# The Evolution of Managing Real Storage and zFLASH's Impact

Session 16180
Room 406
4:30pm
Aug 6, 2014

George Handera
Handerag@aetna.com

aetna℠

# System/360 Announcement

The following is the text of an IBM Data Processing Division press release distributed on April 7, 1964.

A new generation of electronic computing equipment was introduced today by International Business Machines Corporation.

IBM Board Chairman Thomas J. Watson Jr. called the event the most important product announcement in the company's history.

The new equipment is known as the IBM System/360.

System/360 core storage memory capacity ranges from 8,000 characters of information to more than 8,000,000. Information storage devices linked to the system can store additional billions of characters of data and make them available for processing at varying speeds, depending on need.

- **Memory power.** A hierarchy of memories within System/360 makes information in core storage available at varying speeds. Small local store memories operate in as little as 200 billionths-of-a-second. Control memories operate in as little as 250 billionths-of-a-second. Powerful main memories - - containing up to 524,000 characters of information - - range from 2.5 millionths-of-a-second down to one millionth-of-a-second.

  A key development provides 8,000,000 characters in bulk core storage - - each character available in eight millionths-of-a-second and each at the direct command of a computer programmer. This is over sixty times more directly addressable characters than were previously available in IBM computers. The computer's historic limitations on memory size are overcome by this development.

# The Evolution of Managing Real Storage and zFLASH's Impact

## Generations of IBM360 -> 370 -> 390

The original 360 family was announced in 1964, and the lower midrange model 40 was the first to ship a year later. The most interesting version was model 67 (first shipped June 1966) which had hardware to support virtual memory. IBM had planned a special operating system for it (TSS/360), which they never managed to get to work well enough to be usable. Within IBM, model 67 was used with a system known as CP-67, which allowed a single 360/67 to simulate multiple machines of various models. This turned out to be very useful for developing operating systems. In the summer of 1970, IBM announced a family of machines with an enhanced instruction set, called System/370. These machines were all designed with virtual hardware similar to 360/67, and eventually all the operating systems were enhanced to take advantage of it in some way.

**When** System/360 was successful, other companies started making their machines similar to IBM's, but not close enough to actually run the same software. In 1970, however, Gene Amdahl (who had been the chief architect for the 360 family) started a company to build a series of machines that were direct clones of the 360-370 architecture, and later Hitachi followed suit. (The first Amdahl machine was shipped in 1975.)

Big, fast disk drives were one of the strengths of IBM. In 1973, the big mainframe **disk drive was model 3330-11**: 400 MB for $111,600 or **$279/MB**. By 1980, you could get the 3380: 2.5GB for $87,500 or **$35/MB**. **DRAM** prices were dropping, too: In 1979 the price was cut from $75,000/MB to **$50,000/MB**.

Through the 1970's and 1980's, the machines got bigger and faster, and multi-processor systems became common, but the basic architecture did not change. Around 1982, addresses were extended from 24 bits to 31 bits (370-XA), and in 1988 extensions were put in to support multiple address spaces (370-ESA). In 1990, the ES/9000 models came out with fiber-optical I/O channels (ESCON), and IBM began using the name System/390.

# The Evolution of Managing Real Storage and zFLASH's Impact

History
- Real
- Expanded
- Importance of Page Packs

Today
- Large Page
  - 1M
  - 1M Pageable
  - 2G
  - INCLUDE1MAFC
- zFLASH

Philosophy
- Memory allocation
- DB2 Buffers - Fix or don't define
- FLASH - What it's good for (Good move or bad - DB2)
- What's changing

# LFAREA

## LFAREA

```
LFAREA= {xM | xG | xT | x%}
        {[ 1M=(a [,b]) | 1M=(a% [,b%])] [,2G=(a [,b]) | ,2G=(a% [,b%])]
         [,prompt | ,noprompt]}
```

The LFAREA parameter specifies the amount of online real storage available at IPL to reserve for backing 1 MB pages and 2 GB pages. The *x*M, *x*G, *x*T, and *x*% syntax form reserves 1 MB pages, and the 1M= and 2G= syntax form reserves 1 MB and 2 GB pages. The two syntax forms cannot be combined. Each syntax specification uses a different formula for calculating percentage requests and the system limit, as described in "Request handling for the large frame area system limit" on page 449 and "Request handling for insufficiently contiguous online real storage" on page 450.

Note: z/OS 1.13 documentation does not cover ***INCLUDE1MAFC*** parameter.

# The Evolution of Managing Real Storage and zFLASH's Impact

## LFAREA (A,B)

1M=(*a*,*b*)

Specifies the number of 1 MB pages of online real storage to reserve in the large frame area. Up to eight decimal digits each can be specified for *a* and *b*. The value specified for *a* is the target number of pages, and the value specified for *b* is the minimum number. The system attempts to meet the request at or as near as possible up to the target number, but at no less than the minimum number. The value specified for *b* must be less than or equal to the value specified for *a*, and can be zero. A specification of 1M=(0,0) results in zero 1 MB pages being reserved. Once the LFAREA parameter has been processed, no additional amounts of storage are reserved later in an attempt to reach the target. Both a value and a percentage, such as 1M=(*a*,*b*%) or 1M=(*a*%,*b*), cannot be specified.

# The Evolution of Managing Real Storage and zFLASH's Impact

## Keyword Addition

**The V2R1 Exchange is making this APAR known to all members:**

APAR OA41968 applies to V2R1 and V1R13:

Added the INCLUDE1MAFC keyword to the operands for LFAREA in IEASYSxx.

Here are some LFAREA examples using INCLUDE1MAFC:

LFAREA=(64M,INCLUDE1MAFC)
- Note: Using the xM|xG|xT|x% syntax, INCLUDE1MAFC is a positional parameter and must be coded after the xM|xG|xT|x% specification.

