000  B6580000 00000000 01000101 00000000  *. ............................*
                  0020  00000000 0A7D0008 0002EEA8 0000E180  00000000 0100D3C5 6DD9E4E6 C1400600  *.....'.....y.........LE_RUWA ..*

SM 1206 SMPQ  *EXC* - Insufficient_storage_to_satisfy_request - FUNCTION(ALLOCATE_PAGEPOOL_STORAGE) SUBPOOL_TOKEN(0EFAC50C)
                   GET_LENGTH(2EEC0) SUSPEND(YES)

              TASK-02689 KE_NUM-00EB TCB-00458000 RET-8F588460 TIME-17:46:32.4406510041 INTERVAL-00.0000012500    =055012=
                1-0000  00380000 0000010A 00000000 00000000  BEF00000 00000000 019C0201 0EFAC50C  *.................0............E.*
                  0020  0002EEC0 03010000 09448B00 00000040  0EFBAE80 01020000                    *.............. ........        *
                2-0000  C2F0F0F0 F2F6F8F9                                                          *B0002689                        *
                3-0000  E4C4E2C1 40404040                                                          *UDSA                            *

SM 1001 SMSQ  ENTRY - FUNCTION(SUSPEND_REQUEST) GET_LENGTH(2EEC0) SUBPOOL_TOKEN(0EFAC50C) RETRY(NO)

              TASK-02689 KE_NUM-00EB TCB-00458000 RET-8F58A344 TIME-17:46:32.4406522541 INTERVAL-00.0000012500    =055013=
                1-0000  00300000 00000090 00000000 00000000  BC800000 00000000 029C0101 0002EEC0  *..............................*
                  0020  0EFAC50C 03010000 09448B00 02000040                                        *..E............                 *
```

CICS GETMAIN/FREEMAIN Storage Requests

```
DS 0004 DSSR  ENTRY - FUNCTION(RESUME) SUSPEND_TOKEN(003F0001)

              TASK-02689 KE_NUM-00EB TCB-00458000 RET-8F596E90 TIME-17:46:32.4406530041 INTERVAL-00.0000007500    =055014=
                1-0000  00600000 00000014 00000001 00000000  B4000000 00000000 05590172 88C3CC20  *.-.........................hC..*
                  0020  00000110 003F0001 09448D2C 00000030  00000000 00000000 A5C00000 00000000  *.......................v.......*
                  0040  01520201 00000003 0074E000 12060004  09448D2C 00000038 0EFAC50C 00000008  *.........................E.....*

. . The gap in trace entries covers the SM System Task being resumed to handle the storage availability issue

SM 080A SMSY  *EXC* - Short_on_storage_in_the UDSA

              TASK-SM    KE_NUM-001D TCB-00458000 RET-8F53D8F6 TIME-17:46:32.4406585041 INTERVAL-00.0000002500    =055031=
                1-0000  E4C4E2C1 40404040                                                *UDSA                           *
                2-0000  0002A000                                                        *....                           *
                3-0000  00010000                                                        *....                           *
                4-0000  00000001                                                        *....                           *
```

LE/VSE Storage Requests

- LE-enabled CICS programs require CICS to perform GETMAIN/FREEMAIN instead of using LE batch storage management services that use GETVIS/FREEVIS.
- LE requires storage for control blocks, STACK and HEAP storage, and AP=3 formatted task storage elements contain eye-catchers like "C3THDCOM", "LVT", "CEEPCB", "CEEEDB", "RUWP" and "HANC", with eye-catchers like "LE_TWA", "RUWAPOOL" and "LE_RUWA" in trace data.
- CICS LE run-time options must be defined to use 31-bit/ANYWHERE storage unless there are good technical reasons why that is not valid - failure to do that could cause excessive 24-bit DSA usage and potentially SOS Below.
- LE run-time options for CICS are normally taken from CEECOPT, but it is possible to create a CEEUOPT.OBJ that can be included after DFHELII when the CICS program is link-edited, when you will see eye-catcher "CEEOCB" near to the beginning of the phase.
- The RPTSTG run-time option will display individual program storage usage but should not be used in a Production CICS System except for a short time, when it can be activated by CLER - RPTSTG does not actively say if 24-bit or 31-bit storage is used.
- The next slide shows typical CICS LE run-time options as displayed by transaction CLER.

LE/VSE Storage Requests

```
CLER                                                                  09/22/21
                                                                      10:00:34
                  Language Environment for z/VSE V1 R4.10
              Dynamic CICS Run-time Option Modification Facility

ABPERC:    ( NONE  )              STORAGE:   ( 00    , NONE , NONE   , 0      )
ABTERMENC:( ABEND   )             TERMTHD:   ( TRACE  , MSGFL , 0    )
ALL31:     ( ON  )                TRAP:      ( ON   , MAX )
CBLPSHPOP:( ON  )                 USRHDLR:   (          )
CHECK:     ( OFF )           TERMTHDACT LSTQ OPTIONS :
DEBUG:     ( OFF )                CLASS: ( L )   DISPOSITION: ( D )
DEPTHCOND:( 10 )                  NODE-ID: (          )
ERRCOUNT:  ( 20 )                 USER-ID: (          )
HEAP:      ( 4080   , 4080   , ANYWHERE , KEEP , 4096   , 4080   )
LIBSTACK:  ( 4096   , 4080   , FREE )
MSGFILE:   ( CESE )               HEAPCHK:   ( OFF , 1   , 0  )
RPTOPTS:   ( OFF )
RPTSTG:    ( OFF )
STACK:     ( 4080   , 4080   , ANYWHERE , KEEP )

    Please change the runtime option to the setting required.

<PF3> Exit  <PF4> Restore Defaults  <PF5> Update  <PF12> Exit(No Change)
```

LE/VSE Storage Requests

Run the STAT transaction with bad "batch" CEEUOPT options resulting in excessive 24-bit usage:

```
CEEUOPT  CSECT
CEEUOPT  AMODE ANY
CEEUOPT  RMODE ANY
         CEEXOPT ALL31=(OFF),                                    X
               HEAP=(32K,32K,ANYWHERE,KEEP,8K,4K),               X
               STACK=(128K,128K,BELOW,KEEP),                     X
               RPTSTG=(ON)

Storage Report for Enclave DFH0STAT 09/22/21 7:12:51 AM
Language Environment for z/VSE V1 R4.10

    STACK statistics:
      Initial size:                                  131072 ← Allocated in 24-bit storage
      Increment size:                                131072
      Total stack storage used (sugg. initial size):  16072
      Number of segments allocated:                       0
      Number of segments freed:                           0
    LIBSTACK statistics:
      Initial size:                                    4096
      Increment size:                                  4080
      Total stack storage used (sugg. initial size):   7728
      Number of segments allocated:                       1
      Number of segments freed:                           0
```

LE/VSE Storage Requests

```
HEAP statistics:
      Initial size:                                  32768
      Increment size:                                32768
      Total heap storage used (sugg. initial size):   4208
      Successful Get Heap requests:                      1
      Successful Free Heap requests:                     0
      Number of segments allocated:                      0
      Number of segments freed:                          0
    ANYHEAP statistics:
      Initial size:                                  4080
      Increment size:                                4080
      Total heap storage used (sugg. initial size):  7336
      Successful Get Heap requests:                   137
      Successful Free Heap requests:                  114
      Number of segments allocated:                      1
      Number of segments freed:                          0
    BELOWHEAP statistics:
      Initial size:                                  4080
      Increment size:                                4080
      Total heap storage used (sugg. initial size):   888
      Successful Get Heap requests:                      4
      Successful Free Heap requests:                     1
      Number of segments allocated:                      0
      Number of segments freed:                          0
    Additional Heap statistics:
      Successful Create Heap requests:                   0
      Successful Discard Heap requests:                  0
      Total heap storage used:                           0
      Successful Get Heap requests:                      0
      Successful Free Heap requests:                     0
      Number of segments allocated:                      0
      Number of segments freed:                          0
End of Storage Report
```

What is Short-On-Storage (SOS)?

- DFHSM0131 or DFHSM0133 says that the free Below (24-bit) or Above (31-bit) storage is not enough for CICS to be confident that it can continue - CICS either hasn't the necessary free extents to fulfil a request or believes that it might need a free extent in the future and has none.
- DFHSM0132 or a DFHSM0134 shows that CICS recovered.
- To avoid SOS, CICS can move a free extent from one DSA to the one that needs it.
- Before it got to this state, CICS will have noticed that storage usage was beginning to show signs of "stress", and will have been proactively trying to free some of the unused storage in the hope of producing one or more empty extents:
  - Program Compression is used to progressively delete unused programs.
  - CICS will lower the priority of new tasks to reduce the possibility of them adding to the problem while helping older tasks to terminate more quickly and free their storage.
- At SOS, CICS will not allow new tasks to be attached.
- CICS may go in and out of SOS many times and/or hang.
- CICS Statistics reports on all SOS avoidance activity.

The SIT MXT parameter and SOS

- MXT determines how many user tasks can be under the control of the CICS Dispatcher at any one time, but CICS system tasks like TCP are not subject to MXT.
- At MXT, new tasks will be suspended in an MXT wait until other user tasks terminate.
- Setting MXT lower than you might want trades MXT wait time for SOS and may be the only way to resolve DSALIM issues, however, the aim should be to fix the underlying problem.
- The larger the MXT value, the more storage that can be used at any one time and the more likely that SOS will occur if DSALIM or EDSALIM are not sized to match it.
- If you want a simple way to size (E)DSALIM, try to use sets of CICS Statistics to get "normal" values for what you see being used:
  - Required DSALIM = (MXT/Normal Peak MXT) * Normal Peak DSALIM.
  - Do the same for EDSALIM.
- Note: A user transaction is subject to any TCLASS limit wait before the MXT limit.
- Note: If you PURGE or FORCEPURGE task that is in an MXT wait, it is put in a Deferred Purge state and will only be purged when MXT is resolved, and the task is dispatched. A task in a TCLASS wait can be purged unless it then goes into an MXT wait.

Can I get do anything to work around SOS?

- If you keep a CEMT task running permanently (strongly recommended), some action can be taken while CICS is in the SOS state.
- The main problem is that the root cause can be difficult to determine interactively because detailed storage usage information is only available when using ISV CICS tools like C\TREK (and that are already running in the CICS with SOS unless there is a working external interface) and understanding what to look for requires a lot of experience of the CICS internal design.
- Possible actions:
  - Using CEMT I DSA to add (E)DSALIM storage may provide relief.
  - Using CEMT I TA to purge one or more tasks may provide relief, but it does not identify the root cause - a task in (E)xDSA wait may be the victim.
  - Use CEMT I SYS to reduce MXT.

Could I have prevented SOS?

- Regularly track peak (E)DSALIM and MXT using DFH0STAT, DFHSTUP or an ISV product so that you know what is "normal".
- Trigger a warning when peak usage exceeds something like 80% of maximum, and when it triggers, try to identify the causes and make changes if that is appropriate.
- However, a rogue application change, a bug or an capacity issue can catch you by surprise.
- Beware of RELOAD(YES) programs that have more than one copy in storage - this was a known issue for at least one ISV product, and can be seen in DFHPD430 LD=1 output.
- The next three slides show DFH0STAT output produced *after* CICS had been SOS several times and a red highlight shows data values that may be of interest.
- In terms of the SOS conditions, we see the CICS system a long time after it happened and it does not appear to be under any stress now - statistics alone won't give us enough information to tell us what the root cause of SOS is.
- Adding 512K (the difference between DSALIM and the sum of 4 Peak DSA values) may be what is required to avoid SOS, but that would depend on having more than 512K in the GETVIS "largest free area", which we do not have.

Could SOS have been prevented?

```
Partition size established from ALLOC parameter . . :     122,879K

 Storage BELOW 16MB
 ──────────────────

   Partition GETVIS area size under 16 Mb  . . . . . :      11,260K
     Partition GETVIS used area below 16 Mb  . . . . . :     11,032K
     Partition GETVIS free area below 16 Mb  . . . . . :        228K
     Partition GETVIS maximum used below 16 Mb . . . . :     11,048K
     Partition GETVIS largest free area  below 16 Mb . :        216K ← this is slightly lower than I would recommend

     . . . continued on the next slide
```

Could SOS have been prevented?

```
  Current DSA Limit . . . . . :       9,216K  ← 9MB is bigger than many customers can allocate
  Current Allocation for DSAs . :     9,216K
  Peak Allocation for DSAs. . . :     9,216K
                                        CDSA          UDSA          SDSA          RDSA          Totals
  Current DSA Size. . . . . :         4,352K        2,560K        1,792K          512K          9,216K
  Current DSA Used. . . . . :         2,580K           28K        1,656K          456K          4,720K
  Current DSA Used as % of DSA. :        59%           1%            92%           89%             51% of DSA Size
* Peak DSA Used . . . . . :         4,144K        2,496K        1,744K          460K
  Peak DSA Size . . . . . :         4,352K        3,072K        1,792K          512K     ← The total is 9.5MB
  Cushion Size. . . . . :              64K           64K           64K           64K
  Free Storage (inc. Cushion) . :   1,772K        2,532K          136K           56K
* Peak Free Storage . . . . . :     1,812K        2,532K          256K          288K
* Lowest Free Storage . . . . . :     208K           64K           48K           52K
  Largest Free Area . . . . . :       256K          256K           48K           32K
  Largest Free Area as % of DSA :        5%           10%            2%            6%
  Largest Free/Free Storage . . :     0.14          0.10          0.35          0.57
  Current number of extents . . :       17            10             6             2                35
  Number of extents added . . . :       20            13             6             2
  Number of extents released. . :        3             3             0             0  ← SOS avoidance
  Getmain Requests. . . . . :     3,546,391    41,640,453           831            25
  Freemain Requests . . . . . :   3,545,994    41,640,445           727             3  ← beware big difference
  Current number of Subpools. . :       39            11             6             4                60
  Add Subpool Requests. . . . . :  293,675       293,647             6             4
  Delete Subpool Requests . . . :  293,636       293,636             0             0
  Times no storage returned . . :        0             0             0             0  ← > 0 means SOS was a problem
  Times request suspended . . . :       24            54             0             0  ← > 0 means SOS was a problem
  Current requests suspended. . :        0             0             0             0  ← > 0 means SOS was a problem
  Peak requests suspended . . . :        4             8             0             0  ← > 0 means SOS was a problem
  Requests purged while waiting :        0             2             0             0
  Times Cushion released. . . . :      349           215             0           832  ← Pre-SOS signal
  Times Short-On-Storage. . . . :        6            15             0            19  ← Times at SOS
ho   Total time Short-On-Storage . :  00:01:54.40254   00:00:50.97460   00:00:00.00000   00:02:02.41115
  Average Short-On-Storage time :  00:00:19.06708   00:00:03.39831   00:00:00.00000   00:00:06.44269
```

Could SOS have been prevented?

