How to avoid or handle CICS storage availability problems

Mike Poil CICS L3

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®, CICS®

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.

Abstract

- Storage availability issues can often result in CICS or even z/VSE system down conditions, and it is often possible to avoid these situations by ensuring that your system is configured appropriately and is being monitored and reconfigured proactively.
- However, life is not always that simple and issues such as performance problems, changes and new bugs can still result in outages due to storage availability, hence it is important that the appropriate diagnostic information is provided so that IBM can deal with the problem as efficiently as possible.
- For CICS storage availability problems, understanding both CICS and z/VSE storage usage is of vital importance, therefore, Partition, Space and System GETVIS usage, optimization, monitoring and diagnosis are explained as well as the CICS equivalents.
- The session shows what diagnostic material IBM needs based on the symptoms, the correct commands to be used to monitor storage and force CICS dumps for cases such as SOS, and even includes typical SDAID commands for the various types of storage leak.
- This session is based on experience of dealing with customer PMRs .

Agenda

- My experience
- Avoid the avoidable wherever possible
- GETVIS/FREEVIS services
- GETMAIN/FREEMAIN/STORAGE services
- CICS DSA storage overview
- Monitoring CICS DSA usage
- How to monitor CICS Partition GETVIS usage
- Sample messages and output from PARSTAT
- How to optimize CICS Partition GETVIS usage
- How to monitor and optimize Space GETVIS usage
- How to monitor System GETVIS usage
- How to quickly increase System GETVIS-24
- How to reduce the size of the Shared Area-24
- Data collection for a GETVIS leak
- SDAID for a GETVIS or GETMAIN leak
- Data collection for CICS SOS
- Summary

My experience

- This presentation uses the word "CICS" to refer to CICS TS for VSE/ESA 1.1.1 unless specified otherwise.
- In my 6 years at IBM working in Java z/OS and CICS on z/VSE and CICS TS for z/OS, I have had *many* PMRs for storage usage problems.
- However, almost all of them were the result of a lack of effective monitoring and/or a lack of understanding about how storage is or should be used.
- Without excusing customer inaction, official documentation may not discuss the subject adequately, cross-product dependencies may not be covered, and some storage estimates may be missing or based on unrepresentative workloads.
- You need to be aware that:
    1. Most storage configuration limits are adjustable within certain constraints.
    2. Fragmentation may be part of or the whole of the cause - storage APIs need contiguous storage, and the subpooling algorithm used by the software *in conjunction with* configuration options *and* the pattern of requests can result in fragmentation such that the total storage is there, but not in a big enough free extent - *this is typically* ***not*** *a bug*.

My experience

- None of this is rocket science, I have discovered that it needs:
  1. An awareness that storage monitoring can be just as important as other monitoring - you may be closer to a disaster than you realise!
  2. Knowing what to monitor, how to react to situations and understanding the significance of exceeding a threshold or limit condition, specifically:
     - Can you increase the limit dynamically?
     - Do you need to reconfigure and bounce the system?
     - Do you need reconfigure and re-IPL?
  3. Some tooling to monitor usage.
  4. A bit of knowledge about how things work under the covers.

Avoid the avoidable wherever possible

- Ensure that you have a good service level for both IBM (i.e. a suitable RSL and newer APARs) ***and*** vendor products to help stability.
- Performance issues or bugs have the potential to slow CICS down and result in tasks holding on to their working sets longer than is normal and can cause SOS if your system runs close the limit of storage.
- CICS SDUMPs and transaction dumps take unnecessary time, and the serialization effect of *both* types of dump can cause slowdowns and increase concurrent task storage usage.
- *Capture data from **both** z/VSE **and** CICS on a regular basis*, store the information in a machine-readable form and use Threshold reporting (e.g. > 90% full) to prompt for analysis and action ***before*** it becomes a reported Limit condition.
- The WAVV website has a download for my PARSTAT Rexx program, which does exactly that for CICS based on STAT transaction (DFH0STAT) output, and I have included sample messages and graphs later in the presentation.
- If required, implement all available techniques to improve z/VSE 24-bit storage availability, e.g. IPL IODEV=1024, VPOOL=0K, VTAM IOBUF31=YES.
- For CICS 24-bit storage availability issues, try to reduce the size of the z/VSE 24-bit Shared Area, use LE ALL31(ON), convert CICS programs/maps to 31-bit etc.

Avoid the avoidable wherever possible

- Over-allocate 31-bit storage to allow for planned or unexpected expansion needs.
  - Using IPL NOPDS will require extra real storage, but that is cheap now.
  - Otherwise it costs nothing but space in DPD.
- Make sure that PASIZE is at least 10MB bigger than the largest partition ALLOC, you don't want an IPL to grow a CICS partition **now**!
- Make sure that you don't use all of z/VSE available storage - check MAP AVAIL.

```
map
. . .
AR 0015         SYSTEM      6144K
AR 0015         AVAIL     222048K
AR 0015         TOTAL     786432K   <----'
```

- Check user application storage usage *before* anything goes into production.
- Also cross-check Test/QA/Production configurations to ensure that they are compatible or what you port into production may behave differently.

GETVIS/FREEVIS Services

- GETVIS/FREEVIS macros are z/VSE native interfaces:
  - All native GETVIS is managed as a series of 6-character named subpools where each subpool owns zero or more 4K pages, and there is a "DEFAULT" subpool that is used when the requestor does not specify a subpool ID and hence no name.
  - Partition GETVIS storage is acquired and freed in 128-byte units.
  - LOC=BELOW is allocated low-to-high address in 24-bit storage.
  - LOC=ANY is allocated high-to-low address using 31-bit storage, and for a Partition, a single allocation can spill into high addressable free 24-bit storage; if no 31-bit storage is available, it is converted into LOC=BELOW.
  - LOC=RES is either BELOW or ANY depending on where the requestor is located.
  - *All except the first 4K of a CICS partition should be GETVIS, and it should use only a small amount of **native** Partition GETVIS in the DEFAULT and DFHEVP subpools.*
  - Dynamic Partition Space GETVIS is *always* 24-bit storage, and may be used by privileged (e.g. z/VSE) code for partition-related storage that would otherwise need to be allocated from the System GETVIS; the allocation unit is 16 bytes.
  - *CICS itself only requests a very small amount of Space GETVIS, but running CICS results in z/VSE using it, and approximately 140K is required for an SDUMP*.