LFAREA=(20%,INCLUDE1MAFC)
- Note: Using the xM|xG|xT|x% syntax, INCLUDE1MAFC is a positional parameter and must be coded after the xM|xG|xT|x% specification.

LFAREA=(1M=64,INCLUDE1MAFC)
- Note: Using the 1M= syntax, INCLUDE1MAFC can be specified anywhere within the parentheses.

LFAREA=(INCLUDE1MAFC,1M=20%,NOPROMPT)
- Note: Using the 1M= syntax, INCLUDE1MAFC can be specified anywhere within the parentheses.

## Memory Philosophy

- LPARs should be memory rich, CPU management of memory is costly
- I define specific values for LFAREA allocations
  - RMF Monitor 3 provides great insight
- LFAREA is specified in the IEASYSxx member
  - Requires an IPL
- Remember that frequently used items like DB2 bufferpools will always reside in memory so fixing the frames really has no cost… but there are significant savings

# The Evolution of Managing Real Storage and zFLASH's Impact

Storage constrained environment ***without*** the ***INCLUD1MAFC*** APAR. The APAR changes the harvesting of unused LFAREA pages to being aggressive when the LFAREA parmlib member has ***INCLUDE1MAFC*** specified.

**Without** ***INCLUDE1MAFC*** specified

```
RESPONSE=AE91
 IAR019I  14.31.59 DISPLAY VIRTSTOR 974
  SOURCE =  91
  TOTAL LFAREA = 4096M
  LFAREA AVAILABLE = 3941M
  LFAREA ALLOCATED (1M) = 143M
  LFAREA ALLOCATED (4K) = 12M
  MAX LFAREA ALLOCATED (1M) = 433M
  MAX LFAREA ALLOCATED (4K) = 66M
```

# The Evolution of Managing Real Storage and zFLASH's Impact

Storage constrained environment ***with*** the ***INCLUD1MAFC*** APAR. The APAR changes the harvesting of unused LFAREA pages to being aggressive when the LFAREA parmlib member has ***INCLUDE1MAFC*** specified.

## ***INCLUDE1MAFC*** **specified**

```
-D VIRTSTOR,LFAREA
 IAR019I  14.39.48 DISPLAY VIRTSTOR 618
  SOURCE =  91
  TOTAL LFAREA = 4096M , 0G
  LFAREA AVAILABLE = 19M , 0G
  LFAREA ALLOCATED (1M) = 1063M
  LFAREA ALLOCATED (4K) = 3014M
  MAX LFAREA ALLOCATED (1M) = 1064M
  MAX LFAREA ALLOCATED (4K) = 3033M
  LFAREA ALLOCATED (PAGEABLE1M) = 0M
  MAX LFAREA ALLOCATED (PAGEABLE1M) = 0M
  LFAREA ALLOCATED NUMBER OF 2G PAGES = 0
  MAX LFAREA ALLOCATED NUMBER OF 2G PAGES = 0
```

# LFAREA at IPL on a z196

LFAREA=(20%,INCLUDE1MAFC) specified in IEASYSxx member

```
    00000290  IAR040I REAL STORAGE AMOUNTS: 160
160 00000290    TOTAL AVAILABLE ONLINE: 30G
160 00000290      LFAREA LIMIT FOR xM, xG, OR xT       : 22G
160 00000290      LFAREA LIMIT FOR SUM OF 1M= AND 2G= : 21299M
160 00000290      LFAREA LIMIT FOR 2GB PAGES FOR 2G=  : 0 (NOT SUPPORTED)
    00000290  IAR048I LFAREA=(20%,INCLUDE1MAFC) WAS PROCESSED WHICH RESULTED IN
              4096 1MB PAGES AND 0 2GB PAGES.
    00000290  IAR013I 8G STORAGE IS RECONFIGURABLE
```

## LFAREA - 2G page specified on z196

```
IAR041I LFAREA=(1M=20%,2G=2,INCLUDE1MAFC) WAS SPECIFIED BUT 2GB PAGE
SUPPORT IS NOT AVAILABLE.
IAR045I VALID RANGE FOR LFAREA xM, xG, xT IS 0M TO 22G, OR 0% TO 80%.
A 163
MINIMUM OF 7% MUST BE SPECIFIED TO RESERVE AT LEAST ONE 1MB PAGE.
IAR045I VALID RANGE FOR LFAREA 1M= IS 0 TO 21299, OR 0% TO 80%. A 164
MINIMUM OF 1% MUST BE SPECIFIED TO RESERVE AT LEAST ONE 1MB PAGE.
IAR045I VALID VALUE FOR LFAREA 2G= IS 0, OR 0%. HARDWARE SUPPORT
UNAVAILABLE FOR 2GB PAGES.
```

```
IAR047I AT THE FOLLOWING PROMPT, SPECIFY THE COMPLETE LFAREA
PARAMETER OR PRESS ENTER FOR ZERO 1MB AND 2GB PAGES.
IEA341A RESPECIFY LFAREA PARM OR PRESS ENTER
IEE600I REPLY TO 00 IS;LFAREA=(1M=20%,INCLUDE1MAFC)
IAR048I LFAREA=(1M=20%,INCLUDE1MAFC) WAS PROCESSED WHICH RESULTED IN
5324 1MB PAGES AND 0 2GB PAGES.
```

# The Evolution of Managing Real Storage and zFLASH's Impact

```
IAR048I LFAREA=(1M=20%,2G=2,INCLUDE1MAFC,NOPROMPT) WAS PROCESSED
WHICH 246
RESULTED IN 0 1MB PAGES AND 0 2GB PAGES DUE TO: NOPROMPT SPECIFIED AND
NO HARDWARE SUPPORT.
```