```
Loader

  Library Load requests. . . . . . . . . . . . . :          1,575  Total Program Uses . . . . . . . . . . . . . . :      1,678,077
  Total Library Load time. . . . . . . . . . . . :  00:00:23.25243  Program Use to Load Ratio. . . . . . . . . . . :          65.44
  Average Library Load time. . . . . . . . . . . :  00:00:00.01475
  Library Load requests that waited. . . . . . . :              2
  Total Library Load request wait time . . . . . :  00:00:00.03964
  Average Library Load request wait time . . . . :  00:00:00.01982
  Current Waiting Library Load requests. . . . . :              0
  Peak Waiting Library Load requests . . . . . . :              1
  Times at Peak. . . . . . . . . . . . . . . . . :              2  Average Not-In-Use program size. . . . . . . . :            23K
  CDSA                                                             ECDSA
  ____                                                             _____
  Programs Removed by compression. . . . . . . . :             30  Programs Removed by compression. . . . . . . . :              0
  Time on the Not-In-Use Queue . . . . . . . . . :  16:37:44.18565  Time on the Not-In-Use Queue . . . . . . . . . :  00:00:00.00000
  Average Time on the Not-In-Use Queue . . . . . :  00:04:55.17636  Average Time on the Not-In-Use Queue . . . . . :  00:00:00.00000
  Programs Reclaimed from the Not-In-Use Queue . :          1,184  Programs Reclaimed from the Not-In-Use Queue . :         20,886
  Programs Loaded - now on the Not-In-Use Queue. :              6  Programs Loaded - now on the Not-In-Use Queue. :             17
  SDSA                                                             ESDSA
  ____                                                             _____
  Programs Removed by compression. . . . . . . . :            298  Programs Removed by compression. . . . . . . . :              0
  Time on the Not-In-Use Queue . . . . . . . . . :  15:39:38.02778  Time on the Not-In-Use Queue . . . . . . . . . :  00:00:00.00000
  Average Time on the Not-In-Use Queue . . . . . :  00:00:41.41425  Average Time on the Not-In-Use Queue . . . . . :  00:00:00.00000
  Programs Reclaimed from the Not-In-Use Queue . :      1,230,968  Programs Reclaimed from the Not-In-Use Queue . :        334,720
  Programs Loaded - now on the Not-In-Use Queue. :              8  Programs Loaded - now on the Not-In-Use Queue. :            105
  RDSA                                                             ERDSA
  ____                                                             _____
  Programs Removed by compression. . . . . . . . :              3  Programs Removed by compression. . . . . . . . :              0
  Time on the Not-In-Use Queue . . . . . . . . . :  06:05:34.04108  Time on the Not-In-Use Queue . . . . . . . . . :  00:00:00.00000
  Average Time on the Not-In-Use Queue . . . . . :  02:01:51.34702  Average Time on the Not-In-Use Queue . . . . . :  00:00:00.00000
  Programs Reclaimed from the Not-In-Use Queue . :              0  Programs Reclaimed from the Not-In-Use Queue . :          2,599
  Programs Loaded - now on the Not-In-Use Queue. :              0  Programs Loaded - now on the Not-In-Use Queue. :             23
```

What can cause SOS?

- Inappropriate configuration options somewhere in the tier of products being used, for example, using SIT TCTUALOC=BELOW when ANY could be used.
- A workload that CICS cannot process fast enough, pushing MXT past its normal peak and increasing the storage requirements; perhaps CICS or even z/VSE does not have the CPU capacity to cope with the workload at that time; I have even seen a series of abends cause it because abend process is time-consuming.
- An unusual combination of task suspend (wait) states that cause a build up of tasks and their storage - this would also be reflected in a high current/peak MXT value.
- An application bug resulting in more storage being used than expected, for example, a loop or a poor design that results in a build up of VSAM file backout data, or a large GETMAIN; it could even be due to an application Storage Leak.
- Bugs in z/VM, z/VSE, ISV, CICS or other IBM products creating unexpected waits or a Storage Leak.
- A recent change somewhere, which could be anywhere in what is typically a very complex environment!
- The next slide shows an ISV bug causing a leak in DSA program-related storage.

What can cause SOS?

```
PROGRAM STORAGE MAP      (from DFHPD430 LD=1)

PGM NAME ENTRY PT CSECT  LOAD PT. COPY NO. USERS   LOCN TYP ATTRIBUTE

CAJOMON0 00893E80 -noheda- 00893E80  1    1    CDSA RPL RELOAD
CAJOJOBA 008C2338 -noheda- 008C0000  1    0    CDSA RPL RESIDENT
CAJUUTL0 008CD000 -noheda- 008CD000  1    1    CDSA RPL RELOAD
CAJOEDT0 008DF360 -noheda- 008DF000  1    0    CDSA RPL RESIDENT
CASOEDT0 008E5280 -noheda- 008E5280  1    0    CDSA RPL RESIDENT
CAJUUTL0 008EB000 -noheda- 008EB000  2    1    CDSA RPL RELOAD       2 copies in 24-bit storage! There is a fix for this problem.
CAJUUTL0 00900000 -noheda- 00900000  3    1    CDSA RPL RELOAD
CAJUUTL0 00911880 -noheda- 00911880  4    1    CDSA RPL RELOAD
CAJUUTL0 00923100 -noheda- 00923100  5    1    CDSA RPL RELOAD
CAYDPTCH 0093D8A0 DFHYA411 0093D8A0  1    1    CDSA RPL REUSABLE
CAJOMON0 00940000 -noheda- 00940000  13   1    CDSA RPL RELOAD
CAJUUTL0 00980000 -noheda- 00980000  6    1    CDSA RPL RELOAD
CAJUUTL0 00991880 -noheda- 00991880  7    1    CDSA RPL RELOAD
CAJUUTL0 009A3100 -noheda- 009A3100  8    1    CDSA RPL RELOAD
CAJOMON0 009C0000 -noheda- 009C0000  14   1    CDSA RPL RELOAD
CAJOMON0 00A0B770 -noheda- 00A0B770  2    1    CDSA RPL RELOAD
CAJOMON0 00A40000 -noheda- 00A40000  3    1    CDSA RPL RELOAD
CAJOMON0 00A5BDF0 -noheda- 00A5BDF0  4    1    CDSA RPL RELOAD
CAJOMON0 00A80000 -noheda- 00A80000  5    1    CDSA RPL RELOAD
CAJOMON0 00A9BDF0 -noheda- 00A9BDF0  6    1    CDSA RPL RELOAD
CAJOMON0 00AC0000 -noheda- 00AC0000  7    1    CDSA RPL RELOAD
CAJOMON0 00ADBDF0 -noheda- 00ADBDF0  8    1    CDSA RPL RELOAD
CAJOMON0 00B00000 -noheda- 00B00000  9    1    CDSA RPL RELOAD
CAJOMON0 00B1BDF0 -noheda- 00B1BDF0  10   1    CDSA RPL RELOAD
CAJOMON0 00B40000 -noheda- 00B40000  11   1    CDSA RPL RELOAD
CAJOMON0 00B5BDF0 -noheda- 00B5BDF0  12   1    CDSA RPL RELOAD
```

What is a Storage Leak?

- A leak will cause a continuous build up in one or more related types of storage due to unpaired GETMAINs and FREEMAINs.
- However, a large amount of a particular type of storage may be "normal" and not a leak.
- Given enough time, it will result in an SOS or similar out-of-storage condition, but CICS may not run for long enough for a problem to be seen.
- Adding storage to cover the leak will just allow CICS to run longer than before.
- Fragmentation due to the sizes and patterns of GETMAIN/FREEMAIN may look like a leak and will result in a large peak DSA usage combined with an unexpected amount of free storage - this is also known as "Storage Creep" .
- Note: Using EXCI batch jobs will cause a leak in 31-bit z/VSE System Getvis subpool IXXPDC until the CICS system is shut down or CEMT S IRC CLOSE and OPEN is performed if the PTF for APAR PH06333 for CICS TS for z/VSE 2.2.

How do I resolve SOS?

- It depends on what you will find as the root cause, and could include one or more of:
  - Find fixes for IBM or ISV software.
  - Fix the application program(s).
  - Increase DSALIM and/or EDSALIM and re-configure the partition and/or z/VSE as required.
  - Decrease MXT.
  - Use TCLASS to limit transactions that use a lot of resource to a maximum number.
  - Tune and/or change configuration options in ISV products, CICS, z/VSE or z/VM.
  - Provide extra cpu or storage capacity and/or faster dasd.
- If that fails, involve CICS Support.

What diagnostic material is needed?

- Allocate a trace table of at least 4MB with SIT TRTABSZ=4096 and use SIT STNTR=1, but CICS Support may ask for extra trace levels to be set if the problem must be recreated.
- An SM0131 or SM0133 dump *at* the time CICS is SOS - on every COLD start, use CEMT S SYD(SM0131) ADD SYS MAX(1) and CEMT S SYD(SM0133) ADD SYS MAX(1), or a use PLTPI program to add both system dump codes.
- A dump after CICS has recovered from SOS often cannot identify the root cause.
- If CICS is hung after SOS and there is no SM013n dump, either use CANCEL xx to create a CICS system dump, or create a *synchronous* AR DUMP:
  - SUSPEND xx
  - DUMP xx,0-7FFFFFFF,cuu
  - RESUME xx
  - CANCEL xx,NODUMP
- If *you* would like to analyse the problem, run INFOANA with:

  CALL DFHPD430 DATA AP=3,DS=1,KE=1,LD=1,SM=1,TR=3,XM=1

What diagnostic material is needed?

- If the analysis indicates that there might be a performance problem, it may help to have detailed task performance data for half an hour or more leading up to the SOS.
- It is also helpful to know what is "normal" for the transactions that appear in the task performance data so that you can identify when there are problems and what they are.
- It may require z/VSE and even z/VM performance data.
- If CICS Support is to be involved:
  - Open a Case against CICS and add background information.
  - FTP the dump library member in *binary*, with CICS SYSLST and PRINTLOG that includes all related messages in *ASCII* (this applies to any CICS problem).
  - Don't FTP DFHPD430 output unless specifically because CICS Support will format the dump as required - the underlying problem may not be visible in storage that can be formatted and require the dump library member to be browsed.
  - CICS Support will provide FTP instructions.
  - Please *do not* send file formats that you can't look at with a product like Microsoft Word or Open Office.

What signs are available in a dump or statistics?

- The obvious signs are in SM=1 output, with a DSA flagged as SOS, but this may be the victim and not the culprit.
- SM=1 output and/or statistics shows the number of NOSTG responses, the number of suspended requests and the number of times DSA cushions were released.
  - Each DSA has a "Storage Cushion", which is the minimum amount of contiguous storage that CICS should keep to avoid SOS.
  - When CICS uses some of this, it is a warning of a potential SOS.
  - The cushion sizes are 64K for the xDSAs, 0K for the EUDSA, 256K for the ERDSA and 128K for the ECDSA and ESDSA.
- The end of the SM=1 domain dump output will show any tasks waiting for xDSA or ExDSA storage and how much they requested.
- The DS=1 domain output may not show a DSA wait, because SMSY may be trying to free storage in case the failing GETMAIN can be retried.
- Loader Domain statistics provide counts for programs removed by Program Compression.

Analyzing SOS Problems

- I will use examples that are like reported problems, but with a reduced amount of DFHPD430 output to provide an idea of what to look for.
- This may be an unusual number of task waits, for example, many tasks in FCCIWAIT for a file, which means they are all waiting for one CI split to complete, or LMQUEUE waits waiting for locked resources; when there is a slowdown, it will often result in more tasks running concurrently, pushing up peak MXT and total storage usage.
- It could be an abnormal amount of storage allocated to a CICS subpool, which could also be due to a slowdown, an application bug or even a storage leak.
- Knowing what is "normal" for your system will help.
- Note: Being at SOS does not mean that there is no storage available in CICS.
- Note: CICS Service have seen many SOS problems and can often spot a potential problem quickly, however, there could be some guesswork as they don't know what is "normal" for this CICS system.

Problem #1

- SOS Above occurred several times, CICS recovered, and an SM0133 dump was produced.
- SOS has not occurred before in this CICS system.
- No changes were made recently.
- This is a typical symptom of a capacity issue, but is that what it is?
- XM=1 output shows that CICS is close to MXT.
- The typical peak MXT for this CICS is approximately 230 compared to the 295 here.