GETVIS/FREEVIS Services

- System GETVIS may be used by privileged code, and is visible to every Address Space as it is in the Shared Area; the allocation unit is 16 bytes and it is allocated in 24- or 31-bit storage according to LOC=.
- *CICS makes direct use of 24-bit System GETVIS in the DEFAULT subpool for MRO, and uses a small amount of 24-bit and 31-bit in some emulated z/OS subpools, which are described by the next slide.*
- Approximately 140K of 24-bit System GETVIS storage is required for an SDUMP when CICS is running in a Static Partition.
- *Running CICS results in z/VSE using System GETVIS, e.g. you need up to 14K of 24-bit storage for each CICS subtask, using SIT TCPIP=YES adds 2 subtasks, SIT SSL=YES and FEPI=YES each add 1 subtask and vendor software may use additional subtasks*.
- The services and return codes are documented in the appropriate z/VSE System Macros Reference manual, e.g. return code 12/C normally means that GETVIS was unable to obtain the requested amount of contiguous storage.
- *What may not be known is that the System GETVIS area may report "full" when approximately 20K is still free, as there is a "cushion" of free storage that is only available to critical users such as DUMP.*

GETMAIN/FREEMAIN/STORAGE Services

- GETMAIN/FREEMAIN/STORAGE macros are emulated z/OS (was OS/390) interfaces.
  - A request is mapped to a GETVIS area based on the specified *subpool number* in the range 000 to 255 and uses a GETVIS subpool named "IMVSnnn".
  - MVS Diagnosis: Reference GA22-7588 describes each subpool's attributes and also explains z/OS SVCs, e.g. GETMAIN/FREEMAIN SVCs X'04', X'0A' and X'78'.
  - z/OS 8-byte multiples are rounded up to the next higher 128 or 16-byte multiple!
  - *CICS uses a **lot** of Partition GETVIS via these services since they are used to allocate the whole of DSALIM and EDSALIM, and for other uses such as the Internal Trace Table and data areas for the CICS Explorer (most of CICS uses the z/OS emulation API that is available under z/VSE).*
  - *Before you blame CICS for a GETMAIN leak, you need to know that it is possible for any IBM or vendor software product running under CICS (e.g. VSAM) to use a subset of these services - only CICS should use router SVC X'84' to request services whereas other products should use router SVC X'83' (*EXEC ,OS390 is required to be able to use SVC X'84').
  - The STORAGE macro uses a PC instruction not an SVC.

CICS DSA Storage Overview

- *DSALIM and EDSALIM are permanently allocated in Partition GETVIS whether CICS uses all of the storage or not, and EDSALIM **must** be entirely allocated in 31-bit storage.*
- DSALIM storage is sub-allocated for use in units of 256K and EDSALIM in 1MB units.
- DSA usage can be seen by CEMT I DSA**,** DFH0STAT, DFHSTUP, from a dump by DFHPD410 DATA SM=1 or by using vendor monitoring products**.**
- *DSALIM and EDSALIM can be increased by CEMT I DSA when CICS is running providing that there is sufficient contiguous free 24-bit and 31-bit GETVIS storage (*a decrease is also potentially possible in-flight!).
- The 8 CICS DSAs are:
  - The CDSA is used for CICS 24-bit control blocks and CICS-key program storage, and the ECDSA is for 31-bit.
  - The (E)RDSA is used for reentrant (SVA-eligible) CICS nucleus and user programs. (*Linking user reentrant programs as ",SVA" with SIT RENTPGM=PROTECT will force abends if the program is not reentrant, and could avoid some very obscure storage violations*.)
  - The (E)SDSA is used for CICS GETMAIN SHARED, which requires an explicit FREEMAIN, and for non-reentrant CICS nucleus and user programs.
  - The (E)UDSA is used by programs running in USER-key.

CICS DSA Storage Overview

- Each CICS DSA is managed as a series of subpools.
- CICS subpool usage is shown by DFHSTUP or DFHPD410 DATA SM=1.
- Google a subpool name to find an IBM Infocenter link that describes them or look in the CICS TS for VSE/ESA Performance Guide.
- CICS DSA and subpool storage are potentially subject to fragmentation, but I know of no *bugs* in this code.
- Sample output from DFHSTUP:

Domain Subpools

Subpool Name	Location	Access	Getmain Requests	Freemain Requests	Current Elements	Current Elem stg	Current Page stg	Peak Page stg
AITM_TAB	ECDSA	CICS	0	0	53	31376	36K	36K
AP_AFCTE	ECDSA	CICS	0	0	107	3424	4K	4K
AP_TCA24	CDSA	CICS	348526	348526	10	15360	28K	48K
AP_TCA31	ECDSA	CICS	90	90	4	6144	128K	128K
. . .								

Monitoring CICS DSA Usage

- Sample DFH0STAT: (it is repeated for EDSALIM, which shows pure 31-bit GETVIS usage)

```
Storage BELOW 16MB
__________________

  Partition GETVIS area size under 16 Mb  . . . . . :      11,260K
    Partition GETVIS used area below 16 Mb  . . . . . :      10,632K
    Partition GETVIS free area below 16 Mb  . . . . . :         628K
    Partition GETVIS maximum used below 16 Mb . . . . :      10,668K
    Partition GETVIS largest free area  below 16 Mb . :         624K

  ____________________________________________________
  Current DSA Limit . . . . . :       9,216K
  Current Allocation for DSAs . :       3,840K
  Peak Allocation for DSAs. . . :       3,840K
                                       CDSA              UDSA              SDSA              RDSA              Totals
                                  ____________________________________________________________________________________
  Current DSA Size. . . . . :         1,024K            1,792K              512K              512K            3,840K
  Current DSA Used. . . . . :           764K              16K               280K              280K            1,340K
  Current DSA Used as % of DSA. :        74%               0%                54%               54%               34%
* Peak DSA Used . . . . . :           820K            1,712K              284K              280K
  Peak DSA Size . . . . . :         1,024K            1,792K              512K              512K
  Cushion Size. . . . . :              64K               64K               64K               64K
  Free Storage (inc. Cushion) . :      260K            1,776K              232K              232K
* Peak Free Storage . . . . . :       304K            1,788K              264K              280K
* Lowest Free Storage . . . . . :     204K               80K              228K              232K
  Largest Free Area . . . . . :        236K              256K              220K              220K
  Largest Free Area as % of DSA :       23%               14%               42%               42%
  . . .
  Times no storage returned . . :          0                 0                 0                 0
  Times request suspended . . . :          0                 0                 0                 0
  Current requests suspended. . :          0                 0                 0                 0
  Peak requests suspended . . . :          0                 0                 0                 0
  Requests purged while waiting :          0                 0                 0                 0
  Times Cushion released. . . . :          0                 0                 0                 0
  Times Short-On-Storage. . . . :          0                 0                 0                 0
  Total time Short-On-Storage . :  00:00:00.00000    00:00:00.00000    00:00:00.00000    00:00:00.00000
  Average Short-On-Storage time :  00:00:00.00000    00:00:00.00000    00:00:00.00000    00:00:00.00000
  Storage Violations. . . . . :            0                 0                 0                 0
  Access. . . . . :                     CICS              USER              USER            READONLY
  '*' indicates values reset on last DSA Size change
```

How to Monitor CICS Partition GETVIS Usage

- Use z/VSE command GETVIS xx,RESET as soon as possible after CICS has initialized to reset the MAX. EVER USED value (the High Water Mark) for 24-bit storage - this is *not* a bug, CICS initialization allocates all 24-bit storage then frees what it does not need, so normal GETVIS output is factually correct but misleading ☺
- Use GETVIS xx before shutdown or use DFH0STAT to obtain summary usage data.
- GETVIS xx,ALL summarizes usage by subpool, and the subpool names can often be linked to the product that is using the storage.
- You may want to write Rexx code that executes GETVIS commands and creates machine-readable data if you have no other way to get the data.
- *The golden rules:*
  1. *Collect multiple sets of data over a period of time to obtain representative HWM values in order to get the best idea of how much is really free.*
  2. *Never use all free storage, and use a smaller amount if you have only one or two sets of values but you need to increase the allocated storage **now**.*
- GETVIS xx,DETAIL is more useful if you suspect a GETVIS leak, otherwise the amount of output can make it less easy to use than ALL.