- LFAREA specifications that can not be satisfied - hold up the IPL if NOPROMPT ***is not*** specified
- If NOPROMPT is specified, the IPL continues… but the LFAREA parameters that could be satisfied are also ignored

# The Evolution of Managing Real Storage and zFLASH's Impact

## On an EC12

```
 DATA SET LAST UPDATED AT 19:23:17 ON 10/03/2013 (GMT)
IEA940I THE FOLLOWING PAGE DATA SETS ARE IN USE:
        PLPA ..... - SYS1.PG91AA.LPA
        COMMON ..... - SYS1.PG91AA.CSA
        LOCAL ..... - SYS1.PG91AB.LOCAL1
        LOCAL ..... - SYS1.PG91AC.LOCAL1
        LOCAL ..... - SYS1.PG91AD.LOCAL1
        LOCAL ..... - SYS1.PG91AE.LOCAL1
        LOCAL ..... - SYS1.PG91AF.LOCAL1
        LOCAL ..... - SYS1.PG91AG.LOCAL1
        LOCAL ..... - SYS1.PG91AH.LOCAL1
        LOCAL ..... - SYS1.PG91AI.LOCAL1
IEE252I MEMBER IEASVC91 FOUND IN SYS1.PARMLIB.AEPLEX0A
IAR040I REAL STORAGE AMOUNTS: 241
  TOTAL AVAILABLE ONLINE: 11G
    LFAREA LIMIT FOR xM, xG, OR xT       : 6963M
    LFAREA LIMIT FOR SUM OF 1M= AND 2G= : 5734M
    LFAREA LIMIT FOR 2GB PAGES FOR 2G=   : 2
IAR048I LFAREA=(1M=20%,2G=2,INCLUDE1MAFC) WAS PROCESSED WHICH
RESULTED IN 1433 1MB PAGES AND 2 2GB PAGES.
IAR013I 9G STORAGE IS RECONFIGURABLE
IAR031I USE OF STORAGE-CLASS MEMORY FOR PAGING IS ENABLED - PAGESCM=AL
L       , ONLINE=00000000M
```

# The Evolution of Managing Real Storage and zFLASH's Impact

## Memory Management

CPU should not be traded off for Memory

Large Page allocations should back the DB2 BPs at a minimum

Without zFLASH

- Keep your AFC well stocked (I shoot for 10-15GB backing all address space virtual)
- Watch the SYSTEM UIC - know when it drops (hint sorts)

With zFLASH

- Expand your memory use (DB2 BP's)
- Keep healthy AFC (2 or more GB - after address spaces are trimmed to only keep WSS frames in storage)
- Active paging should only be a result of the initial trimming, it should not be an ongoing event

# FLASH

Flash Memory aka SCM (Storage Class Memory) will be available from 1 to 4 flash features. Each has 1 feature providing 1.4 TB of user storage.

- Goals
  - Paging Relief on Test LPARs
  - Implement 1MB paging for Websphere clocking the Websphere Memory issues

- SOFTWARE
  - z/OS 1.13 - Requires a web deliverable download of support
  - z/OS 2.1 - Support delivered in base

- *Parmlib (IEASYSxx*)
  - PAGE= (new Option *NONE*) discuss warm start and options
  - PAGESCM=ALL (flash used only for paging… it is the default value - fine for today but hoping)