```
==XM: MXT SUMMARY
   Maximum user tasks (MXT):          310
   System currently at MXT:           No
   Current active user tasks:         295
   Current queued user tasks:         0
 * Peak active user tasks:            295
 * Peak queued user tasks:            0
 * Times at MXT limit:                0
```

- DFHPD430 XM=1 output shows task information and any TCLASS/MXT waits.

Problem #1

- DS=1 is a snapshot of task activity at the time of the dump, but can provide an insight into what was happening before SOS occurred and possibly the root cause.
- Each task that can be dispatched has a state, and you normally see one of three values:
  - "R" for Running - the task is being dispatched by CICS; in an SM013n dump this will be SMSY. (The SM CICS system task SMSY is used to look for and deal with SOS, and normally runs every 3 minutes to check storage conditions, but every 2 seconds or less when CICS is under stress).
  - "D" for Dispatchable - this is an implied wait for access to the cpu, and seeing many of them waiting for QR could indicate a cpu availability problem
  - "S" means the task is in a Suspend (wait) state, optionally with a purge timeout.
- The CICS Problem Determination Guide Chapter 6 describes each "S" state.
- It may show a "normal" wait, e.g. FCIOWAIT is when you are waiting for VSAM I/O to complete (but it should be for a short duration), or an ICWAIT for "n" seconds.
- It may show contention, e.g. FCCIWAIT says that the task's VSAM I/O is being delayed because an active I/O is performing a CI (and possibly a CA) split; only when this is finished does this task get a chance to retry its VSAM request.

Problem #1

- DS=1 shows CICS System Tasks in normal waits with the SM System Task (AD=SM) running as expected during SOS; the dump time was 12:56:04.
- There are many FCPSWAITs for file MAST001, which is the FILE definition STRINGS wait (not an LSRPOOL string wait); this could be a bad choice for STRINGS or be due to a slowdown stopping them being released fast enough for another task to use.
- There are a small number of FCIOWAITs - the one below shows a 2-second I/O wait!
- No other significant types of wait and only there were only a small number of tasks in a "D" status, so there is no clear sign of a cpu problem, but it is just a snapshot.

```
KEY FOR SUMMARY
    T  = TYPE OF TASK            S=SYSTEM  N=NON-SYSTEM
    S  = STATE OF TASK           D=DISPATCHABLE  S=SUSPENDED  R=RUNNING  E=RESUMED EARLY
. . .
DS_TOKEN KE_TASK  T S F P TT RESOURCE RESOURCE_NAME    W TIME OF      TIMEOUT          AD      XM_TXN_TOKEN
                             TYPE                        SUSPEND      DUE                              (task#)
00940001 03F1D080 S R                                                                 SM
0112157D 046D8B00 N S P N -  FCPSWAIT MAST001          C 12:56:02.210      -         XM      06D2550000058761C
02161329 0466DB00 N S N N -  FCIOWAIT MAST001          W 12:56:02.216      -         XM      06D2550000058773C
```

Problem #1

- SM=1 summarises the status of CICS DSA storage.
- CICS is at SOS above with no EDSALIM expansion possible.
- 1MB of DSALIM expansion is available (4 * 256K extents).
- Typical peak EDSALIM usage for this CICS is 90MB.

```
SM Domain status:                    INITIALISED
   Storage recovery:                    YES
   Storage protection requested:        YES
   Storage protection active:           YES
   Reentrant program option:            PROTECT
   Current DSA limit:                      9216K
   Current DSA total:                      8192K
   Currently SOS below 16M:                  NO
   Current EDSA limit:                      110M
   Current EDSA total:                      110M
   Currently SOS above 16M:                  YES
```

Problem #1

- The ECDSA large and is close to SOS, with only the cushion of contiguous storage left.

```
==SM: ECDSA Summary
   Size:                          40960K
   Cushion size:                    128K
   Current free space:             1112K  ( 2%)    ← some fragmentation
 * Lwm free space:                    68K  ( 0%)
 * Hwm free space:                  2212K  ( 5%)
   Largest free area:               128K           ← at cushion size
 * Times nostg returned:               0
 * Times request suspended:            0
   Current suspended:                  0
 * Hwm suspended:                      0
 * Times cushion released:             0
   Currently SOS:                     NO
 * Times went SOS:                     0
 * Time at SOS:              00:00:00.000
```

Problem #1

- The EUDSA is large and is at SOS, with one suspended GETMAIN request.

```
==SM: EUDSA Summary
   Size:                          63488K
   Cushion size:                      0K
   Current free space:              128K  ( 0%)    ← no fragmentation
 * Lwm free space:                  128K  ( 0%)
 * Hwm free space:                 1152K  ( 1%)
   Largest free area:               128K           ← maximum contiguous storage
 * Times nostg returned:              0
 * Times request suspended:           1
   Current suspended:                 1
 * Hwm suspended:                     1
 * Times cushion released:            0
   Currently SOS:                   YES
 * Times went SOS:                    1
 * Time at SOS:             00:00:00.000
```

Problem #1

- Although not relevant to this SOS analysis, SM=1 provides an extent summary.
- The EUDSA has 57 extents and only the last one has free storage, which shows no obvious signs of fragmentation.

```
Number of extents:                  57

Extent list:      Start      End          Size     Free

                  06300000   063FFFFF     1024K      0K
                  06800000   069FFFFF     2048K      0K
                  . . .
                  0A800000   0A8FFFFF     1024K      0K
                  0A900000   0A9FFFFF     1024K      0K
                  0AA00000   0AAFFFFF     1024K      0K
                  0AB00000   0ABFFFFF     1024K      0K
                  0AC00000   0ACFFFFF     1024K    128K
```

Problem #1

```
==SM: ESDSA Summary
   Size:                          4096K      ← insignificant
   Cushion size:                   128K
   Current free space:            2620K  (63%)
 * Lwm free space:                 336K  ( 8%)
 * Hwm free space:                2620K  (63%)
   Largest free area:              772K
 * Times nostg returned:             0
 * Times request suspended:          0
   Current suspended:                0
 * Hwm suspended:                    0
 * Times cushion released:           0
   Currently SOS:                   NO
 * Times went SOS:                   0
 * Time at SOS:            00:00:00.000
```

Problem #1

```
==SM: ERDSA Summary
   Size:                          4096K      ← insignificant
   Cushion size:                   256K
   Current free space:             312K  ( 7%)
 * Lwm free space:                 312K  ( 7%)
 * Hwm free space:                2628K  (64%)
   Largest free area:              312K
 * Times nostg returned:             0
 * Times request suspended:          0
   Current suspended:                0
 * Hwm suspended:                    0
 * Times cushion released:           0
   Currently SOS:                   NO
 * Times went SOS:                   0
 * Time at SOS:           00:00:00.000
```

Problem #1

- An analysis of the task subpools showed that tasks 38 and 141 use a lot of storage - task 38 had 36 more GETMAINs than FREEMAINs but task 41 had only 5; the XM domain has the transaction ids and the usage will need to be validated.
- You can see the difference between element storage (actual usage) and the page storage 4K or 64K multiples suballocated from the DSA extent storage.
- Other user tasks show EUDSA usage from 64K to 320K.

```
==SM: Task subpool summary
   Current number of tasks:     303       ← includes system tasks

   SMX Addr Name     Id Loc Acc   Gets   Frees  Elems  Elemstg Pagestg
    . . .
   03E8A2C4 M0000038 01  B   C       1       1      0        0      0K  (CDSA)
            C0000038 03  A   C       0       0      0        0      0K  (ECDSA)
            B0000038 02  B   U       6       4      2     1584      4K  (UDSA)
            U0000038 04  A   U     130      94     36  1432128   1472K  (EUDSA)
            (task 38)                                 (1398.5K)
```

Problem #1

```
03E8A498 M0000141 01   B    C        0       0       0         0       0K
         C0000141 03   A    C        1       0       1        48       4K
         B0000141 02   B    U      520     518       2      9056      12K
         U0000141 04   A    U    62317   62312       5   5085616    5056K
. . .
03E8DB3C M0058770 01   B    C        0       0       0         0       0K
         C0058770 03   A    C        0       0       0         0       0K
         B0058770 02   B    U        1       1       0         0       0K
         U0058770 04   A    U        2       0       2     15680      64K
. . .      (58770 is having the issue with EUDSA storage)
03E8BE64 M0085812 01   B    C        0       0       0         0       0K
         C0085812 03   A    C        0       0       0         0       0K
         B0085812 02   B    U       18      16       2      9056      12K
         U0085812 04   A    U        8       5       3    151632     256K
      (task 85812 is an example of a "normal" task in terms of EUDSA usage)
```

Problem #1

- The CICS subpools show that DFHTDG31 (Transient Data general storage) and DFHTDIOB (Transient Data I/O buffers) use more than 50% of the 40MB of ECDSA storage.
- This could be "normal", but the SIT TD= parameter may be over-allocated.
- No other CICS subpool usage is significant.

```
==SM: Domain subpool summary (ECDSA)
   Name      Id Chn  Initf Bndry Fxlen Q-c    Gets   Frees   Elems  Elemstg Pagestg
   AITM_TAB  B3          4K     8   592  Y       60       0      60    35520     40K
   AP_AFCTE  C7  Y       4K    16                286       0     286     9968     12K
   AP_TCA31  4D        128K   128  1536  Y    27032   26750     282   433152    564K
   . . .
   DFHTDG31  81                16                6006       0    6006   578544    568K
   DFHTDIOB  84                16                   1       0       1 24576000  24000K  (23MB+)
   DFHTDWCB  85          4K    16    64  Y    77707   77707       0        0       4K
```

Problem #1

- DSA waits are shown at the end of SM=1.
- A request for 180912 bytes (177K), which will not fit in the 128K of contiguous storage that is available, and will probably require 192K or 3 * 64K.
- If 192K was added to the existing 64K for task 58770, it would result in 256K, which is a size often seen for transactions in this CICS system and appears to be reasonable.

```
==SM: Suspend queue summary

   KE Task  Tran #  Susptok  Subpool  DSA       Request

   04416780 0058770 0416139B U0058770 EUDSA      180912
```

Problem #1

- The trace shows more than 1,000 FCIOWAITs in a short time - CICS is busy!
- A Rexx program analyzed the trace and showed that CICS was dispatched more than 85% of the time, a level of activity that could be causing a slowdown.
- These trace entries lead up to the SOS.
- The X'2C298' byte request is for LE RUWA (LE control blocks and initial storage allocations) 31-bit working storage, and CICS was looking for additional 64K subpool pages.

```
AP 1940 APLI  ENTRY - FUNCTION(START_PROGRAM) PROGRAM(NPG00001) CEDF_STATUS(CEDF) EXECUTION_SET(FULLAPI) ENVIRONMENT_TYPE(EXEC)
                    SYNCONRETURN(NO) LANGUAGE_BLOCK(06791648) COMMAREA(00000000 , 00000000) LINK_LEVEL(1) SYSEIB_REQUEST(NO)

              TASK-58770 KE_NUM-0020 TCB-004E5000 RET-8D37F904 TIME-12:56:02.2605341901 INTERVAL-00.0000003125    =098282=
                1-0000  00580000 000000DA 00000000 00000000  B81B5D00 00000000 02680100 D6E5E2C2  *.................).........OVSB*
                  0020  E4F2F040 064EA730 B5000000 064EA730  01000001 01F10202 07F11D20 06791648  *U20 .+x......+x......1...1.....*
                  0040  00003928 00000000 00000000 00000000  00010002 02000000                    *.....................       *

SM 0301 SMGF  ENTRY - FUNCTION(GETMAIN) GET_LENGTH(2C298) SUSPEND(YES) REMARK(RUWAPOOL) STORAGE_CLASS(TASK31)

              TASK-58770 KE_NUM-0020 TCB-004E5000 RET-8D3187B8 TIME-12:56:02.2605344401 INTERVAL-00.0000002500    =098283=
                1-0000  00400000 0000000E 00000000 00000000  B6580000 00000000 01400102 00000000  *. ..................... ......*
                  0020  00000000 19400000 0002C298 00010058  00000000 0100D9E4 E6C1D7D6 D6D307C0  *..... ....Bq.........RUWAPOOL..*
```

Problem #1

```
SM 1206 SMPQ  *EXC* - Insufficient_storage_to_satisfy_request - FUNCTION(ALLOCATE_PAGEPOOL_STORAGE) SUBPOOL_TOKEN(03089E30)
                   GET_LENGTH(2C2B0) SUSPEND(YES)

              TASK-58770 KE_NUM-0020 TCB-004E5000 RET-8B8D4714 TIME-12:56:02.2605378776 INTERVAL-00.0000034375    =098284=
                1-0000  00380000 0000010A 00000000 00000000  BEF00000 00000000 01000201 03089E30  *.................0.............*
                  0020  0002C2B0 03012332 0441DF10 00000040  0B882284 018822B4                    *..B.......... .h.d.h..       *
                2-0000  E4F0F0F5 F8F7F7F0                                                          *U0058770                     *
                3-0000  C5E4C4E2 C1404040                                                          *EUDSA                        *

SM 1001 SMSQ  ENTRY - FUNCTION(SUSPEND_REQUEST) GET_LENGTH(2C2B0) SUBPOOL_TOKEN(03089E30) RETRY(NO)

              TASK-58770 KE_NUM-0020 TCB-004E5000 RET-8B8D65D6 TIME-12:56:02.2605390026 INTERVAL-00.0000011250    =098285=
                1-0000  00300000 00000090 00000000 00000000  BC800000 00000000 02000101 0002C2B0  *.............................B.*
                  0020  03089E30 03012332 0441DF10 02000040                                        *.............                 *
. . . . .
SM 080A SMSY  *EXC* - Short_on_storage_in_the EUDSA

              TASK-SM    KE_NUM-001D TCB-004E5000 RET-8B88A816 TIME-12:56:02.2605576276 INTERVAL-00.0000002500    =098321=
                1-0000  C5E4C4E2 C1404040                                                          *EUDSA                        *
                2-0000  00020000                                                                  *....                         *
                3-0000  00000000                                                                  *....                         *
                4-0000  00000001                                                                  *....                         *
```

Problem #1 Conclusion

- The large task and TD storage usage was "normal".
- Nothing else was observed to be "abnormal" in this CICS system.
- No changes were made before the SOS occurred.
- SOS is not a common problem for the customer.
- This is a match for a short-term capacity problem.
- One option is to increase EDSALIM.
  - The typical peak MXT is 230 and peak EDSALIM usage is 90MB, therefore the suggested new EDSALIM value is (310/230) * 90MB = 121MB.
  - With 1MB still available in DSALIM, SOS below is unlikely to occur.
- Another option would be to reduce MXT to a bit less than 295.
- However, I would suggest a review of the performance of the z/VSE and CICS system to ensure that it has the capacity to handle the workload during times of peak load with appropriate response times.

Problem #2

- DSA usage grows quickly in size until an SOS below occurs and CICS does not recover.
- This happens every time CICS is started.
- This is the classic symptom of a Storage Leak, and should be easy to diagnose.
- Being at MXT may or may not be significant.