How to Monitor CICS Partition GETVIS Usage

- If a RESET was done:

```
GETVIS F2
AR 0015 GETVIS USAGE    F2-24     F2-ANY                        F2-24     F2-ANY
AR 0015  AREA SIZE:   11,260K   122,876K (122,876K = ALLOC 120MB - 4K for SIZE=DFHSIP)
AR 0015  USED AREA:    9,828K   104,068K MAX. EVER USED:    9,828K   104,068K
AR 0015  FREE AREA:    1,432K    18,808K LARGEST FREE:      1,432K    18,808K
```

- If no RESET was done, MAX. EVER USED = AREA SIZE:

```
GETVIS F2
AR 0015 GETVIS USAGE    F2-24     F2-ANY                        F2-24     F2-ANY
AR 0015  AREA SIZE:   11,260K   122,876K
AR 0015  USED AREA:    9,828K   104,068K MAX. EVER USED:   11,260K   104,068K
AR 0015  FREE AREA:    1,432K    18,808K LARGEST FREE:      1,432K    18,808K
```

- *F2-ANY = 31-bit + F2-24* (only DFH0STAT shows 24-bit and pure 31-bit).
- USED is rounded to the next 4K; FREE AREA = AREA SIZE - USED AREA.
- MAX. EVER USED is the High-Water-Mark (HWM); AREA SIZE - LARGEST FREE is an approximate 24-bit HWM if no RESET was done.
- LARGEST FREE is contiguous and may be less than FREE AREA.
- "Available" contiguous storage is the smaller of (AREA SIZE - MAX. EVER USED) and LARGEST FREE.
- ***Assuming that the data is representative***, DSALIM can be ***safely*** increased by a maximum of about 1,024K and EDSALIM (or other usage) by about 16MB.

How to Monitor CICS Partition GETVIS Usage

- z/VSE 5.1 GETVIS xx,ALL with CICS usage in red, shared in blue and others in black (this CICS has STGPROT=YES and RENTPGM=PROTECT):

```
. . .
SUMMARY REPORT
SUBPOOL        REQUEST    <---F8-24-AREA---     ---F8-ANY-AREA-->
IMVS129                             3,584K               66,560K *** UNUSED DSALIM and EDSALIM ***
Default                             3,144K                3,212K     GETVIS default subpool
IMVS000                             2,688K                  464K     GETMAIN default subpool
IMVS252                               548K                5,496K     RDSA and ERDSA
IMVS130                               512K                2,048K     Used DSALIM and EDSALIM
IMVS130                               512K              366,592K     Used DSALIM and EDSALIM
CELH24                                 80K                    0K
IJBVSM                                 44K                   32K     VSAM default subpool
IMVS132                                40K                  212K     Kernel Stack based on MXT
CELHAN                                 28K                   36K
IJBAU                                  24K                  460K     VSAM AIX control blocks
IPNRSO                                 16K                   12K
IMVS229                                 8K                   32K
IMVS254        SVA                      4K                    0K
IMVS253        SVA                      4K                    0K
IMVS230                                 4K                    0K
IMVS253        SVA                      4K                    0K
IMVS255        SVA                      4K                    4K
IJBCTG                                  4K                    8K     VSAM catalog management
DFHEVP                                  4K                   48K
IMVS251                                 0K                   44K
IMVS229                                 0K                    4K
IJBPLH                                  0K                    4K     VSAM PLH etc.
USHEAP                                  0K                  128K
USTKAN                                  0K                   32K
IJBLSR                                  0K                  144K     VSAM LSR buffers etc.
IJBBUF                                  0K                  684K     VSAM NSR buffers
SUBPOOL TOTALS                     11,240K              446,260K     AREA SIZE is 460,784K
```

How to Monitor CICS Partition GETVIS Usage

- The main IMVSnnn subpools used by CICS are:
  - Subpool 000 is for general use by CICS *and other products:*
    - The CICS Trace Table is allocated in 31-bit storage and can be big - the CICS Service recommended minimum is 4MB.
    - The CICS Explorer may use a lot.
  - Subpool 129 is unallocated DSA extents if SIT STGPROT=YES.
  - Subpool 130 is allocated DSA storage, but see the note below about subpool 252.
  - Subpool 132 is CICS Nucleus Stack storage; this contains a save area and variables for each CICS module when executed for a CICS task; the size is a factor of MXT.
  - Subpool 252 is unallocated DSA extents if SIT STGPROT=NO, and (E)RDSA with SIT RENTPGM=PROTECT.
- *CICS moves DSA storage between subpools as extents are allocated and freed, therefore, do **not** report subpool 130 growing as a leak!*

How to Monitor CICS Partition GETVIS Usage

- *GETVIS ALL or DETAIL "xx-ANY-AREA" subpool total usage is pure 31-bit.*
- Subpool duplication is due to different storage keys being used.
- The final total for F8-ANY does **not** include the storage required for the GETVIS control information located at the high address end of any GETVIS area, however, this is included in the F8-ANY AREA SIZE seen in the start of the command output:

```
getvis f8,all
AR 0015 GETVIS USAGE    F8-24     F8-ANY                        F8-24     F8-ANY
AR 0015  AREA SIZE:   12,284K   460,784K
AR 0015  USED AREA:   11,240K   459,740K MAX. EVER USED:   12,284K   460,784K
AR 0015  FREE AREA:    1,044K     1,044K LARGEST FREE:      1,044K     1,044K
Summary Report
. . .
```

- The GETVIS area has been completely full at some point in time!
- The 1,044K of available storage is actually 24-bit - the F8-24 and F8-ANY LARGEST FREE are the same.
- *But, the matching GETVIS ALL output showed that 3.5MB of DSALIM and 65MB of EDSALIM had not been used (you would also see this in CEMT I DSA), and could potentially be reduced to allow the GETVIS storage to be used for something else.*
- If EDSALIM cannot be reduced without causing SOS, the Partition ALLOC must be increased to allow some GETVIS expansion.