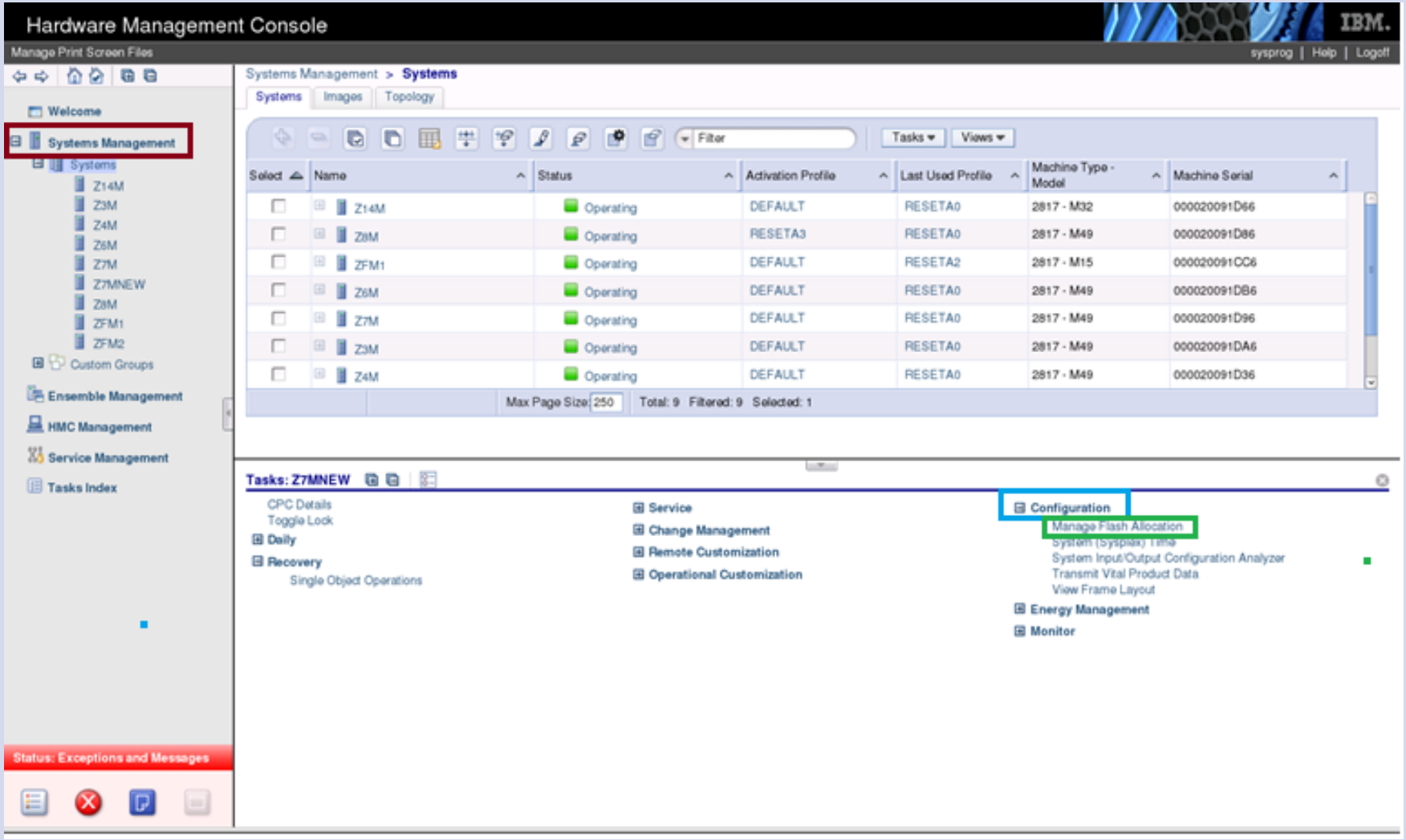
- HMC - Defining the flash to an LPAR

# The Evolution of Managing Real Storage and zFLASH's Impact

- Select the SYSTEMS MANAGEMENT tag

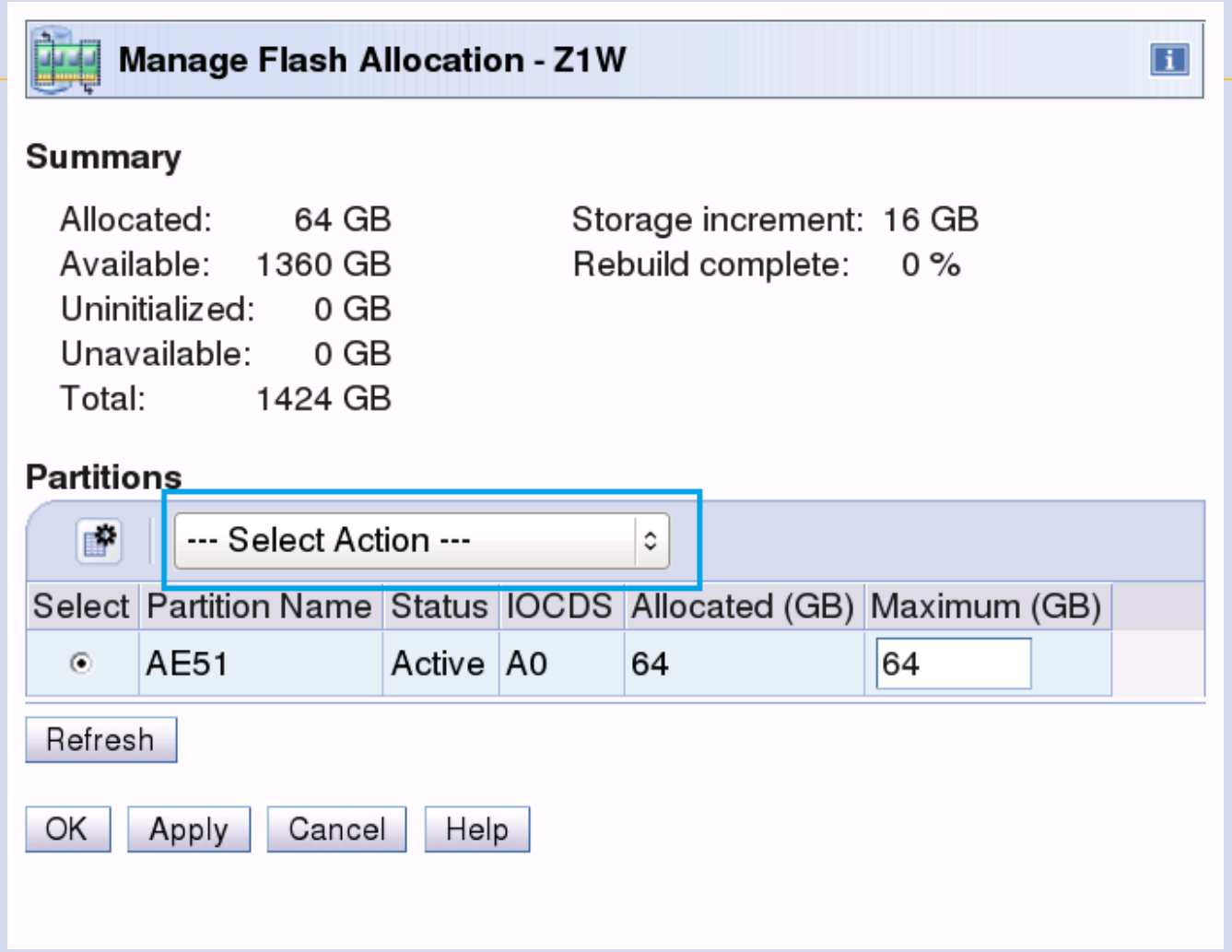
  Select the radio button for the CPC desired (Z1W)