```
==XM: MXT SUMMARY

   Maximum user tasks (MXT):              60
   System currently at MXT:               Yes
   Current active user tasks:             60
   Current queued user tasks:             3
 * Peak active user tasks:                60
 * Peak queued user tasks:                6
 * Times at MXT limit:                    4
```

- DS=1 shows no obvious sign of any problems, so I will not include any output from it.

Problem #2

- SM=1 shows SOS Below.

```
===SM: STORAGE MANAGER DOMAIN - SUMMARY

   SM Domain status:                INITIALISED
   Storage recovery:                NO
   Storage protection requested:    YES
   Storage protection active:       YES
   Reentrant program option:        PROTECT
   Current DSA limit:                  7424K
   Current DSA total:                  7424K
   Currently SOS below 16M:             YES

   Current EDSA limit:                  55M
   Current EDSA total:                  40M
   Currently SOS above 16M:              NO
```

Problem #2

```
==SM: UDSA Summary

   Size:                          512K
   Cushion size:                   64K
   Current free space:            212K  (41%)
 * Lwm free space:                140K  (27%)
 * Hwm free space:                496K  (96%)
   Largest free area:             152K
 * Times nostg returned:             0
 * Times request suspended:          0
   Current suspended:                0
 * Hwm suspended:                    0
 * Times cushion released:           0
   Currently SOS:                   NO
 * Times went SOS:                   0
```

Problem #2

```
==SM: CDSA Summary

   Size:                        1280K
   Cushion size:                  64K
   Current free space:            688K  (53%)
 * Lwm free space:               120K  ( 9%)
 * Hwm free space:               688K  (53%)
   Largest free area:             240K
 * Times nostg returned:             0
 * Times request suspended:          0
   Current suspended:                 0
 * Hwm suspended:                     0
 * Times cushion released:            0
   Currently SOS:                    NO
 * Times went SOS:                    0
```

Problem #2

```
==SM: SDSA Summary

   Size:                          5120K      ← the biggest usage
   Cushion size:                    64K
   Current free space:              60K  ( 1%)
 * Lwm free space:                  60K  ( 1%)
 * Hwm free space:                 256K  ( 5%)
   Largest free area:               60K
 * Times nostg returned:             0
 * Times request suspended:          0
   Current suspended:                 0
 * Hwm suspended:                     0
 * Times cushion released:            8
   Currently SOS:                   YES
 * Times went SOS:                    1
 * Time at SOS:           00:00:00.000
```

Problem #2

```
==SM: RDSA Summary

   Size:                          512K
   Cushion size:                   64K
   Current free space:            216K  (42%)
 * Lwm free space:                 216K  (42%)
 * Hwm free space:                 292K  (57%)
   Largest free area:             216K
 * Times nostg returned:             0
 * Times request suspended:          0
   Current suspended:                0
 * Hwm suspended:                    0
 * Times cushion released:           0
   Currently SOS:                   NO
 * Times went SOS:                   0
 * Time at SOS:           00:00:00.000
```

Problem #2

- The issue is related to the size of the SDSA and GETMAIN SHARED actvity.
- Look at the big difference between Gets and Free requests and the number of elements, and is a typical sign of a leak!
- GETMAIN SHARED has no task-related information maintained, so who is requesting it?

```
==SM: Domain subpool summary (SDSA)
   Name      Id Chn  Initf Bndry Fxlen Q-c    Gets   Frees   Elems  Elemstg Pagestg
   APECA     5D                 8     8          1       1       0        0      0K
   DFHAPU24  46                16                1       0       1     3584      4K
   LDPGM     28                16               20      14       6   100128    108K
   LDRES     24                16                1       0       1    23952     24K
   SMSHRU24  60  Y             16            14087       2   14085  5032704   4924K
```

Problem #2

- Perhaps there is some evidence in the trace, if not, auxtrace might be an option.
- There are more than 300 abbreviated trace entries for GETMAIN SHARED that are identical apart from the task number.
- This one shows the EXEC CICS request for task 14656, it was successful and returned the address EAFF40.

```
14656 1 AP 00E1 EIP   ENTRY GETMAIN                                  0004,05F95B08 .9$.,09000C02 ....          =174462=
. . .
14656 1 SM 0C01 SMMG  ENTRY GETMAIN            154,NO,SHARED_USER24,EXEC                                       =174465=
. . .
14656 1 SM 0C02 SMMG  EXIT  GETMAIN/OK         00EAFF40                                                        =174468=
14656 1 AP 00E1 EIP   EXIT  GETMAIN            OK                    00F4,00000000 ....,00000C02 ....          =174469=
```

Problem #2

- If we look at the full trace, we see more information.

```
AP 00E1 EIP ENTRY GETMAIN                          REQ(0004) FIELD-A(05F95B08 .9$.) FIELD-B(09000C02 ....)

             TASK-14656 KE_NUM-001F TCB-00472000 RET-85B81138 TIME-09:39:15.3141357197 INTERVAL-00.0000003125    =174462=
 . . .       *** The EXEC CICS return address is 05B81138 when you remove the 31-bit addressing 8 bit ***

SM 0C01 SMMG  ENTRY - FUNCTION(GETMAIN) GET_LENGTH(154) SUSPEND(NO) STORAGE_CLASS(SHARED_USER24) CALLER(EXEC)

             TASK-14656 KE_NUM-001F TCB-00472000 RET-851FEFCC TIME-09:39:15.3141391884 INTERVAL-00.0000003437    =174465=
               1-0000  00780000 00000011 00000000 00000000  B6580000 00000000 0208016C 05B81B8E  *.........................%....*
                 0020  05F95B70 0508FB6C 00000154 80000000  05B81B8E 027F2001 007F0BD2 00000000  *.9$....%.............".....K....*
                 0040  0508F988 0508F680 85220FC4 050657A8  00000000 05221FC4 0508FB6C 00000004  *..9h..6.e..D...y.......D...%....*
                 0060  00000000 80000080 44040140 07010000  00680000 00000028                    *.......... ............        *
. . .
SM 0C02 SMMG  EXIT  - FUNCTION(GETMAIN) RESPONSE(OK) ADDRESS(00EAFF40)

             TASK-14656 KE_NUM-001F TCB-00472000 RET-851FEFCC TIME-09:39:15.3141476572 INTERVAL-00.0000000937    =174468=
               1-0000  00780000 00000011 00000000 00000000  B6580000 00000000 0208016C 05B81B8E  *.........................%....*
                 0020  05F95B70 00EAFF40 00000154 80000000  05B81B8E 027F2001 007F0BD2 00000000  *.9$.... .............".....K....*
                 0040  0508F988 0508F680 85220FC4 050657A8  00000000 05221FC4 0508FB6C 00000004  *..9h..6.e..D...y.......D...%....*
                 0060  00000000 80000080 44040140 07010000  00680000 00000028                    *.......... ............        *
```

Problem #2

- This is what I saw when I use a dump browser and go backwards from the EXEC CICS return address -2 to look for a program eye-catcher.

```
05B80EA8    -290   C4C6C8E8 C9F4F1F1 58F00014 58F0F0B4 | DFHYI411.0...00. |
05B80EB8    -280   58F0F00C 58FF000C 07FF0000 00000000 | .00............. |
05B80EC8    -270   47F0F028 00C3C5C5 00000000 00000014 | .00..CEE........ |  ← an LE-managed program
05B80ED8    -260   47F0F001 4ACEAC00 05B8790C 00000000 | .00.¢........... |
05B80EE8    -250   00000000 00000000 90ECD00C 4110F038 | ..............0. |
05B80EF8    -240   98EFF04C 07FF0000 05B87860 05B87954 | q.0<.......-.... |
05B80F08    -230   05B8A330 05B878C0 05B87860 05B87E20 | ..t........-..=. |
05B80F18    -220   05B8AAD0 05B87920 00000000 00000008 | ................ |
05B80F28    -210   C3C9C3E2 D6E2E4D4 F2F0F1F4 F0F7F1F9 | CICSPG0120140719 |  ← the actual program
05B80F38    -200   F1F0F2F2 F1F8F0F1 F0F1F0F1 00000000 | 102218010101.... |
. . .
05B81118     -20   01544110 D06841E0 80AE41F0 D1704100 | ..........0J... |
05B81128     -10   805290E0 10009680 100858F0 CF8405EF | ......o....0.d.. |  ← the EXEC CICS call
```

Problem #2

- The same return address can be found in the other trace entries for SHARED-24.
- I can also see the program in the LD=1 output shown below.
- Using the return address 05B81138, find the next higher Load Point and go back one program.
- If the return address is not found in LD=1 OUTPUT, you will need to use eye-catcher information to identify whose code it is.

```
PROGRAM STORAGE MAP

PGM NAME ENTRY PT  CSECT    LOAD PT. REL. PTF LVL. LAST COMPILED  COPY NO. USERS     LOCN  TYP

CICSPG01 85B80EC8 DFHYI411 05B80EA8 411                              1       11      ESDSA RPL
CICSPG05 85B817E8 DFHYI411 05B817C8 411                              1        2      ESDSA RPL
```

Problem #2 Conclusion

- It is a leak of 154-byte requests in the SDSA.
- Program CICSPG01 is performing a GETMAIN SHARED but there appears to be no code that is issuing a FREEMAIN.
- Fix the application program!

Problem #3

- After a CICS/VSE 2.3 and a z/VSE migration (with testing), recoverable SOS conditions started to occur and SM0131 SOS Below dumps were produced.
- SM=1 showed DSALIM=8192K, the UDSA had 4.75MB with 326K contiguous free, the CDSA had 1.5MB with 244K contiguous free, the SDSA had 1.25MB with 44K contiguous free and was SOS due to the 64K cushion having been released, the RDSA had 0.5MB with 148K contiguous free.
- XM showed MXT=150 user tasks, current active 139 with peak active 149.
- These symptoms suggest a capacity problem, and GETVIS availability suggested that it would not be difficult to add another 256K or 512K to DSALIM.
- But a migration was also involved, and making a diagnosis based only on symptoms can be very dangerous, so we needed to do more analysis.

Problem #3

- The DS domain showed:
  - More than 50 FCCIWAIT for the same file, with one task in FCIOWAIT and a few FCPSWAITs - the culprit or a victim? (Trace analysis shows it is the victim.)
  - More than 40 tasks in a Dispatchable state - a significant backlog of tasks that could be running - it is cpu availability or something else? (Trace analysis shows the cause.)
  - Other task states are "normal".
- A trace analysis showed that CICS was very busy (and an unusual amount of elapsed time was captured), that could explain why there were Dispatchable states, but is more than 40 of 139 tasks reasonable? (My experience would say "no".)

```
Trace elapsed time 35.0872540312   (seconds)
Task dispatch time 35.0501192200
Task idle time      0.0371348112
Task elapsed utilisation 99.89%
```

Problem #3

- The analysis showed that some tasks ran quickly and completed, but others ran very slowly, and two of the tasks with low task numbers (i.e., they started before many of the others in the reported interval) were dispatched for more than 34 seconds and did not even finish.

```
Task 32009 *** Response 35.0872540312 Total Dispatched 16.8256172195 Total Wait 18.2616368117 Elapsed:Dispatch ratio = 2.09
Task 32077 *** Response 35.0816723125 Total Dispatched 17.6481242812 Total Wait 17.4335480313 Elapsed:Dispatch ratio = 1.99
```

- The execution summary is not "normal", and GEMAST is not the file with the FCCIWAIT states; here is a part of the task summary showing some very long times.

```
Task 32077 Dispatched Elapsed: 0.2544422187 Start: =003323= 10:18:56.5282236262 End: =003425= 10:18:56.7826658449
Task 32077 Wait       Elapsed: 0.0026335313 Possible dispatch delay  (high priority TCP task is dispatched)
Task 32077 Dispatched Elapsed: 0.2656721875 Start: =003801= 10:18:56.7852993762 End: =003903= 10:18:57.0509715637
Task 32077 Wait       Elapsed: 0.0356217500 Possible dispatch delay  (so many VSAM requests that CICS let another task run)
Task 32077 Dispatched Elapsed: 0.2733920937 Start: =005394= 10:18:57.0865933137 End: =005496= 10:18:57.3599854074
Task 32077 Wait       Elapsed: 0.2659688125 Possible dispatch delay  (same as the previous reason)
Task 32077 Dispatched Elapsed: 0.2625850938 Start: =005914= 10:18:57.6259542199 End: =006016= 10:18:57.8885393137
Task 32077 Wait       Elapsed: 0.2664264375 Possible dispatch delay  (same as the previous reason)
Task 32077 Dispatched Elapsed: 0.0000440000 Start: =006482= 10:18:58.1549657512 End: =006484= 10:18:58.1550097512
Task 32077 Wait       Elapsed: 0.5313980000 FUNCTION(WAIT_OLDW) RESOURCE_NAME(GEMAST) RESOURCE_TYPE(FCIOWAIT)
```

Problem #3

- The full trace showed more than 13,000 VSAM exception *EXC* trace entries.
- The exception entry was repeated in the long dispatch times when the start and end trace sequence number are used to view the full trace output.