How to Monitor CICS Partition GETVIS Usage

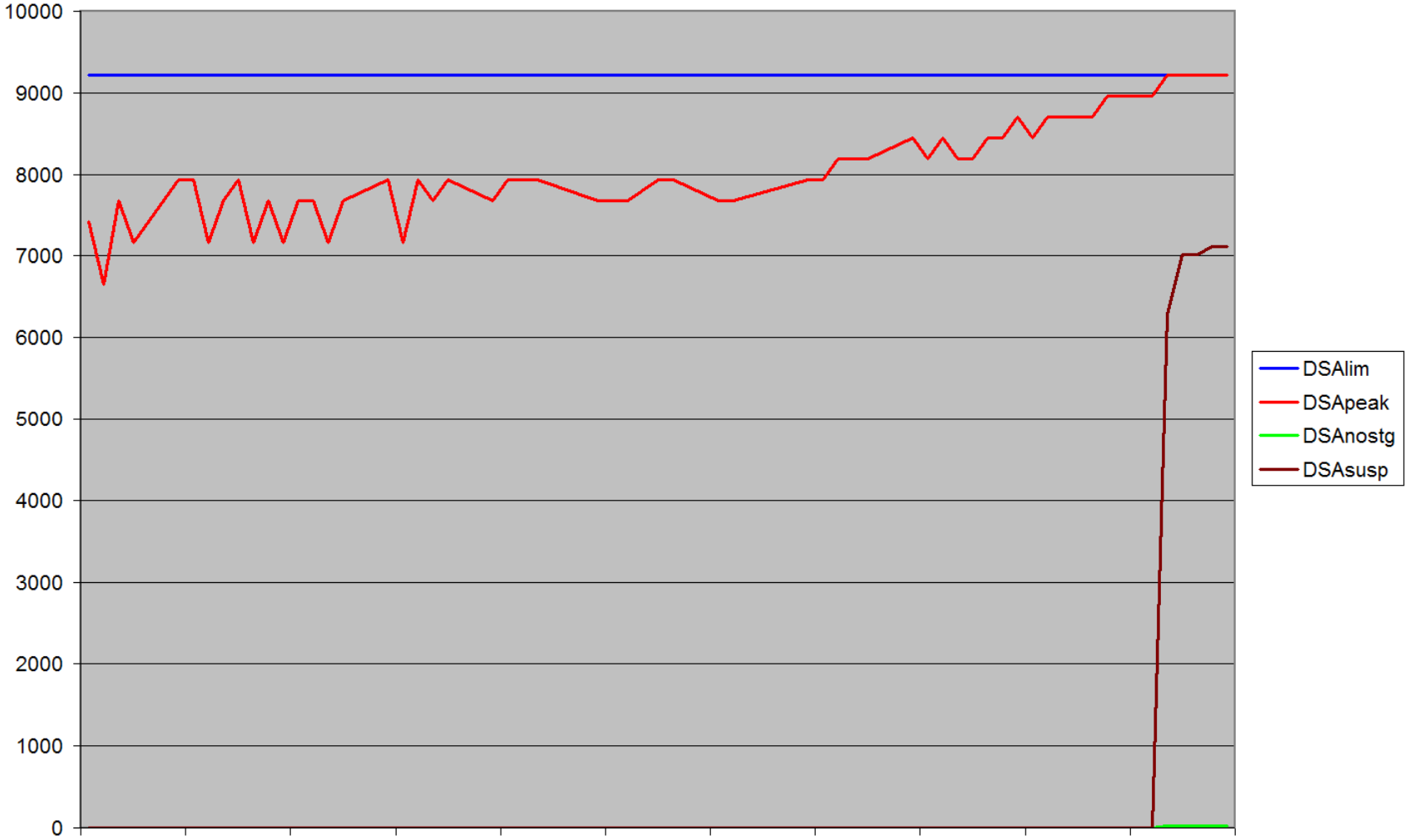
- CEMT I DSA compared to GETVIS will typically show DSALIM close to GETVIS 24-bit usage, although allocated ICCF 24-bit interactive partitions and possibly vendor products will stop that being true:

```
I DSA
STATUS:  RESULTS - OVERTYPE TO MODIFY
   Sosstatus(Notsos)

  Dsalimit( 05242880 )     DSALIM 5MB     you can change the value
   Cdsasize(00524288)       CDSA  0.5MB
   Rdsasize(00524288)       RDSA  0.5MB
   Sdsasize(00262144)       SDSA  0.25MB
   Udsasize(00262144)       UDSA  0.25MB  In use total 1.5MB of 5MB
                                          G1-24 usage due to DSA is 5MB

  Edsalimit( 0020971520 )  EDSALIM 20MB   you can change the value
   Ecdsasize(0003145728)    ECDSA  3MB
   Erdsasize(0007340032)    ERDSA  7MB
   Esdsasize(0001048576)    ESDSA  1MB
   Eudsasize(0001048576)    EUDSA  1MB   In use total 12MB of 20MB
                                          G1-ANY usage due to DSA is 25MB
getvis g1
AR 0015 GETVIS USAGE    G1-24     G1-ANY                      G1-24     G1-ANY
AR 0015  AREA SIZE:   11,260K    39,932K
AR 0015  USED AREA:    5,524K    28,972K MAX. EVER USED:    5,628K    29,180K
AR 0015  FREE AREA:    5,736K    10,960K LARGEST FREE:      5,632K    10,752K
. . .
```