  Under CONFIGURATION select MANAGE FLASH



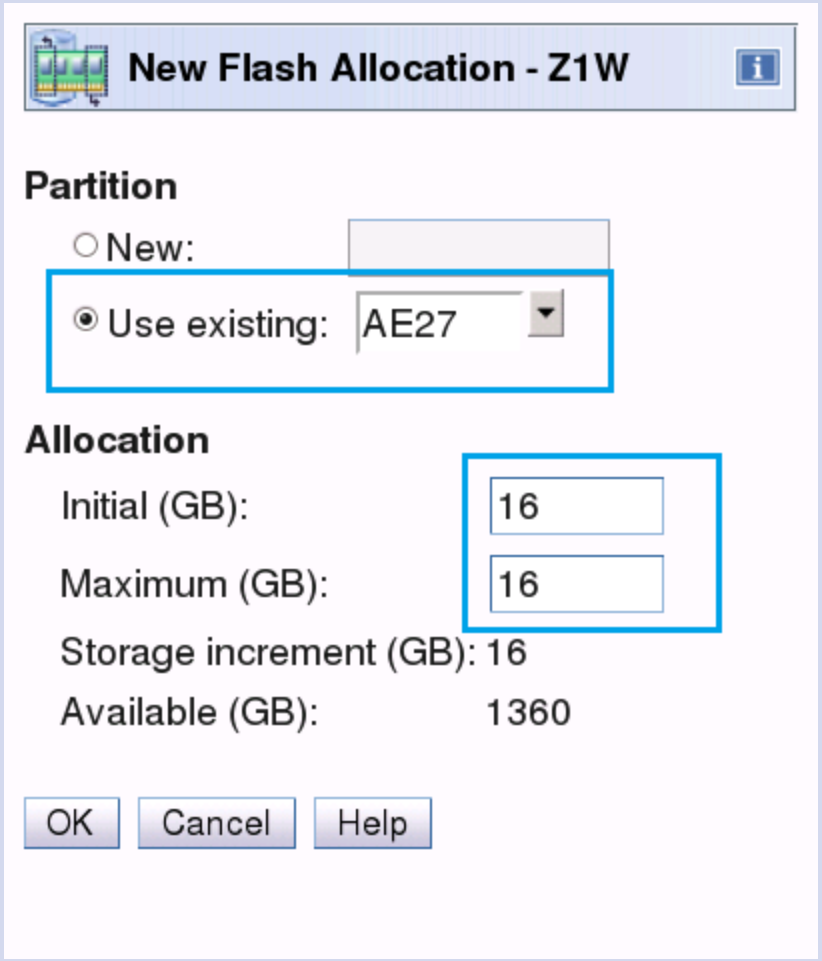
# The Evolution of Managing Real Storage and zFLASH's Impact

Select *ADD ALLOCATION* from the scroll down SELECT ACTION box



# The Evolution of Managing Real Storage and zFLASH's Impact

Use Radio button on *Use existing*
Select LPAR from Pull down box



# FLASH Commands

```
-D ASM,SCM
 IEE207I 19.23.52 DISPLAY ASM 220
 STATUS       FULL                SIZE              USED          IN-ERROR
 IN-USE        27%          67,108,864        18,741,379                 0
```

```
-D M=SCM(DETAIL)
 IEE174I 19.25.57 DISPLAY M 539
 STORAGE-CLASS MEMORY STATUS - INCREMENT DETAIL
 256G DEFINED
 ADDRESS  IN USE  STATUS
      0G      0%  ONLINE
     16G      0%  ONLINE
     32G      0%  ONLINE
     48G      0%  ONLINE
     64G      0%  ONLINE
     80G      0%  ONLINE
     96G      0%  ONLINE
    112G     89%  ONLINE
    128G     90%  ONLINE
    144G     91%  ONLINE
    160G     92%  ONLINE
    176G     63%  ONLINE
    192G      0%  ONLINE
    208G      0%  ONLINE
    224G     27%  ONLINE
    240G      0%  ONLINE
 ONLINE: 256G  OFFLINE-AVAILABLE: 0G  PENDING OFFLINE: 0G
 28% IN USE
 SCM INCREMENT SIZE IS 16G
```

FLASH Config commands
May be issued to config on and config off Flash in 16GB increments.

I believe in allocating FLASH in larger than required segments.

1.4 TB of FLASH goes a long way. If you need that much order more

Up to 4 - 1.4TB features may be ordered. Max of 2 features per drawer.

# The Evolution of Managing Real Storage and zFLASH's Impact

## RMFIII - Storage Use of Memory Objects - storage constrained system (1 of 2)

```
                        RMF V2R1   Storage Memory Objects          Line 1 of 690
Command ===>                                                    Scroll ===> CSR

Samples: 60      System:          Date:           Time:           Range:       Sec

------------------------------ System Summary -------------------------------
---MemObj---    ---Frames---     -1MB MemObj-     --1MB Fixed--   -1MB Pageable-
Shared    25    Shared  666K     Total     16     Total   1900    Initial 15992
Common   488    Common 38235     Common     1     Common     8    Dynamic     0
                %Used   50.1                      %Used    100    %Used     100
-----------------------------------------------------------------------------

           Service        ---- Memory Objects ---  -1MB Frames-  ----- Bytes -----
Jobname  C Class     ASID Total   Comm    Shr  1 MB Fixed Pgable Total   Comm    Shr

WQSN5BS  S STCWASRV 0722   774      0      0     0     0   1219 34.6G     0       0
WQIS4BS  S STCWASRV 0845   765      0      0     0     0    838 33.9G     0       0
WQIS4BS  S STCWASRV 0824   757      0      0     0     0    860 33.6G     0       0
WSSN5AS  S STCWASRV 0924   718      0      0     0     0     28 32.2G     0       0
WQPIS1DA S STCWASRV 0905   715      0      0     0     0      0 30.9G     0       0
WSPIS1AA S STCWASRV 0843   709      0      0     0     0      0 30.7G     0       0
WQPIS1FA S STCWASRV 0903   708      0      0     0     0      0 30.6G     0       0
WSPIS1CA S STCWASRV 0907   707      0      0     0     0      0 30.6G     0       0
```