```
AP 04B7 FCVS *EXC* VSAM EXCEPTION - VSAM RPL
          TASK-32077 KE_NUM-0079 TCB-0031A000 RET-8D49EDE2 TIME-10:18:54.4216879074 INTERVAL-00.0000030937    =003324=
            1-0000  00780000 00000038 00000000 00000000  B4278C3C 00000000 05000100 00000000  *.............................*
              0020  00000000 04D157B0 00000009 00000000  00000000 00000000 00000000 00000000  *.....J.....................*
              0040  04D157B0 00000000 0677C2E0 0677C2E0  00000092 04D1CB70 00000000 00000000  *.J........B...B....k.J.....*
              0060  00140000 00000000 00000102 02010000  02000012 00000000                    *.....................       *
            2-0000  0011003C 00000000 0677C2E0 0677C2E0  00000092 00000004 0057B210 07100000  *.........B...B....k...........*
              0020  98100000 60080014 00000000 00000000  00003710 0677D4B8 00800000           *q...-.................M.....    *
```

- The second data area is the RPL, and offset X'24' contains the return code and error code.
- z/VSE Messages Volume 2 says X'08' and X'14' is a VSAM CI exclusive control issue and is significant when found in almost every exception trace.
- If you saw something like X'08' and X'10', it is a No Record Found, which is a "normal" exception in most cases (unless it repeats in a program loop).

Problem #3 Conclusion

- CICS and VSAM are looping on an exclusive control conflict until it is resolved, and the I/O can be started.
- This is not how they are designed to work together, so we would work with VSAM Support.
- One or more tasks monopolising CICS will have an impact on its ability to run normally, there will be a build up of tasks and their storage requirements until it is resolved, and even then, it will take time to get back to normal as CICS will probably have a backlog of work to deal with.
- Interestingly, CICS detects when a task performs many consecutive VSAM requests to the same file and does a CHANGE_PRIORITY to recalculate (lower) its dispatch priority and allow other user tasks to do some work.
- This will also affect lower PRTY z/VSE partitions while it is happening.
- CICS Monitoring task performance records would show a large ratio of VSAM requests to application EXEC CICS requests in this situation.

Thank You

Please forward your questions or remarks to
michaelalanpoil@gmail.com

Bonus Material

- The following slides were added after the first presentation.

Sample AP=3 Task Data Allocation

- Note the storage_location.taskid comment and the Storage Check Zone (SCZ)/Storage Accounting Area (SAA) in the first and last 8 bytes that are added for Storage Violation checking, meaning that task 38 was given the address 11EBF3B8 as the start of the allocation and not 11EBF3B0.

```
USER31.00038 11EBF3B0 USER storage above 16MB

  0000  E4F0F0F0 F0F0F3F8 000801B0 00000000  00000000 00000000 00000000 11EB005C  *U0000038.....................**    11EBF3B0

  0020  00000000 91DFE8BA 00000000 00000000  11EBF430 11D54120 00000000 11D9D728  *....j.Y...........4..N.......RP.*    11EBF3D0

  0040  11A18FFF 11A19FFE 11A1AFFD 11EBD920  11EB00D0 11EBF3C8 91DFE4B0 11787680  *..............R.......3Hj.U.....*    11EBF3F0

  0060  00000000 00000000 00000000 11EBF3C8  00530000 11EB00D0 11EBD920 00000000  *..............3H..........R.....*    11EBF410

  0080  91DFE8F7 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *j.Y7............................*    11EBF430

  00A0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*    11EBF450

  00C0 -    017F LINES SAME AS ABOVE                                                                            11EBF470

  0180  00000000 00000000 00000000 00000000  00000000 00000000 80587DC0 11D54120  *.........................'..N..*    11EBF530

  01A0  08000500 11EBD920 00000000 00000000  20000000 00000000 E4F0F0F0 F0F0F3F8  *......R.................U0000038*    11EBF550
```

Problem #4

- SOS Below occurred after a migration to the latest release of z/VSE that included the consolidation of two CICS systems into one.
- DSALIM peak usage in the consolidated CICS system grew from 3MB before the migration to 7MB or higher afterwards.
- If DSALIM was increased, the peak usage grew to fit the larger allocation and SOS continued to occur!
- Nothing obvious appeared to have changed in the definitions, but something obviously had!
- A lot of analysis later . . .
- In the migration, a COBOL program definition from the old CICS system was not added to the CSD and Program Autoinstall defined it as DATALOCATION(BELOW) instead of DATALOCATION(ANY), causing LE to allocate 188K of 24-bit RUWA storage instead of 31-bit RUWA storage.
- This program is used millions of times each day and the constant GETMAIN and FREEMAIN activity caused fragmentation in the UDSA.
- Successful Load Testing before going into Production is not a guarantee that there will be no problems!

Problem #4

```
===SM: STORAGE MANAGER DOMAIN - SUMMARY

   SM Domain status:                 INITIALISED
   Storage recovery:                 YES
   Storage protection requested:     YES
   Storage protection active:        YES
   Reentrant program option:         PROTECT
   Current DSA limit:                   7936K
   Current DSA total:                   7936K
   Currently SOS below 16M:              YES

   Current EDSA limit:                  100M
   Current EDSA total:                   87M
   Currently SOS above 16M:              NO
```

Problem #4

```
==SM: UDSA Summary

   Size:                          5120K
   Cushion size:                    64K
   Current free space:            1308K  (25%) ← A lot of free storage in total, but only 168K contiguous
 * Lwm free space:                1152K  (22%)
 * Hwm free space:                4524K  (88%)
   Largest free area:              168K
 * Times nostg returned:             0
 * Times request suspended:          30
   Current suspended:                1
 * Hwm suspended:                    8
 * Times cushion released:           0
   Currently SOS:                  YES
 * Times went SOS:                  15
 * Time at SOS:             00:00:00.208
 * Storage violations:               0
   Access:                        USER
 * Extents added:                   24
 * Extents released:                 4

   Number of extents:               20
```

Problem #4

Extent list:	Start	End	Size	Free
	00980000	009BFFFF	256K	0K
	00A40000	00A7FFFF	256K	0K
	00A80000	00ABFFFF	256K	20K
	00AC0000	00AFFFFF	256K	56K
	00B00000	00B3FFFF	256K	124K
	00B40000	00B7FFFF	256K	108K
	00B80000	00BBFFFF	256K	60K
	00BC0000	00BFFFFF	256K	48K
	00C00000	00C3FFFF	256K	48K
	00C40000	00C7FFFF	256K	84K
	00CC0000	00CFFFFF	256K	64K
	00D00000	00D3FFFF	256K	44K
	00D40000	00D7FFFF	256K	24K
	00D80000	00DBFFFF	256K	36K
	00E00000	00E3FFFF	256K	120K
	00E40000	00E7FFFF	256K	200K
	00E80000	00EBFFFF	256K	68K
	00EC0000	00EFFFFF	256K	68K
	00F00000	00F3FFFF	256K	68K
	00F40000	00F7FFFF	256K	68K

Problem #4

```
AP 00E1 EIP ENTRY LINK                                REQ(0004) FIELD-A(09EAD900 ..R.) FIELD-B(0D000E02 ....)

             TASK-02689 KE_NUM-00EB TCB-00458000 RET-89C55EAC TIME-17:46:32.4406461291 INTERVAL-00.0000003750    =054998=

PG 1101 PGLE  ENTRY - FUNCTION(LINK_EXEC) PROGRAM_NAME(xxxxxxxx) COMMAREA(09ECECC0 , 0000021C) SYSEIB_REQUEST(NO)

             TASK-02689 KE_NUM-00EB TCB-00458000 RET-88D99A7E TIME-17:46:32.4406463791 INTERVAL-00.0000002500    =054999=
. . .
AP 1940 APLI  ENTRY - FUNCTION(START_PROGRAM) PROGRAM(ISF6016) CEDF_STATUS(CEDF) EXECUTION_SET(FULLAPI) ENVIRONMENT_TYPE(EXEC)
                   SYNCONRETURN(NO) LANGUAGE_BLOCK(09A89864) COMMAREA(09ECECC0 , 0000021C) LINK_LEVEL(2) SYSEIB_REQUEST(NO)

             TASK-02689 KE_NUM-00EB TCB-00458000 RET-88C44824 TIME-17:46:32.4406477541 INTERVAL-00.0000007500    =055004=
. . .
SM 0301 SMGF  ENTRY - FUNCTION(GETMAIN) GET_LENGTH(E178) SUSPEND(YES) REMARK(LE_RUWA) STORAGE_CLASS(TASK31)

             TASK-02689 KE_NUM-00EB TCB-00458000 RET-88CD8F98 TIME-17:46:32.4406490041 INTERVAL-00.0000007500    =055009=
               1-0000  00400000 0000000E 00000000 00000000  B6580000 00000000 01000101 00000000  *. ............................*
                 0020  00000000 00000000 0000E178 00000000  00000000 0100D3C5 6DD9E4E6 C1400700  *.....................LE_RUWA ..*

SM 0302 SMGF  EXIT  - FUNCTION(GETMAIN) RESPONSE(OK) ADDRESS(0A7D0008)

             TASK-02689 KE_NUM-00EB TCB-00458000 RET-88CD8F98 TIME-17:46:32.4406495041 INTERVAL-00.0000005000    =055010=
               1-0000  00400000 0000000E 00000000 00000000  B6580000 00000000 01000101 00000000  *. ............................*
                 0020  00000000 0A7D0008 0000E178 0000E180  00000000 0100D3C5 6DD9E4E6 C1400700  *.....'.............LE_RUWA ..*

SM 0301 SMGF  ENTRY - FUNCTION(GETMAIN) GET_LENGTH(2EEA8) SUSPEND(YES) REMARK(LE_RUWA) STORAGE_CLASS(TASK24) ← 188K ,only 168K free

             TASK-02689 KE_NUM-00EB TCB-00458000 RET-88CD8F98 TIME-17:46:32.4406497541 INTERVAL-00.0000002500    =055011=
               1-0000  00400000 0000000E 00000000 00000000  B6580000 00000000 01000101 00000000  *. ............................*
                 0020  00000000 0A7D0008 0002EEA8 0000E180  00000000 0100D3C5 6DD9E4E6 C1400600  *.....'.....y.........LE_RUWA ..*

SM 1206 SMPQ  *EXC* - Insufficient_storage_to_satisfy_request - FUNCTION(ALLOCATE_PAGEPOOL_STORAGE) SUBPOOL_TOKEN(0EFAC50C)
                   GET_LENGTH(2EEC0) SUSPEND(YES)

             TASK-02689 KE_NUM-00EB TCB-00458000 RET-8F588460 TIME-17:46:32.4406510041 INTERVAL-00.0000012500    =055012=
```

Problem #4

```
. . .
SM 080A SMSY  *EXC* - Short_on_storage_in_the UDSA

              TASK-SM    KE_NUM-001D TCB-00458000 RET-8F53D8F6 TIME-17:46:32.4406585041 INTERVAL-00.0000002500    =055031=
                1-0000  E4C4E2C1 40404040                                                *UDSA                        *
                2-0000  0002A000                                                        *....                        *
                3-0000  00010000                                                        *....                        *
                4-0000  00000001                                                        *....                        *

ME 0301 MEME ENTRY - FUNCTION(SEND_MESSAGE) MESSAGE_NUMBER(83) COMPONENT_ID(SM)

              TASK-SM    KE_NUM-001D TCB-00458000 RET-8F593722 TIME-17:46:32.4406617541 INTERVAL-00.0000032500    =055032=
. . .
```

Problem #5

- SOS below, SM0131 dump captured and formatted.
- The trace shows SOS in the SDSA when trying to load a 314K program.
- CICS cannot find 314K contiguous in the SDSA.
- DSALIM is exhausted so it cannot add 512K (2 x 256K is the minimum it can add) to the SDSA to allow the program to be loaded.
- The SDSA occupies 5MB of the 6MB DSALIM, but SDSA free space is very fragmented.
- Other DSA usage is nominal and hence shows no problems.
- Almost all of the SDSA usage is due to subpool LDPGM, which is used for loading User Key application programs.
- The LD domain shows a lot of application programs being loaded many times and LDPGM has many Gets and Frees, both of which show that Program Compression was very active.
- The customer has a large number of 24-bit programs.
- DSALIM increased to 8M.

Problem #5

```
 AP 00E1 EIP ENTRY XCTL                                REQ(0004) FIELD-A(00788448 ..d.) FIELD-B(09000E04 ....)

              TASK-17148 KE_NUM-002D TCB-004C8000 RET-509FB2F6 TIME-14:42:23.2514827197 INTERVAL-00.0000045000    =042252=

 PG 1301 PGXE  ENTRY - FUNCTION(PREPARE_XCTL_EXEC) PROGRAM_NAME(INS001) SYSEIB_REQUEST(NO)

              TASK-17148 KE_NUM-002D TCB-004C8000 RET-8A3CEAB2 TIME-14:42:23.2514860634 INTERVAL-00.0000033437    =042253=
                1-0000  00500000 00000095 00000000 00000000  B8A00000 00000000 01EC1008 C5E2C3F3  *.&.....n.....................ESC3*
                  0020  F1F94040 02DF8390 00000003 0A138102  00000004 00000000 035BB01C 00702080  *19  ..c.......a..........f......*
                  0040  00780000 00000011 00020000 00000000                                      *.................               *

. . .