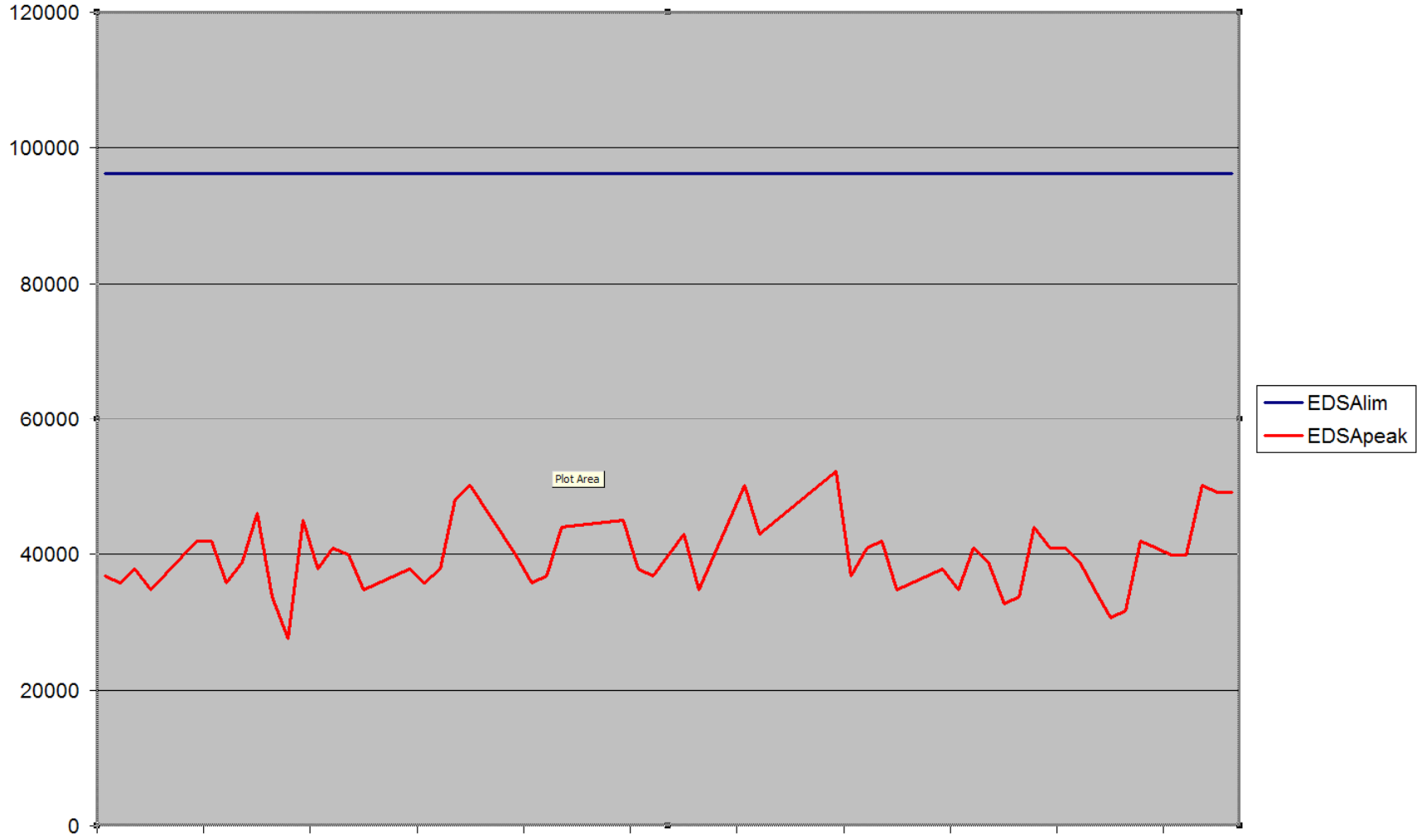
Sample messages from PARSTAT

```
Limit         DSALIM  peak usage 100%,        0K is available
Limit         DSALIM  Getmain below requests caused task suspends       7012 times
Limit         DSALIM  Getmain below requests had response "No Storage" returned         27 times
Limit         DSALIM  SOS below 16MB          4 times
Limit         DSALIM  SOS time below 16MB      1.13 minutes
Threshold     DSALIM  cushion released          6 times, CICS was approaching SOS below 16MB
Threshold     DSALIM  cushion released        103 times, CICS was approaching SOS below 16MB
Threshold     DSALIM  peak usage  92%,     768K is available
Threshold     DSALIM  peak usage  97%,     256K is available
Threshold     GETVIS below 16MB has only     488K available based on current usage
```

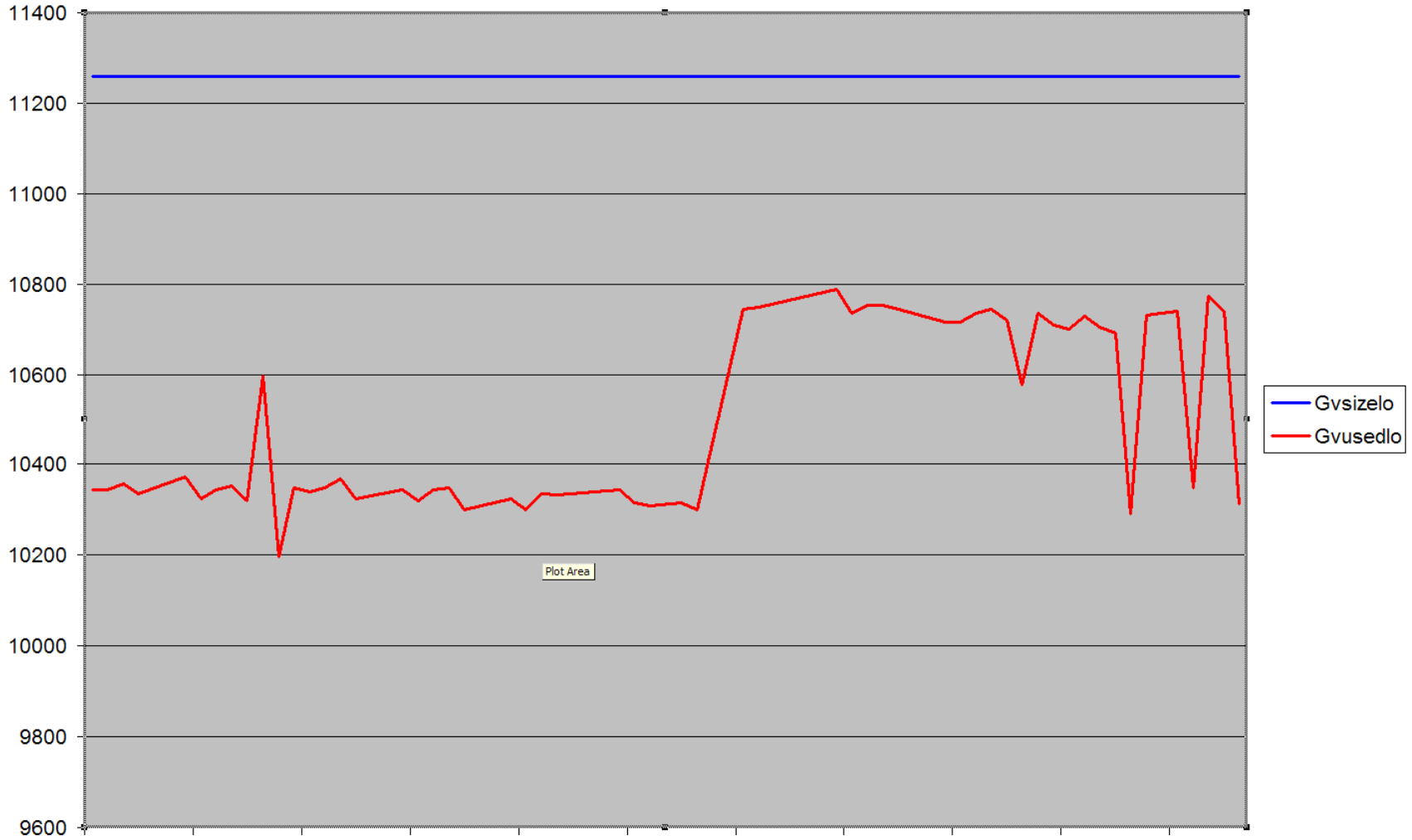
Sample output from PARSTAT-produced data