# The Evolution of Managing Real Storage and zFLASH's Impact

## RMFIII - Storage Use of Memory Objects - storage constrained system (2 of 2)

```
                        RMF V2R1   Storage Memory Objects        Line 66 of 690
Command ===>                                                  Scroll ===> CSR

Samples: 60       System: xxxx  Date: xx/xx/xx  Time: xx.xx.xx  Range: 60    Sec

------------------------------ System Summary -------------------------------
---MemObj---    ---Frames---    -1MB MemObj-    --1MB Fixed--   -1MB Pageable-
Shared    25    Shared  666K    Total     16    Total   1900    Initial 15992
Common   488    Common 38235    Common     1    Common     8    Dynamic     0
                %Used   50.1                   %Used    100    %Used     100
-----------------------------------------------------------------------------

           Service        ---- Memory Objects ---  -1MB Frames-  ----- Bytes -----
Jobname  C Class     ASID Total   Comm    Shr  1 MB Fixed Pgable Total  Comm    Shr

WSIS8AS  S STCWASRV 0957   529      0      0     0     0      4 24.2G     0      0
WSIN4AS  S STCWASRV 0830   528      0      0     0     0    620 24.1G     0      0
WQPIS1HS S STCWASRV 0874   526      0      0     0     0      0 24.1G     0      0
WSIS9AS  S STCWASRV 0831   522      0      0     0     0    638 24.4G     0      0
WSIS6AS  S STCWASRV 0958   516      0      0     0     0      4 23.6G     0      0
WSDP1AS  S STCWASRV 0940   464      0      0     0     0      3 20.9G     0      0
WQNA1B   S STCWASCR 0684   455      1      0     1   513     11 19.8G  1024K      0
WQSN3B   S STCWASCR 0707   443      1      0     0     0    140 19.0G  1024K      0
WSSN3A   S STCWASCR 0914   434      1      0     0     0      0 18.8G  1024K      0
```

# The Evolution of Managing Real Storage and zFLASH's Impact

## RMFIII - Storage Use of Memory Objects - storage $RICH$ system

```
                        RMF V2R1   Storage Memory Objects          Line 1 of 134
Command ===>                                                   Scroll ===> CSR

Samples: 120     System:        Date:           Time:            Range:       Sec

------------------------------ System Summary -------------------------------
---MemObj---    ---Frames---    -1MB MemObj-    --1MB Fixed--   -1MB Pageable-
Shared    10    Shared  750K    Total    197    Total  45000    Initial 11688
Common   132    Common 25996    Common     1    Common     8    Dynamic   322
               %Used   22.4                    %Used   98.4    %Used     100
-----------------------------------------------------------------------------

           Service        ---- Memory Objects ---  -1MB Frames-  -----  Bytes  -----
Jobname  C Class     ASID Total   Comm    Shr  1 MB Fixed Pgable Total   Comm    Shr

WPPIS1BA S STCWASCR 0264    800      0      0     4  1046      0 34.5G      0      0
WPPIS1BS S STCWASRV 0263    774      0      0     4  4172      0 33.5G      0      0
WPIS4DS  S STCWASRV 0211    771      0      0     1  1537      0 34.2G      0      0
WPIS4DS  S STCWASRV 0269    741      0      0     1  1537      0 32.8G      0      0
WPIN1DS  S STCWASRV 0258    680      0      0     1  1537    128 30.1G      0      0
WPIS5DS  S STCWASRV 0272    663      0      0     1  2049    128 29.7G      0      0
WPIS5DS  S STCWASRV 0212    661      0      0     1  2049    128 29.6G      0      0
WPSN2DS  S STCWASRV 0270    660      0      0     1  1537     56 29.4G      0      0
WPIS5DS  S STCWASRV 0277    660      0      0     1  2049    128 29.6G      0      0
WPSN5DS  S STCWASRV 0208    660      0      0     1  1537     42 29.6G      0      0
WPIS5DS  S STCWASRV 0275    658      0      0     1  2049    128 29.6G      0      0
```

# The Evolution of Managing Real Storage and zFLASH's Impact

## RMFIII - Storage Frames - storage constrained system

RMF V2R1 Storage Frames Line 1
Command ===> Scroll ==

Samples: 60 System: Date: Time: Range:

| Jobname | C | Service Class | Cr | -- Frame Occup.-- TOTAL | ACTV | IDLE | - Active Frames - WSET | FIXED | DIV | AUX SLOTS | PGIN RATE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DBBADBM1 | S | STCHI | | 886K | 886K | 0 | 886K | 258K | 12562 | 491K | 5 |
| DBAADBM1 | S | STCHI | | 830K | 830K | 0 | 830K | 228K | 12328 | 688K | 10 |
| ZFS | S | SYSSTC | | 805K | 805K | 0 | 805K | 30350 | 0 | 478K | 0 |
| WQPIS1BS | S | STCWASRV | | 780K | 780K | 0 | 780K | 3988 | 205 | 1743 | 0 |
| WQIS5BS | S | STCWASRV | | 541K | 541K | 0 | 541K | 1800 | 161 | 102 | 0 |
| WSSN5AS | S | STCWASRV | | 522K | 522K | 0 | 522K | 3409 | 161 | 74504 | 0 |
| WQSN5BS | S | STCWASRV | | 512K | 512K | 0 | 512K | 2332 | 154 | 150K | 0 |
| WSPIS1AS | S | STCWASRV | | 497K | 497K | 0 | 497K | 3288 | 182 | 426K | 0 |
| WSPIS1ES | S | STCWASRV | | 490K | 490K | 0 | 490K | 3320 | 186 | 145K | 0 |
| WQPIS1DS | S | STCWASRV | | 481K | 481K | 0 | 481K | 3292 | 198 | 194K | 0 |
| WQPIS1FS | S | STCWASRV | | 479K | 479K | 0 | 479K | 3229 | 195 | 248K | 0 |
| WQIS8BS | S | STCWASRV | | 476K | 476K | 0 | 476K | 2866 | 162 | 0 | 0 |
| WSPIS1CS | S | STCWASRV | | 469K | 469K | 0 | 469K | 3251 | 198 | 220K | 0 |