SM 0301 SMGF  ENTRY - FUNCTION(GETMAIN) SUBPOOL_TOKEN(04B8D944 , 00000024) GET_LENGTH(4E7D8) SUSPEND(YES)

              TASK-17148 KE_NUM-002D TCB-004BB000 RET-8A4E2A94 TIME-14:42:23.2515384697 INTERVAL-00.0000039375    =042260=
                1-0000  00400000 0000000E 00000000 00000000  BF400000 00000000 01520118 04B8D944  *. ................ ............R.*
                  0020  00000024 0359F660 0004E7D8 000000B0  0A4EA037 01EC1D02 01000000 00000008  *.....6-..XQ.....+..............*
                2-0000  D3C4D7C7 D4404040                                                        *LDPGM                          *

 SM 1206 SMPQ  *EXC* - Insufficient_storage_to_satisfy_request - FUNCTION(ALLOCATE_PAGEPOOL_STORAGE) SUBPOOL_TOKEN(04B8D944)
                   GET_LENGTH(4E7E0) SUSPEND(YES)

              TASK-17148 KE_NUM-002D TCB-004BB000 RET-8A50BE30 TIME-14:42:23.2515665009 INTERVAL-00.0000280312    =042261=
                1-0000  00380000 0000010A 00000000 00000000  BEF00000 00000000 01510201 04B8D944  *.................0............R.*
                  0020  0004E7E0 03011F2C 02EC1B98 00000040  04B8D944 01000008                    *..X........q... ..R.....        *
                2-0000  D3C4D7C7 D4404040                                                        *LDPGM                          *
                3-0000  E2C4E2C1 40404040                                                        *SDSA                           *
```

Problem #5

```
===SM: STORAGE MANAGER DOMAIN - SUMMARY

    SM Domain status:                  INITIALISED
    Storage recovery:                  YES
    Storage protection requested:      YES
    Storage protection active:         YES
    Reentrant program option:          PROTECT
    Current DSA limit:                    6144K
    Current DSA total:                    6144K
    Currently SOS below 16M:                YES
    Current EDSA limit:                     30M
    Current EDSA total:                     22M
    Currently SOS above 16M:                 NO

 ==SM: UDSA Summary

    Size:                              256K
    Cushion size:                       64K
    Current free space:                216K  (84%)
  * Lwm free space:                    112K  (43%)
  * Hwm free space:                    224K  (87%)
    Largest free area:                 216K
  * Times nostg returned:                 0
  * Times request suspended:              0
    Current suspended:                    0
  * Hwm suspended:                        0
  * Times cushion released:               0
    Currently SOS:                       NO
  * Times went SOS:                       0
  * Time at SOS:              00:00:00.000
  * Storage violations:                   0
    Access:                            USER
  * Extents added:                        2
  * Extents released:                     1

    Number of extents:                    1
```

Problem #5

```
==SM: CDSA Summary

   Size:                            512K
   Cushion size:                     64K
   Current free space:               92K  (17%)
 * Lwm free space:                    4K  ( 0%)
 * Hwm free space:                  260K  (50%)
   Largest free area:                56K
 * Times nostg returned:               0
 * Times request suspended:            0
   Current suspended:                   0
 * Hwm suspended:                       0
 * Times cushion released:            102
   Currently SOS:                      NO
 * Times went SOS:                      0
 * Time at SOS:              00:00:00.000
 * Storage violations:                  0
   Access:                           CICS
 * Extents added:                       2
 * Extents released:                    0

   Number of extents:                   2
```

Problem #5

```
==SM: SDSA Summary

   Size:                          5120K
   Cushion size:                    64K
   Current free space:            2240K  (43%)
 * Lwm free space:                 916K  (17%)
 * Hwm free space:                2240K  (43%)
   Largest free area:              288K
 * Times nostg returned:             0
 * Times request suspended:          3
   Current suspended:                 1
 * Hwm suspended:                     1
 * Times cushion released:            0
   Currently SOS:                   YES
 * Times went SOS:                    1
 * Time at SOS:           00:00:00.000
 * Storage violations:                0
   Access:                         USER
 * Extents added:                    24
 * Extents released:                  7

   Number of extents:                17
```

Problem #5

```
   Extent list:      Start      End          Size     Free

                     006C0000   006FFFFF     256K      64K
                     007C0000   007FFFFF     256K     116K
                     00800000   0083FFFF     256K     176K
                     00840000   0087FFFF     256K      76K
                     00880000   008BFFFF     256K     116K
                     008C0000   008FFFFF     256K      80K
                     00900000   0097FFFF     512K     152K
                     00980000   009BFFFF     256K      80K
                     009C0000   009FFFFF     256K       0K
                     00A00000   00A3FFFF     256K     180K
                     00A40000   00A7FFFF     256K      44K
                     00A80000   00ABFFFF     256K     140K
                     00AC0000   00AFFFFF     256K     252K
                     00B00000   00B7FFFF     512K     196K
                     00B80000   00BFFFFF     512K     288K
                     00C00000   00C3FFFF     256K      24K
                     00C40000   00C7FFFF     256K     256K
. . . .

==SM: Domain subpool summary (SDSA)

   Name     Id Chn  Initf Bndry Fxlen Q-c   Gets  Frees  Elems  Elemstg Pagestg

   APECA    65               8     8          1      1      0        0      0K
   DFHAPU24 4C              16               1      0      1      512      4K
   LDPGM    28              16            5939   5872     67  2699456   2748K    User-Key application programs
   LDRES    24              16               1      0      1    48080     48K
   SMSHRU24 68  Y           16               7      1      6    76800     80K
```

Problem #5

```
==SM: RDSA Summary

   Size:                          256K
   Cushion size:                   64K
   Current free space:            212K  (82%)
 * Lwm free space:                212K  (82%)
 * Hwm free space:                256K (100%)
   Largest free area:             212K
 * Times nostg returned:             0
 * Times request suspended:          0
   Current suspended:                0
 * Hwm suspended:                    0
 * Times cushion released:           0
   Currently SOS:                   NO
 * Times went SOS:                   0
 * Time at SOS:           00:00:00.000
 * Storage violations:               0
   Access:                   READONLY
 * Extents added:                    1
 * Extents released:                 0

   Number of extents:                1
```

Problem #5

PGM NAME	USE CNT.	USERS	LOADS	COPIES	LENGTH	USE	TYP	ATTRIBUTE	EXEC KEY	R/A OVERRIDE	MODE	DEFINITION DATE/TIME	CPE ADDR	STATUS
INS059	99	0	31	0	00000759	APP	RPL	REUSABLE	USER	-	-	7/05/19 07:32:03GMT	0352B3B0	LOCATED
INS060	19	0	14	0	000007C2	APP	RPL	REUSABLE	USER	-	-	7/05/19 07:32:03GMT	0352B490	LOCATED
INS061	94	0	23	0	00000705	APP	RPL	REUSABLE	USER	-	-	7/05/19 07:32:03GMT	0352B570	LOCATED
INS062	15	0	12	0	000006AA	APP	RPL	REUSABLE	USER	-	-	7/05/19 07:32:03GMT	0352B650	LOCATED
INS063	25	0	17	0	00000800	APP	RPL	REUSABLE	USER	-	-	7/05/19 07:32:03GMT	0352B730	LOCATED
INS064	63	0	18	0	00000721	APP	RPL	REUSABLE	USER	-	-	7/05/19 07:32:03GMT	0352B810	LOCATED
INS065	1	0	1	0	00000692	APP	RPL	REUSABLE	USER	-	-	7/05/19 07:32:03GMT	0352B8F0	LOCATED
INS066	57	0	16	0	0000071C	APP	RPL	REUSABLE	USER	-	-	7/05/19 07:32:03GMT	0352B9D0	LOCATED
INS067	102	0	25	1	00000725	APP	RPL	REUSABLE	USER	-	-	7/05/19 07:32:03GMT	0352BAB0	LOADED
INS068	32	0	22	0	0000071B	APP	RPL	REUSABLE	USER	-	-	7/05/19 07:32:03GMT	0352BB90	LOCATED

Problem #6

- SOS Above, CICS hung and a dump was produced and formatted.
- Many exception trace entries for allocation problems and suspends for storage, DS=1 shows suspends for ECDSA and EUDSA storage.
- EDSALIM=70MB and the ECDSA uses 59MB of it, and the EUDSA only has 1M allocated!
- The active tasks in the "Task Subpool Summary" have little storage allocated.
- The "Domain subpool summary (ECDSA)" a suspiciously large allocation with a large number of GETMAINs but no FREEMAINs!
- It looks like we have a CICS bug in the SOckets Domain!

Problem #6

```
==SM: Domain subpool summary (ECDSA)

   Name     Id Chn  Initf Bndry Fxlen Q-c   Gets   Frees  Elems  Elemstg Pagestg

   AITM_TAB D7         4K     8   592  Y       0       0     53    31376     36K
   AP_AFCTE B8  Y      4K    16              0       0     56     3712      4K
   AP_TCA31 53       128K   128  1536  Y   16205   16191     19    29184    128K
   . . .
   SOGENRAL 35              16              0       0      3     4656      8K
   SOLTE    71  Y           128  1024        0       0      1     1024      4K
   SOSERBUF 74              4  32768        0       0      0        0      0K
   SOSTE    72  Y           8    656        58256    0     72005 47235280  46884K  ← whoops!
   SOTBR    76         4K     8    48   Y       1       1      0        0      4K
   SOTDB    75         4K     8    184  Y       0       0      1      184      4K
   . . .
   WBGENRAL 2F              16              30536   30525     14     7888      8K
   WBPATHN1 8B              16    32    Y       0       0      0        0      0K
   WBPATHN2 8C              16    272   Y       0       0      0        0      0K
   WBRQB    85         4K     8    864   Y   58252   58241     11     9504     12K
   WBURIMAP 87              16    240   Y       0       0      0        0      0K
   WBURIXT1 88              16    128   Y       0       0      0        0      0K
   WBURIXT2 89              16    288   Y       0       0      0        0      0K
   WBVHOST  8A              16    176   Y       0       0      0        0      0K
   WBWRBR   86               8     40   Y       0       0      0        0      0K
   WEB_STA  82  Y            4    588          0       0      0        0      0K
   WEB_STA  DA  Y            4    588          418     420    115     67620     76K
   WEBELEM  DE  Y            8     96          418     420    114     10944     16K
   WEBHTML  DD  Y           16                 418     420    114   7167232   7292K
   WEB327B  83  Y            8  32768          0       0      0        0      0K
   WEB327B  DC  Y            8  32768          836     839      2    65536     64K
```

Problem #7

- SOS Above, CICS hung and a dump was produced and formatted with:

  CALL DFHPD430 DATA AP=3,DS=1,KE=1,LD=1,SM=1,TR=3,XM=1
- TR=3 and XM=1 were not helpful in this case.
- Only two user tasks appear to be active and one is in ECDSA wait, as is the TCP System Task number 00026.
- Try to find the cause.
- Hint: the CSNE System Task does not terminate until CICS terminates.

Problem #7

```
===AP: AP DOMAIN TRANSACTION SUMMARY

    Tran      Tran   Orig     TCA       TWA       EIB       SEIB      EIS       EIUS       Facility   Facility
     No        Id    Tran     Addr      Addr      Addr      Addr      Addr      Addr         Type       Id

    TCP       CSTP   CSTP   03583680  03B42438  03B420D0  03583A8C  03583988  03B42008
    00006     CSSY   CSSY   00713080  006B1000  0071E0D0  0071348C  00713388  0071E008
    00007     CSSY   CSSY   00713680  006B1000  0071F0D0  00713A8C  00713988  0071F008
    JBS       CSSY   CSSY   00714680  006B1000  007250D0  00714A8C  00714988  00725008
    00020     CSSY   CSSY   0073A680  006B1000  007320D0  0073AA8C  0073A988  00732008
    J01       CSSY   CSSY   00715080  006B1000  007200D0  0071548C  00715388  00720008
    J02       CSSY   CSSY   00715680  006B1000  007340D0  00715A8C  00715988  00734008
    00024     CSSY   CSSY   0073A080  006B1000  0073B0D0  0073A48C  0073A388  0073B008
    00026     CSNE   CSNE   03584080  03BA1438  03BA10D0  0358448C  03584388  03BA1008
    11819     FTRP   FTRP   00714080  006B1000  008450D0  0071448C  00714388  00845008
    11833     WATE   WATE   00712080  00840438  008400D0  0071248C  00712388  00840008       TC         P198
```

Problem #7

```
CICS24.00026 00A93000 CICS storage below 16MB

  0000  D4F0F0F0 F0F0F2F6 01206EE3 C1C3C240  00A92ED8 00005046 C4C6C8E3 C1C3C240  *M0000026..>TACB .z.Q..&.DFHTACB *
  0020  00000000 C1C5E9C4 C9C5E2C3 D3C5C1D5  00000000 00000000 00000000 00000000  *....AEZDIESCLEAN................*
  0040  00000000 00000000 00000000 0000FF00  00000000 00000000 D9C5C7E2 50D7E2E6  *........................REGS&PSW*
  0060  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
  0080 -    00FF LINES SAME AS ABOVE
  0100  00000000 00000000 00000000 00000000  00000000 00000000 00000000 03000000  *................................*
  0120  00000000 00000000 D4F0F0F0 F0F0F2F6                                       *........M0000026                *

. . . Repeats many times . . .