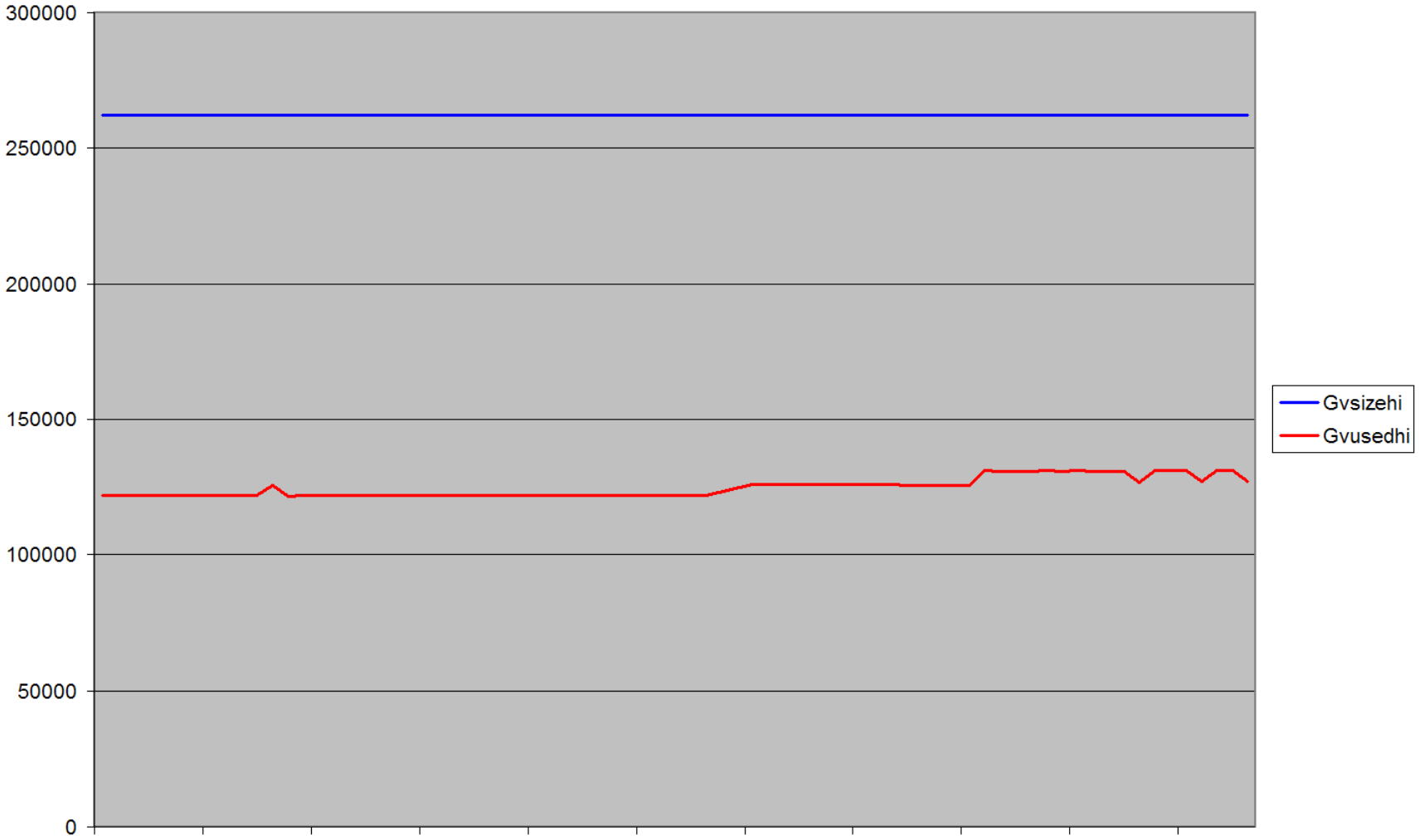
Sample output from PARSTAT-produced data



Sample output from PARSTAT-produced data



Sample output from PARSTAT-produced data



How to Optimize CICS Partition GETVIS Usage

- Use sensible DSALIM and EDSALIM values after using DFH0STAT or DFHSTUP to see what is your normal peak usage, but always allow for some expansion of uses.
- Avoid VSAM NSR (CEDA DEFINE FILE LSRPOOLID=NONE) where possible to stop VSAM allocating a unique buffer pool for the dataset (and you should get better performance from LSR at the same time ☺).
- Optimize CICS (E)DSA usage - see my WAVV presentation.
- Install all fixes that correct GETVIS usage.

How to Monitor and Optimize Space GETVIS Usage

- Get the data before shutdown.
- In this case, CICS needs a minimum of 92K + 140K for a dump - round up to at least 256K.
- *You will get 768K more CICS 24-bit storage and the same amount of 31-bit storage* ☺ ☺
- Change DTR$DYNz, PLOAD DYNC<,ID=z> and bounce CICS to exploit the difference.
- Use a Static Partition if you are really desperate for CICS 24-bit storage, but remember that System GETVIS will now be used for what was in the Space GETVIS.

```
GETVIS G1
. . .
AR 0015 DYNAMIC-SPACE GETVIS USAGE
AR 0015  AREA SIZE:    1,024K
AR 0015  USED AREA:       92K            MAX. EVER USED:       92K
AR 0015  FREE AREA:      932K            LARGEST FREE:        932K
AR 0015 1I40I  READY
```

How to Monitor System GETVIS Usage

- Make sure that you get fully representative data as its usage may fluctuate over time.
- A single allocation *cannot* span the 16MB line.
- The data is interpreted in a similar way to the output from the Partition GETVIS.
- System GETVIS full can occur when there still is 20K of FREE storage - this is not a bug.
- A CICS dump will need about 140K free.
- GETVIS SVA,ALL and DETAIL are also possible.
- An IPL is required to change its size.
- The example below shows no problems, in fact, the 24-bit area appears to be over-allocated and it looks like the 24-bit Shared Area size could easily be reduced by 1MB to provide more 24-bit CICS storage if it is required - AREA SIZE - MAX. EVER USED = 1,528K!

```
getvis sva
AR 0015 GETVIS USAGE   SVA-24    SVA-ANY                      SVA-24    SVA-ANY
AR 0015  AREA SIZE:    3,172K    14,016K
AR 0015  USED AREA:    1,584K     5,200K MAX. EVER USED:     1,644K     5,268K
AR 0015  FREE AREA:    1,588K     8,816K LARGEST FREE:       1,588K     7,228K
```