# The Evolution of Managing Real Storage and zFLASH's Impact

## RMFIII - Storage Frames - storage $RICH$ system (2 of 2)

RMF V2R1 Storage Frames Line

Command ===> Scroll =

Samples: 120 System: Date: Time: Range:

| Jobname | C | Service Class | Cr | -- Frame Occup.-- TOTAL | ACTV | IDLE | - Active Frames - WSET | FIXED | DIV | AUX SLOTS | PGIN RATE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| WPPIS1BS | S | STCWASRV | | 1802K | 1802K | 0 | 1802K | 1072K | 208 | 0 | 0 |
| DB3IDBM1 | S | STCHI | | 1385K | 1385K | 0 | 1385K | 1062K | 12570 | 0 | 0 |
| DB3LDBM1 | S | STCHI | | 1040K | 1040K | 0 | 1040K | 649K | 10392 | 0 | 0 |
| WPIS5DS | S | STCWASRV | | 859K | 859K | 0 | 859K | 527K | 162 | 0 | 0 |
| WPIS5DS | S | STCWASRV | | 853K | 853K | 0 | 853K | 527K | 162 | 0 | 0 |
| WPIS5DS | S | STCWASRV | | 850K | 850K | 0 | 850K | 527K | 162 | 0 | 0 |
| WPIS5DS | S | STCWASRV | | 849K | 849K | 0 | 849K | 527K | 162 | 1 | 0 |
| WPIS5DS | S | STCWASRV | | 843K | 843K | 0 | 843K | 527K | 162 | 0 | 0 |
| ZFS | S | SYSSTC | | 732K | 732K | 0 | 732K | 29304 | 0 | 0 | 0 |
| WPIN1DS | S | STCWASRV | | 683K | 683K | 0 | 683K | 396K | 162 | 2 | 0 |
| WPIS4DS | S | STCWASRV | | 643K | 643K | 0 | 643K | 396K | 207 | 0 | 0 |
| WPIS4DS | S | STCWASRV | | 640K | 640K | 0 | 640K | 396K | 208 | 0 | 0 |
| WPSN2DS | S | STCWASRV | | 636K | 636K | 0 | 636K | 395K | 208 | 0 | 0 |

# The Evolution of Managing Real Storage and zFLASH's Impact

## Post Processor - Paging Activity

FRAME AND SLOT COUNTS

(364 SAMPLES)

| CENTRAL STORAGE FRAMES | TOTAL | AVAILABLE | SQA | LPA | CSA | LSQA | REGIONS+SWA | HV SHARED | HV COMMON |
|---|---|---|---|---|---|---|---|---|---|
| MIN | 37945344 | 72,599 | 33,125 | 17,530 | 40,538 | 448,821 | 36293424 | 852,014 | 26,491 |
| MAX | 37945344 | 230,425 | 33,366 | 17,548 | 42,525 | 454,362 | 36454912 | 884,467 | 26,510 |
| AVG | 37945344 | 121,541 | 33,236 | 17,542 | 41,473 | 451,833 | 36397696 | 862,710 | 26,506 |

| FIXED FRAMES | TOTAL | NUCLEUS | SQA | LPA | CSA | LSQA | REGIONS+SWA | <16 MB | 16MB-2GB |
|---|---|---|---|---|---|---|---|---|---|
| MIN | 1,486,126 | 3,043 | 24,624 | 144 | 20,941 | 180,318 | 1,256,874 | 883 | 98,365 |
| MAX | 1,501,798 | 3,043 | 24,868 | 145 | 20,941 | 184,177 | 1,270,628 | 934 | 100,859 |
| AVG | 1,489,631 | 3,043 | 24,732 | 144 | 20,941 | 182,051 | 1,258,719 | 890 | 99,143 |

| SHARED FRAMES / SLOTS | TOTAL | CENTRAL STORAGE | FIXED TOT | FIXED BEL | AUX DASD | AUX SCM |
|---|---|---|---|---|---|---|
| MIN | 34,877 | 8,143 | 218 | 0 | 0 | 6,455 |
| MAX | 34,930 | 9,004 | 221 | 0 | 0 | 6,819 |
| AVG | 34,892 | 8,576 | 220 | 0 | 0 | 6,617 |

| LOCAL PAGE DATA SET SLOTS | TOTAL | AVAILABLE | BAD | NON-VIO | VIO |
|---|---|---|---|---|---|
| MIN | 64349984 | 63960768 | 0 | 389,156 | 0 |
| MAX | 64349984 | 63960832 | 0 | 389,221 | 0 |
| AVG | 64349984 | 63960784 | 0 | 389,191 | 0 |

| SCM PAGING BLOCKS | TOTAL | AVAILABLE | BAD | IN-USE |
|---|---|---|---|---|
| MIN | 67108864 | 46856464 | 0 | 19819536 |
| MAX | 67108864 | 47289328 | 0 | 20252400 |
| AVG | 67108864 | 46982304 | 0 | 20126560 |

# The Evolution of Managing Real Storage and zFLASH's Impact

## Post Processor - Memory Objects

P A G I N G  A C T I V I T Y

z/OS V2R1 SYSTEM ID AE32 START 08/04/2014-09.59.00 INTERVAL 000.59.59
RPT VERSION V2R1 RMF END 08/04/2014-10.59.00 CYCLE 1.000 SECONDS