CICS31.00026 03BA5130 CICS storage above 16MB

  0000  C3F0F0F0 F0F0F2F6 0008022D 00000000  00000000 00000000 00000000 03BA105C  *C0000026.....................**
  0020  00000000 83571A5A 00000000 00000090  03BA51B0 80707DC0 837B9DF0 09A5FE00  *....c..!.............'.c#.0.v..*
  0040  037BADEF 037BBDEE 037BCDED 037BDDEC  00707648 83571968 03BA10D0 09A5F820  *.#...#...#...#......c........v8.*
  0060  00000000 00000000 00000000 03BA5148  006B0000 03BA10D0 09A5F820 00000000  *................,........v8.....*
  0080  03571B70 83571B40 00000000 00000000  00000000 00000000 00000000 00000000  *....c.. ........................*
  00A0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
  00C0 -    017F LINES SAME AS ABOVE
  0180  00000000 00000000 09A5F820 00000000  00000000 00000000 00384173 0344802C  *.........v8.....................*
  01A0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
  01C0 -    021F LINES SAME AS ABOVE
  0220  00000000 00000000 00000000 00000000  00000000 00000000 C3F0F0F0 F0F0F2F6  *........................C0000026*

CICS31.00026 03BB8130 CICS storage above 16MB

  0000  C3F0F0F0 F0F0F2F6 01206EE3 C1C3C240  03BA5E28 00005046 C4C6C8E3 C1C3C240  *C0000026..>TACB ..;...&.DFHTACB *
  0020  00000000 C1C5E9C4 C9C5E2C3 D3C5C1D5  00000000 00000000 00000000 00000000  *....AEZDIESCLEAN................*
  0040  00000000 00000000 00000000 0000FF00  00000000 00000000 D9C5C7E2 50D7E2E6  *........................REGS&PSW*
  0060  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
  0080 -    00FF LINES SAME AS ABOVE
  0100  00000000 00000000 00000000 00000000  00000000 00000000 00000000 03000000  *................................*
  0120  00000000 00000000 C3F0F0F0 F0F0F2F6                                       *........C0000026                *

. . . Repeats many times . . .
```

Problem #7

```
===DS: DISPATCHER DOMAIN - SUMMARY

 KEY FOR SUMMARY

     T  = TYPE OF TASK            S=SYSTEM  N=NON-SYSTEM
     S  = STATE OF TASK           D=DISPATCHABLE  S=SUSPENDED  R=RUNNING  E=RESUMED EARLY
     F  = PURGEABILITY FLAG       P=PURGEABLE  N=NOT PURGEABLE
     P  = PURGE STATUS            N=NO PURGE  X=PURGED  P=PURGE PENDING
     TT = TIMEOUT TYPE            IN=INTERVAL  DD=DEADLOCK DELAYED  DI=DEADLOCK IMMEDIATE
     W  = WAIT/SUSPEND TYPE       X=WAIT_EXTERNAL  S=SUSPEND  C=WAIT_OLDC  W=WAIT_OLDW
     DTA= DISPATCHER TASK AREA
     AD = ATTACHING DOMAIN
     M  = TASK MODE               Q=QUASI-REENTRANT  R=RESOURCE OWNING  S=FEPI OWNING
                                  O=SOCKET OWNING  L=LISTENER OWNING  Y=SSL OWNING


DS_TOKEN KE_TASK  T S F P TT RESOURCE RESOURCE_NAME    W TIME OF      TIMEOUT      DTA      AD ATTACHER M SUSPAREA XM_TXN_TOKEN
                         TYPE                         SUSPEND      DUE          (DSTSK)     TOKEN

00000003 09A7CC80 S S P N -  ECDSA                    S 15:19:04.802      -       0993B080 XM 03506700 Q 0993B080 035067000000026C
00020003 09A54C80 S S N N -  TIEXPIRY DS_NUDGE        S 15:25:57.359      -       0993B180 TI 00390003 Q 0993A5C0
00060003 03B4DB00 S S P N -  RECEIVE  DFHPSPIO        W 03:40:38.916      -       0993B380 XM 03506D00 Q 03BA607D 03506D000000020C
000C0003 09A7C900 S S N N -  TCP_NORM DFHZDSP         W 15:25:57.359      -       0993B680 XM 03506500 Q 00703C20 035065000000004C
00140003 09915C80 S S N N -  ICMIDNTE DFHAPTIM        S 04:00:01.001      -       0993BA80 XM 03506100 Q 0993BA80 035061000000006C
00160003 03B96780 S S N N -  ICEXPIRY DFHAPTIX        S 15:25:57.359      -       0993BB80 XM 03506900 Q 0993A590 035069000000007C
00180005 03B96B00 S S N N -  KCCOMPAT SINGLE          W 03:40:39.497      -       0993BC80 XM 03506B00 Q 0070369C 03506B000000024C
001C0003 03BAF780 S S N N -  JCTERMN  SUBTASK         W 03:40:38.564      -       0993BE80 XM 03BAE500 Q 00703000 03BAE5000000019C
00840005 03B57B00 S S N N -  JCJOURDS DFHJ01A 2..Qc:X. S 15:16:21.209      -       09919280 XM 03507100 Q 09919280 035071000000022C
008E0005 03B57780 S S N N -  JCJOURDS DFHJ02A ........ S 03:40:39.415      -       09919780 XM 03507300 Q 09919780 035073000000023C
00940001 0351D080 S S N N IN SMSYSTEM                 S 15:25:57.359 15:25:59.359 09919A80 SM 00000002 Q 0993A650
018C1017 03B96080 N S P N -  ECDSA                    S 15:19:04.794      -       09923680 XM 03507B00 Q 09923680 03507B000011833C
018E05FD 03B89400 N S N N -  EKCWAIT  SINGLE          W 15:17:39.563      -       09923780 XM 03506300 Q 0084588D 035063000011819C
```

Problem #7

```
PROGRAM REPERTOIRE

PGM NAME USE CNT.  USERS    LOADS    COPIES   LENGTH USE TYP ATTRIBUTE EXEC R/A MODE DEFINITION DATE/TIME CPE ADDR STATUS
                                                                      KEY OVERRIDE
IBM3VEX  0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:41GMT 03EDD3B0 UNUSED
IBM3VGT  0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:41GMT 03EDD490 UNUSED
IBM3VIA  0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:41GMT 03EDD570 UNUSED
IBM3VOCA 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:41GMT 03EDD650 UNUSED
IBM3VOCU 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:41GMT 03EDD730 UNUSED
IBM9LMSA 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:41GMT 03EDD810 UNUSED
IBM9LMSN 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:41GMT 03EDD8F0 UNUSED
IBM9LMSU 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:41GMT 03EDD9D0 UNUSED
IBM9LM2A 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:41GMT 03EDDAB0 UNUSED
IBM9LM2N 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:41GMT 03EDDB90 UNUSED
IBM9LM2U 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:41GMT 03EDDC70 UNUSED
IESAPT   0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5C110 UNUSED
IESBQP2  0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5C1F0 UNUSED
IESBQP2A 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5C2D0 UNUSED
IESBQP2B 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5C3B0 UNUSED
IESBQUP  0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5C490 UNUSED
IESBQUR  0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5C570 UNUSED
IESBSTA  0        0        0        0       00000000 APP RPL REUSABLE CICS  -    -  27/09/21 03:40:40GMT 03E5C650 UNUSED
IESBSTD  0        0        0        0       00000000 APP RPL REUSABLE CICS  -    -  27/09/21 03:40:40GMT 03E5C730 UNUSED
IESBSTG  0        0        0        0       00000000 APP RPL REUSABLE CICS  -    -  27/09/21 03:40:40GMT 03E5C810 UNUSED
IESBSTL  0        0        0        0       00000000 APP RPL REUSABLE CICS  -    -  27/09/21 03:40:40GMT 03E5C8F0 UNUSED
IESBSTR  0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5C9D0 UNUSED
IESBSTT  0        0        0        0       00000000 APP RPL REUSABLE CICS  -    -  27/09/21 03:40:40GMT 03E5CAB0 UNUSED
IESBSTU  0        0        0        0       00000000 APP RPL REUSABLE CICS  -    -  27/09/21 03:40:40GMT 03E5CB90 UNUSED
IESBXRT  0        0        0        0       00000000 APP RPL REUSABLE CICS  -    -  27/09/21 03:40:40GMT 03E5CC70 UNUSED
IESCFA   0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5CD50 UNUSED
IESCHLOG 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5CE30 UNUSED
IESCICIN 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5CF10 UNUSED
IESCICS  0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5E030 UNUSED
IESCICSD 0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5E110 UNUSED
IESCLEAN 296599   0        1        1       00000B2A APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5E1F0 LOADED
IESCLN1  0        0        0        0       00000000 APP RPL REUSABLE USER  -    -  27/09/21 03:40:40GMT 03E5E2D0 UNUSED
```

Problem #7

```
===SM: STORAGE MANAGER DOMAIN - SUMMARY

   SM Domain status:                 INITIALISED
   Storage recovery:                 YES
   Storage protection requested:     NO
   Storage protection active:        NO
   Reentrant program option:         PROTECT
   Current DSA limit:                   5120K
   Current DSA total:                   3840K
   Currently SOS below 16M:               NO

   Current EDSA limit:                   100M
   Current EDSA total:                   100M
   Currently SOS above 16M:               YES

. . .
```

Problem #7

```
==SM: ECDSA Summary

   Size:                          92160K
   Cushion size:                    128K
   Current free space:                0K  ( 0%)
 * Lwm free space:                   0K  ( 0%)
 * Hwm free space:                1024K  ( 1%)
   Largest free area:                 0K
 * Times nostg returned:           5200
 * Times request suspended:           4
   Current suspended:                 2
 * Hwm suspended:                     2
 * Times cushion released:          247
   Currently SOS:                   YES
 * Times went SOS:                    4
 * Time at SOS:           00:00:37.710
 * Storage violations:                0
   Access:                          CICS
 * Extents added:                    84
 * Extents released:                  0

   Number of extents:                90
```

Problem #7

```
==SM: EUDSA Summary

   Size:                          1024K
   Cushion size:                     0K
   Current free space:             960K  (93%)
 * Lwm free space:                 896K  (87%)
 * Hwm free space:                1024K (100%)
   Largest free area:              960K
 * Times nostg returned:             0
 * Times request suspended:          0
   Current suspended:                 0
 * Hwm suspended:                     0
 * Times cushion released:            0
   Currently SOS:                    NO
 * Times went SOS:                    0
 * Time at SOS:           00:00:00.000
 * Storage violations:                0
   Access:                         CICS
 * Extents added:                     1
 * Extents released:                  1

   Number of extents:                 1
```

Problem #7

```
==SM: ESDSA Summary

   Size:                          2048K
   Cushion size:                   128K
   Current free space:            1724K  (84%)
 * Lwm free space:                1236K  (60%)
 * Hwm free space:                1748K  (85%)
   Largest free area:              852K
 * Times nostg returned:             0
 * Times request suspended:          0
   Current suspended:                0
 * Hwm suspended:                    0
 * Times cushion released:           0
   Currently SOS:                   NO
 * Times went SOS:                   0
 * Time at SOS:           00:00:00.000
 * Storage violations:               0
   Access:                        CICS
 * Extents added:                    1
 * Extents released:                 2

   Number of extents:                2
```

Problem #7

```
==SM: ERDSA Summary

   Size:                          7168K
   Cushion size:                   256K
   Current free space:             356K  ( 4%)
 * Lwm free space:                  144K  ( 2%)
 * Hwm free space:                  356K  ( 4%)
   Largest free area:              200K
 * Times nostg returned:             0
 * Times request suspended:          0
   Current suspended:                0
 * Hwm suspended:                    0
 * Times cushion released:         273
   Currently SOS:                  YES
 * Times went SOS:                   7
 * Time at SOS:           00:00:38.186
 * Storage violations:               0
   Access:                    READONLY
 * Extents added:                    0
 * Extents released:                 0

   Number of extents:                6
```

Problem #7

```
==SM: Task subpool summary

   Current number of tasks:      11

   SMX Addr Name     Id Loc Acc   Gets  Frees  Elems  Elemstg Pagestg
   0994A020 M0000004 01  B   C       0      0      0        0      0K
            C0000004 03  A   C       0      0      1     1472      4K
            B0000004 02  B   C       0      0      0        0      0K
            U0000004 04  A   C       0      0      0        0      0K
   09932088 M0000006 01  B   C       0      0      1     1088      4K
            C0000006 03  A   C       0      0      0        0      0K
            B0000006 02  B   C       0      0      0        0      0K
            U0000006 04  A   C       0      0      0        0      0K
   099320BC M0000007 01  B   C       0      0      1     1088      4K
            C0000007 03  A   C       0      0      0        0      0K
            B0000007 02  B   C       0      0      0        0      0K
            U0000007 04  A   C       0      0      0        0      0K
   09932228 M0000019 01  B   C       0      0      1     1088      4K
            C0000019 03  A   C       0      0      1      128      4K
            B0000019 02  B   C       0      0      0        0      0K
            U0000019 04  A   C       0      0      0        0      0K
   099321F4 M0000020 01  B   C       0      0      1     1088      4K
            C0000020 03  A   C       0      0      2     1232      4K
            B0000020 02  B   C       0      0      0        0      0K
            U0000020 04  A   C       0      0      0        0      0K
   09932158 M0000022 01  B   C       0      0      2    66640     72K
            C0000022 03  A   C       0      0      1      128      4K
            B0000022 02  B   C       0      0      0        0      0K
            U0000022 04  A   C       0      0      0        0      0K
   0993218C M0000023 01  B   C       0      0      2     5200     12K
            C0000023 03  A   C       0      0      1      128      4K
            B0000023 02  B   C       0      0      0        0      0K
            U0000023 04  A   C       0      0      0        0      0K
```

Problem #7

SMX Addr	Name	Id	Loc	Acc	Gets	Frees	Elems	Elemstg	Pagestg
099321C0	M0000024	01	B	C	0	0	1	1088	4K
	C0000024	03	A	C	0	0	0	0	0K
	B0000024	02	B	C	0	0	0	0	0K
	U0000024	04	A	C	0	0	0	0	0K
09932054	M0000026	01	B	C	5048	0	5048	1534592	1504K
	C0000026	03	A	C	88110	96594	91555	88635920	86576K
	B0000026	02	B	C	0	0	0	0	0K
	U0000026	04	A	C	0	0	0	0	0K
099320F0	M0011819	01	B	C	0	0	0	0	0K
	C0011819	03	A	C	0	0	0	0	0K
	B0011819	02	B	C	2	0	2	4032	4K
	U0011819	04	A	C	0	0	0	0	0K
09932124	M0011833	01	B	C	0	0	0	0	0K
	C0011833	03	A	C	0	0	0	0	0K
	B0011833	02	B	C	5	2	3	6608	8K
	U0011833	04	A	C	4	2	2	51824	64K

Problem #7