How to Monitor System GETVIS Usage

- The two examples below show that the 24-bit area is nearly too small for safety (i.e. to handle a CICS SDUMP) and the effect of fragmentation on contiguous storage:

```
GETVIS SVA
AR 0015 GETVIS USAGE   SVA-24    SVA-ANY                    SVA-24    SVA-ANY
AR 0015  AREA SIZE:    2,232K    19,188K
AR 0015  USED AREA:    2,064K    12,600K MAX. EVER USED:    2,140K    14,060K
AR 0015  FREE AREA:      168K     6,588K LARGEST FREE:        132K     5,288K

GETVIS SVA
AR 0015 GETVIS USAGE   SVA-24    SVA-ANY                    SVA-24    SVA-ANY
AR 0015  AREA SIZE:    2,872K    35,220K 6066
AR 0015  USED AREA:    2,648K    11,824K MAX. EVER USED:    2,844K    12,716K
AR 0015  FREE AREA:      224K    23,396K LARGEST FREE:        96K    23,024K
```

- Going "full" will look like this:

```
getvis
AR 0015 GETVIS USAGE   SVA-24    SVA-ANY                    SVA-24    SVA-ANY
AR 0015  AREA SIZE:    1,568K    12,340K
AR 0015  USED AREA:    1,548K     4,084K MAX. EVER USED:    1,548K     4,084K
AR 0015  FREE AREA:       20K     8,256K LARGEST FREE:        20K     8,236K

L257I  SYSTEM GETVIS SPACE OR DYNAMIC SPACE GETVIS EXHAUSTED
1S40I  SYSTEM ERROR, PROCMAC  RET.CODE=24
1S78I  JOB TERMINATED ABNORMALLY

0D22I INSUFFICIENT GETVIS FOR REQUESTED FUNCTION (no console redisplay!)
```

How to quickly increase System GETVIS-24

- Increasing 31-bit is easy providing you have available VSIZE or real storage.
- Increasing the allocation requires a change to the IPL SVA command, which specifies two values to be added to z/VSE's calculation of the required base sizes as 24-bit and 31-bit storage respectively.

```
SVA PSIZE=(652K,6M),SDL=700,GETVIS=(768K,9M)
```

- *Incorrectly increasing SVA command 24-bit allocations may cause the overall size of the 24-bit Shared Area to grow by 1MB and result in many problems.*
- If the MAP command shows UNUSED > 64K, you could exploit some or all of it in 64K units - the output below shows that 256K - 64K = 196K could be added.
- The resultant 0J45I IPL message shows that z/VSE added 6K above what was requested.

```
map
AR 0015  SPACE AREA       V-SIZE   GETVIS   V-ADDR   UNUSED NAME
AR 0015    S   SUP         788K                0          $$A$SUPI
AR 0015    S   SVA-24      1376K    1676K    C5000     256K

SVA PSIZE=(652K,6M),SDL=700,GETVIS=(960K,9M)

0J45I 24-BIT SYSTEM GETVIS AREA ROUNDED BY 6K
      31-BIT SYSTEM GETVIS AREA ROUNDED BY 1004K
```

How to quickly increase System GETVIS-24

- Additional 24-bit storage may be available by swapping unused 64K units out of the 24-bit Virtual Library.
- The LIBR LD SDL output shows that up to 256K could be swapped for GETVIS.
- The IPL SVA command can now be modified again by subtracting 256K from the 24-bit PSIZE and adding it to the 24-bit GETVIS.

```
STATUS DISPLAY       SDL  AND  SVA                  DATE: 2013-06-06
                                                    TIME: 13:05
-------------------------------------------------------------------
SDL       TOTAL ENTRIES :    908   (100%)
          USED  ENTRIES :    516   ( 57%)
          FREE  ENTRIES :    392   ( 43%)

SVA(24)   TOTAL SPACE   :   1312K  (100%)
          USED  SPACE   :    996K  ( 76%)
           - PFIXED AREA:     96K  (  7%)  START AT: 00204D08
          FREE  SPACE   :    316K  ( 24%)


SVA PSIZE=(396K,6M),SDL=700,GETVIS=(1216K,9M)
```

How to reduce the size of the Shared Area-24

- My WAVV 2012 presentation on How to Monitor and Optimize CICS TS Storage has a detailed example of how to do this.
- Part 1:
  1. Exploit VTAM IOBUF31=YES first.
  2. Optimize IPL options, e.g. VPOOL=0K, IODEV=1024, remove redundant ADDs.
  3. Re-IPL to before doing Part 2.
- Part 2:
  1. Start with a value for how much System GETVIS-24 is available above the biggest HWM, but keep a safety margin.
  2. Add unused Virtual Library-24 in 64K units.
  3. Add UNUSED Shared Area-24 less 64K.
  4. If you have more than 1MB of savings, decrease the SVA command 24-bit PSIZE and/or GETVIS sizes based on savings from (1) and (2), you will just absorb the savings from (3).
  5. Re-IPL, check and correct as required.
  6. Some Partition ALLOCs may need to be increased by 1MB.

Data collection for a GETVIS leak

- If it is a fast leak you will probably want IBM involvement as quickly as possible, but you can still collect some data.
- Use GETVIS xx,DETAIL at regular intervals to give you an idea of where the increase is occurring in terms of subpools, you may then identify the product and hence the vendor.
- I use a compare product on my PC (after sorting the outputs) to find the differences.
- What I require for CICS will include:
  - Repeated GETVIS xx,DETAIL.
  - AR DUMP xx,0-7FFFFFFF,cuu immediately after a couple of the GETVIS commands that show an *actual* increase in usage. (N.B. This dump is *asynchronous* with CICS execution so it should not affect it, unlike a CEMT P SNAP, which is *synchronous* and stops almost all of CICS processing.)
  - An SDAID trace according to which type of GETVIS usage is affected.
- FYI: Enter this console command *once* to make it easy to take a *synchronous* CICS console dump; e.g. using DUMPCICS G1,400 will dump G1 to tape drive address X'400'.

```
STACKP DUMPCICS|STATUS &0|SUSPEND &0|DUMP &0,0-7FFFFFFF,&1|RESUME &0
```

Data collection for a GETVIS leak

- GETVIS xx,DETAIL includes storage address ranges.
- The storage addresses may provide clues about how it is being used when viewed in a dump or by the z/VSE SHOW command, e.g. SHOW V1,509000.FFF.

```
getvis v1,detail
AR 0015 GETVIS USAGE    V1-24     V1-ANY                      V1-24     V1-ANY
AR 0015  AREA SIZE:   11,260K   203,768K
AR 0015  USED AREA:    5,728K    96,508K MAX. EVER USED:   11,260K   102,052K
AR 0015  FREE AREA:    5,532K   107,260K LARGEST FREE:      5,532K   107,248K
AR 0015 DYNAMIC-SPACE GETVIS USAGE
AR 0015  AREA SIZE:    1,024K
AR 0015  USED AREA:      104K            MAX. EVER USED:      104K
AR 0015  FREE AREA:      920K            LARGEST FREE:        920K
AR 0015 SUMMARY REPORT
AR 0015 SUBPOOL       REQUEST   <---V1-24-AREA---    ---V1-ANY-AREA-->
AR 0015 IMVS129                          3,584K             62,464K
AR 0015                         00700000-00A7FFFF  08C00000-0C8FFFFF
AR 0015
AR 0015 IMVS252                            520K              7,544K
AR 0015                         00509000-00509FFF  0CA0E000-0CA0EFFF
AR 0015                         0052F000-0052FFFF  0C93A000-0C97FFFF
AR 0015                         0052F000-0052FFFF  0C93A000-0C97FFFF
AR 0015                         005C0000-005FFFFF  0C928000-0C931FFF
AR 0015                         00680000-006BFFFF  08400000-085FFFFF
AR 0015                                            07A00000-07EFFFFF
AR 0015                                            078FC000-078FDFFF
AR 0015                                            078C8000-078D0FFF
AR 0015                                            077DD000-077DEFFF
. . .
```