```
==SM: Domain subpool summary (ECDSA)

   Name     Id Chn  Initf Bndry Fxlen Q-c   Gets   Frees  Elems  Elemstg Pagestg

   AITM_TAB DB         4K     8   592  Y       0       0     53    31376     36K
   AP_AFCTE CC  Y      4K    16                0       0    735    23520     24K
   AP_TCA31 53        72K   128  1536  Y     130     130      3     4608     72K
   AP_TXDEX 56         4K     8    64  Y       0       0   1493    95552     96K
   APAID31  65         4K     8   152  Y      75      74      1      152      4K
   APBMS    61  Y              16                0       0      0        0      0K
   APCOMM31 62                16              2714    2710     29    13584     16K
   APICE31  64         4K     8   168  Y    1214    1214      5      840      4K
   APURD    63                16                0       0      0        0      0K
   ASYNCBUF 73                 4  4096           0       0      0        0      0K
   BR_BSB   5A                16    64  Y       0       0      0        0      0K
   BRGENRAL 58                16                0       0      1       80      4K
   BRPC     59                16                0       0      0        0      0K
   BRVS     5B                16                0       0      0        0      0K
   BRVSCA   5D                16    32  Y       0       0      0        0      0K
   BRVSXA   5C                16    32  Y       0       0      0        0      0K
   CCNV_BCE 5F        64K     16                0       0      8   524288    512K
   CCNV_CCE 60                16   208  Y       0       0      0        0      0K
   CCNVG_AN 5E         4K     16                0       0      1       80      4K
   COLARAY  E4  Y             16                0       0      0        0      0K
   DDBROWSE 06         4K     16   304  Y       1       1      0        0      4K
   DDGENRAL 05                16                0       0     34    54112     64K
   DDS_DHT1 8C         4K     8    80  Y       0       0      8      640      4K
   DDS_DHT2 8D         4K     8    40  Y       0       0      8      320      4K
   DDS_PPT  43         4K     8    40  Y       0       0   4465   178600    180K
   DDS_RTXD 10         4K     8    40  Y       0       0      1       40      4K
   DDS_TCL  12         4K     8    40  Y       0       0     14      560      4K
   DDS_TPNM 11                8    96  Y       0       0      0        0      0K
   DDS_TXD  0F         4K     8    32  Y       0       0   1510    48320     48K
   DDS_USD1 33         4K     8    72  Y      38      40     73     5256      8K
   DDS_USD2 34         4K     8    32  Y      38      40     73     2336      4K
```

Problem #7

DDS_WBST	E6			8	40	Y	0	0	0	0	0K
DDS_WBUR	83			8	40	Y	0	0	0	0	0K
DFHAPDAN	4B			16			2	1	64	124080	124K
DFHTDDCT	94			16			0	0	0	0	0K
DFHTDG31	92			16			0	0	8	1776	4K
DFHTDIOB	95			16			0	0	1	8192	8K
DFHTDWCB	96		4K	16	64	Y	93817	93817	0	0	4K
DHDBB	86	Y		8	56		0	0	0	0	0K
DHDCR	85	Y		8	96		0	0	0	0	0K
DHDDB	88	Y		16			0	0	0	0	0K
DHDOA	84	Y		8	40		0	0	0	0	0K
DHGENRAL	49			16			0	0	1	144	4K
DHSTB	87	Y		16			0	0	0	0	0K
DHTLPOOL	89	Y		8	120		0	0	8	960	4K
DMSUBPOL	2C	Y	4K	16			0	1	1	1968	4K
DSBROWSE	07			8	24	Y	0	0	0	0	0K
FC_ABOVE	9A	Y	4K	16			1	1	1	864	4K
FC_DAM	C7		4K	8	128	Y	0	0	0	0	4K
FC_DLI	C8		4K	8	32	Y	0	0	0	0	4K
FC_DSNAM	CD		4K	8	160	Y	32	0	83	13280	16K
FC_FFLE	D0		4K	8	24	Y	2	0	11	264	4K
FC_FRAB	CE		4K	8	32	Y	1	0	4	128	4K
FC_FRTE	CF		4K	8	80	Y	0	0	8	640	4K
FC_SHRCT	CB		4K	16	1152	Y	0	0	15	17280	20K
FC_VSAM	C6		4K	8	240	Y	0	0	735	176400	184K
FCB_C1K	BB		4K	8	1024	Y	1	2	0	0	4K
FCB_C12K	BF			8	12288		0	0	0	0	0K
FCB_C16K	C0			8	16384		0	0	0	0	0K
FCB_C2K	BC			8	2048		0	0	0	0	0K
FCB_C20K	C1			8	20480		0	0	0	0	0K
FCB_C24K	C2			8	24576		0	0	0	0	0K
FCB_C256	B9		4K	8	256	Y	1	2	0	0	4K
FCB_C28K	C3			8	28672		0	0	0	0	0K
FCB_C32K	C4			8	32768		0	0	0	0	0K
FCB_C4K	BD			8	4096		1	2	0	0	0K

Problem #7

FCB_C512	BA			8	512	Y	0	0	0	0	0K
FCB_C8K	BE			8	8192		0	0	0	0	0K
IE_GENRL	45			16			0	0	1	64	4K
IECCB	47	Y		4	96		0	0	0	0	0K
IECSB	46	Y		4	96		0	0	0	0	0K
JAGLOBAL	97		4K	16			0	0	1	1200	4K
JCDYNLOG	A0			16			0	0	0	0	0K
JCFWDREC	98			8	24	Y	0	0	0	0	0K
KESTK31	14			128			0	0	28	34404	336K
KESTK31E	16			16			3	0	9	3684	36K
KETASK	17		4K	128	896	Y	0	0	28	2508	36K
LD_APES	19		4K	8	128	Y	50	100	368	4704	64K
LD_CNTRL	18			16			1	1	2	1516	16K
LD_CPES	1B		4K	8	224	Y	0	0	4717	105608	1052K
LD_CSECT	1A		4K	8	176	Y	50	123	564	9924	120K
LDENRS	22			16			0	0	28	35936	360K
LDENUC	1E			16			1	17	3	24680	256K
LI_PLB	57		4K	4	60	Y	45	1	199	1940	12K
MDTTABLE	DD	Y		16			0	0	0	0	0K
MN_CNTRL	44			16			1	1	10	7132	76K
MN_TMAS	77		4K	8	976	Y	4162	4161	11	1076	12K
OVERLAPD	E5	Y		16			0	0	0	0	0K
PGCHCB	3D			8	48	Y	0	0	0	0	0K
PGCPCB	3E			8	40	Y	0	0	0	0	0K
PGCRBB	42			16			0	0	0	0	0K
PGCRCB	3F			8	96	Y	0	0	0	0	0K
PGCSCBV	41			16			0	0	0	0	0K
PGCSCB4K	40			4096	4096		0	0	0	0	0K
PGGENRAL	36			16			0	0	1	240	4K
PGHMRSA	3B		4K	8	72	Y	15968	15968	0	0	4K
PGHTB	3A			8	2048		611	611	0	0	0K
PGLLE	38		4K	8	16	Y	3920	3919	29	464	4K
PGPGWE	39			8	24	Y	0	0	0	0	0K
PGPPTE	37		4K	8	72	Y	0	0	4465	32148	320K
PGPTA	3C		4K	8	72	Y	4182	4181	11	792	4K
PR_TABLE	D9			8	120	Y	0	0	0	0	0K

Problem #7

RCF_PFX	9E			16			0	0	0	0K
RCF_PSC	9D			8	1136	Y	0	0	0	0K
RCF_PSR	9C			8	400	Y	0	0	0	0K
RCF_PSU	9F			16			0	0	0	0K
RCF_PSW	9B			8	1232	Y	0	0	0	0K
RCLELEM	E1	Y		16			0	0	0	0K
RCTABLE	E2	Y		16			0	0	0	0K
RM_TABLE	55		4K	8	112	Y	4182	4181	11	1232	4K
ROWARAY	E3	Y		16			0	0	0	0K
SMSHRC31	6A	Y		16			29	29	2	2096	4K
SMTP	6E	Y		16			7188	7177	11	3696	4K
SOGENRAL	35			16			0	0	2	560	4K
SOLTE	71	Y		128	1024		0	0	0	0	0K
SOSERBUF	74			4	32768		0	0	0	0	0K
SOSTE	72	Y		8	656		0	0	0	0	0K
SOTBR	76			8	48	Y	0	0	0	0	0K
SOTDB	75			8	184	Y	0	0	0	0	0K
STSUBPOL	50	Y	36K	16			0	0	4	33024	36K
TIA_POOL	4E			16			0	0	1	64	4K
TIQCPOOL	4F		4K	8	88	Y	2249	2249	5	440	4K
TSBUFFRS	D3			16			0	0	1	262144	256K
TSGENRAL	D1			16			0	0	7	19040	20K
TSGIDS	D8		4K	8	96	Y	40	8	128	1228	16K
TSMAIN	D6			64			8	0	23	27968	32K
TSQERES	D7		4K	8	72	Y	8004	8004	0	0	4K
TSUTCTRL	D4			16			0	0	1	64	4K
TSUTNODE	D5		4K	4	44	Y	282	256	99	4356	8K
UE_EPBPL	70			8	80	Y	0	0	0	0	0K
USGENRAL	30			16			0	0	2	320	4K
USIDTBL	B4			8	32	Y	0	0	0	0	0K
USRTMQUE	4D		8K	8	16	Y	62	68	1	16	8K
USUDB	31		8K	128	128	Y	38	40	73	9344	12K
USXDPOOL	32		4K	8	32	Y	4136	4107	97	3104	4K
WBGENRAL	2F			16			0	0	3	4640	8K
WBPATHN1	81			16	32	Y	0	0	0	0	0K

Problem #7

WBPATHN2	82			16	272	Y	0	0	0	0K
WBRQB	7B			8	864	Y	0	0	0	0K
WBURIMAP	7D			16	240	Y	0	0	0	0K
WBURIXT1	7E			16	128	Y	0	0	0	0K
WBURIXT2	7F			16	288	Y	0	0	0	0K
WBVHOST	80			16	176	Y	0	0	0	0K
WBWRBR	7C			8	40	Y	0	0	0	0K
WEB_STA	78	Y		4	588		0	0	0	0K
WEB_STA	DC	Y		4	588		0	0	1	588 4K
WEBELEM	E0	Y		8	96		0	0	0	0 0K
WEBHTML	DF	Y		16			0	0	0	0 0K
WEB327B	79	Y		8	32768		0	0	0	0 0K
WEB327B	DE	Y		8	32768		0	0	0	0 0K
XMGENRAL	08		20K	16			80	80	12	15184 20K
XMLQEA	51		8K	8	296	Y	369	369	0	0 8K
XMSQEA	52		16K	8	64	Y	4611	4611	0	0 16K
XMTCLASS	0D		4K	8	128	Y	0	0	15	1920 4K
XMTRANSN	09		8K	256	512	Y	4182	4181	11	5632 12K
XMTXDINS	0A		4K	8	208	Y	0	0	1493	310544 316K
XMTXDSTA	0B		4K	8	152	Y	0	0	1493	226936 232K
XMTXDTPN	0C			8	64	Y	0	0	0	0 0K
XSGENRAL	2D			16			0	0	1	64 4K
XSXMPOOL	2E		4K	8	56	Y	4107	4106	2	112 4K
ZCBIMG	AD	Y		16			28	24	109	4720 8K
ZCBMSEXT	A8		4K	8	56	Y	28	24	139	7784 8K
ZCBUF	B1	Y		16			0	0	0	0 0K
ZCCCE	AA		4K	8	48	Y	0	0	30	1440 4K
ZCGENERL	B8	Y		16			0	0	6	70848 72K
ZCLUCBUF	B2			16			0	0	0	0 0K
ZCLUCEXT	A6		4K	8	240	Y	0	0	25	6000 8K
ZCNIBD	A7		4K	8	96	Y	28	24	109	10464 12K
ZCNIBISC	B0		8K	8	576	Y	0	0	0	0 8K
ZCNIBTRM	AF		8K	8	320	Y	96568	96567	1	320 8K
ZCRAIA	DA		4K	8	256	Y	0	0	10	2560 4K
ZCRPL	AE		8K	8	152	Y	3145	3133	12	1824 8K

Problem #7

ZCSETB	B6	Y		16			0	0	0	0	0K
ZCSKEL	A9			8	56	Y	0	0	0	0	0K
ZCSNEX	B3		4K	8	48	Y	28	24	139	6672	8K
ZCTCME	A5		4K	8	144	Y	0	0	5	720	4K
ZCTCSE	A4		4K	8	320	Y	0	0	4	1280	4K
ZCTCTTEL	A1		4K	8	512	Y	28	24	110	56320	64K
ZCTCTTEM	A2			8	448	Y	0	0	0	0	0K
ZCTCTTES	A3		4K	8	368	Y	28	24	169	62192	68K
ZCTPEXT	AB			8	24	Y	0	0	0	0	0K
ZC2RPL	B5			8	304	Y	0	0	0	0	0K

Problem #7

- In case you did not find the problem, it is due to IESCLEAN running in USER key and not CICS key, causing thousands of AEZD abends that GETMAIN TACB control blocks for CSNE, and because CSNE does not terminate they are not FREEMAINed - a classic example of a Storage Leak.
- The real issue though is why were there more than 150,000 terminals errors (the number of ">TACB" control blocks in storage)?