SDAID for a GETVIS leak

- I normally provide the SDAID job, you need to:
  - Allocate a real tape drive (*not* a Virtual Tape).
  - Run the initialization job.
  - Use STARTSD to start SDAID.
  - Capture enough data to show the leak (check GETVIS outputs).
  - STOPSD to stop it.
  - ENDSD to remove the SDAID hooks.
- To print the trace data use:

```
// EXEC DOSVSDMP,PARM='PRINT SDAID TAPE=cuu FILE=1'
```

- A basic Partition native GETVIS trace would look like this (n.b. the ADDR= *is* required):

```
// EXEC SDAID
  OUTDEV  T=cuu
  TRACE GETVIS=PARTITION AREA=xx ADDR=0:7FFFFFFF
/*
```

- See z/VSE Diagnosis Tools for this command, GETVIS=SPACE and GETVIS=SVA and other SDAID requirements and possibilities.

SDAID for a GETMAIN leak

- For GETMAIN, the job would like this:

```
// EXEC SDAID
  OUTDEV  T=cuu
  TRACE SVC=(83,84) AREA=xx ADDR=0:7FFFFFFF OUTPUT=(GREG)
  TRACE INSTR=B218  AREA=xx ADDR=0:7FFFFFFF OUTPUT=(GREG)
/*
```

- The SVC 83 and 84 traces will catch ***all*** emulated z/OS services, only z/OS SVCs 4, 10 (X'0A') and 120 (X'78') are the ones to look at.
- The next slide is provided for reference purposes and shows how to read the output.
- The B218 trace will catch the STORAGE macro and ***all*** other PC instructions, and it is a big overhead if it is not needed - maybe don't use it for the first trace.
- A PC call with R14=0000030B/00000311 is for STORAGE OBTAIN/RELEASE, and has the same register usage that is shown on the next slide.

SDAID for a GETMAIN leak

- Here is SDAID output for a GETMAIN leak at address 10944822, which was in phase IKQNEX - this was fixed by APAR DY47426 or DY47427:

```
SVC    C2 C2  C00  SVC=83  ADDR=10944822  R00=000000F0 R01=00000000 R15=00000002
                 MVS-SVC=78  (SIMULATED SVC)
GR 0-7   000000F0 00000000 109461E8 909447C0 06692100 030AA670 00000000 1096D890
   8-F   00000000 0B400000 030AA3B8 030AA018 109457BF 034B0FD0 000118F0 00000002
```

- R0 is the number of bytes to GETMAIN/FREEMAIN, R1 is the address for FREEMAIN, and the R15 bytes are:

```
0 Options
1 Key for special subpools only
2 Subpool number
3 Option byte:
0... .... Reserved - Ignored, should be zero.
.1.. .... Storage can be backed anywhere.
..00 .... Storage should have residency of caller.
..01 .... Storage address must be 24 bits.
..11 .... Storage address valid to full 31 bits.
.... 1... Request is variable.
.... .1.. Storage should be on page boundary.
.... ..1. Request is unconditional.
.... ...1 Request is a FREEMAIN (0=GETMAIN - odd value=FREEMAIN, even=GETMAIN).
```

Data Collection for CICS SOS

- *Please don't send us dumps after SOS has occurred, the chances of finding the problem are likely to be zero and it just wastes time for both of us.*
- Get a dump ***at*** SOS by using this command:

```
CEMT S SYD(SM013n) ADD SYS MAX(1)
```

- SM0131 is for SOS-below and SM0133 is for SOS-above, if in doubt, use both.
- We normally ask you to use CETR to ensure that CICS is now set up to trace correctly:
  - Ensure that the "Master System Trace Flag" is ON (i.e. internal trace is active).
  - Define an "Internal Trace Table Size" of at least 4096K (this requires contiguous 31-bit Partition GETVIS and should *always* be your default size).
  - Use F4 and ensure that all components have level 1 tracing active.
  - Change AP and EI components to level 1-2 and press F3 to return.
- This equates to SIT STNTR=1, STNTRAP=(1,2),STNTREI=(1,2),TRTABSZ=4096.
- *We appreciate you sending console output and the whole CICS SYSLST for any type of problem reported in a PMR plus telling us which z/VSE version and release it is*.
- *Please, please, **don't** send us a formatted dump, we always need you to FTP the raw (i.e. binary) dump(s) as stored in SYSDUMP.*

Summary

- Ensure that your system is at a good service level.
- Remember to monitor free storage at the z/VSE and CICS levels on a regular basis and check usage deltas before you migrate changes from test or QA into production.
- While you are doing that, you might want to monitor other allocations that could cause problems such as BUFSIZE.
- Capture appropriate GETVIS command output for GETVIS usage and use DFH0STAT, DFHSTUP or vendor CICS monitoring software to track CICS DSA usage; make sure that the data is machine-readable and report threshold and limit conditions to avoid information overload.
- SDAID TRACE GETVIS is the most effective way of locating a GETVIS leak in conjunction with GETVIS xx,ALL output; at least two dumps that *show* increased usage will also be required.
- DSALIM and EDSALIM may be dynamically increased if there is sufficient contiguous free 24-bit GETVIS storage in 256K multiples and free 31-bit GETVIS in 1,024K multiples; if you have a active CEMT session, you can do that when CICS is at SOS.
- Decreasing over-allocated DSALIM and EDSALIM will release GETVIS storage for use by VSAM and other products.

Спасибо
Russian

धन्यवाद
Hindi

Bedankt
Nederlands

شكراً
Arabic

Merci
French

Obrigado
Brazilian Portuguese

English

Gracias!
Spanish

多谢
Simplified Chinese

Danke
German

多謝
Traditional Chinese

ありがとうございました
Japanese

감사합니다

Thank You

Please forward your questions or remarks to

poilmike@uk.ibm.com

z/VSE Live Virtual Classes

ADOBE® CONNECT™

z/VSE @ http://www.ibm.com/zvse/education/

LINUX + z/VM + z/VSE @ http://www.vm.ibm.com/education/lvc/

Read about upcoming LVCs on @ http://twitter.com/IBMzVSE

Join the LVC distribution list by sending a short mail to alina.glodowski@de.ibm.com