OPT = IEAOPT32 LFAREA SIZE = 2306867K MEMORY OBJECTS AND HIGH VIRTUAL STORAGE FRAMES

| MEMORY OBJECTS | COMMON | SHARED | 1 MB |
|---|---|---|---|
| MIN | 396 | 26 | 33 |
| MAX | 396 | 26 | 33 |
| AVG | 396 | 26 | 33 |

| 1 MB FRAMES | FIXED | | | PAGEABLE | | |
|---|---|---|---|---|---|---|
| | TOTAL | AVAILABLE | IN-USE | TOTAL | AVAILABLE | IN-USE |
| MIN | 2,200 | 0 | 2,200 | 15,992 | 0 | 15,990 |
| MAX | 2,200 | 0 | 2,200 | 15,992 | 2 | 15,992 |
| AVG | 2,200 | 0 | 2,200 | 15,992 | 0 | 15,992 |

| HIGH SHARED FRAMES | TOTAL | CENTRAL STORAGE | AUX DASD | AUX SCM |
|---|---|---|---|---|
| MIN | 136902.1M | 852,014 | 0 | 333,859 |
| MAX | 136902.1M | 884,467 | 0 | 345,516 |
| AVG | 136902.1M | 862,710 | 0 | 342,207 |

| HIGH COMMON FRAMES | TOTAL | CENTRAL STORAGE | FIXED 4K | AUX DASD | AUX SCM |
|---|---|---|---|---|---|
| MIN | 17301504 | 26,491 | 3,254 | 0 | 7,262 |
| MAX | 17301504 | 26,510 | 3,254 | 0 | 7,262 |
| AVG | 17301504 | 26,506 | 3,254 | 0 | 7,262 |

# The Evolution of Managing Real Storage and zFLASH's Impact

## Post Processor - Page Dataset Activity

P A G E  D A T A  S E T  A C T I V I T Y

z/OS V2R1 SYSTEM ID AE31 START 08/04/2014-09.59.00 INTERVAL 001.00.00
RPT VERSION V2R1 RMF END 08/04/2014-10.59.00 CYCLE 1.000 SECONDS

NUMBER OF SAMPLES = 3,600

PAGE DATA SET AND SCM USAGE

| PAGE SPACE TYPE | VOLUME SERIAL | DEV NUM | DEVICE TYPE | SLOTS ALLOC | SLOTS USED MIN | SLOTS USED MAX | SLOTS USED AVG | BAD SLOTS | % IN USE | PAGE TRANS TIME | NUMBER IO REQ | PAGES XFER'D | V I O | DATA SET NAME |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PLPA | PG31B0 | 9177 | 33903 | 89999 | 11915 | 11915 | 11915 | 0 | 0.00 | 0.000 | 0 | 0 | | SYS1.PG31B0.LPA |
| COMMON | PG31B0 | 9177 | 33903 | 89999 | 37 | 37 | 37 | 0 | 0.00 | 0.000 | 0 | 0 | | SYS1.PG31B0.CSA |
| LOCAL | PG31B1 | 9BFB | 33909 | 5850K | 19189 | 19339 | 19325 | 0 | 0.00 | 0.000 | 26 | 196 | Y | SYS1.PG31B1.LOCAL1 |
| LOCAL | PG31B2 | 9B20 | 33909 | 5850K | 19030 | 19155 | 19142 | 0 | 0.00 | 0.000 | 39 | 175 | Y | SYS1.PG31B2.LOCAL1 |
| LOCAL | PG31B3 | 9B21 | 33909 | 5850K | 17967 | 18133 | 18120 | 0 | 0.00 | 0.000 | 37 | 223 | Y | SYS1.PG31B3.LOCAL1 |
| LOCAL | PG31B4 | 9BFE | 33909 | 5850K | 19330 | 19496 | 19481 | 0 | 0.00 | 0.000 | 36 | 215 | Y | SYS1.PG31B4.LOCAL1 |
| LOCAL | PG31B5 | 9BFF | 33909 | 5850K | 18315 | 18451 | 18438 | 0 | 0.00 | 0.000 | 44 | 204 | Y | SYS1.PG31B5.LOCAL1 |
| LOCAL | PG31B6 | D9ED | 3390A | 5850K | 17742 | 17938 | 17918 | 0 | 0.00 | 0.000 | 44 | 244 | Y | SYS1.PG31B6.LOCAL1 |
| LOCAL | PG31B7 | D9F1 | 3390A | 5850K | 18497 | 18737 | 18714 | 0 | 0.00 | 0.000 | 47 | 320 | Y | SYS1.PG31B7.LOCAL1 |
| LOCAL | PG31B8 | 9C02 | 33909 | 5850K | 19472 | 19682 | 19662 | 0 | 0.03 | 0.003 | 61 | 309 | Y | SYS1.PG31B8.LOCAL1 |
| LOCAL | PG31B9 | 9C11 | 33909 | 5850K | 19591 | 19801 | 19782 | 0 | 0.00 | 0.000 | 66 | 310 | Y | SYS1.PG31B9.LOCAL1 |
| LOCAL | PG31BA | 9BA6 | 33909 | 5850K | 19581 | 19775 | 19755 | 0 | 0.00 | 0.000 | 43 | 251 | Y | SYS1.PG31BA.LOCAL1 |
| LOCAL | PG31BB | 9BA7 | 33909 | 5850K | 19103 | 19373 | 19345 | 0 | 0.03 | 0.003 | 70 | 379 | Y | SYS1.PG31BB.LOCAL1 |
| LOCAL | PG31BC | 9AAA | 33909 | 5850K | 18886 | 19033 | 19018 | 0 | 0.03 | 0.005 | 35 | 210 | Y | SYS1.PG31BC.LOCAL1 |
| SCM | N/A | N/A | N/A | 67109K | 23306K | 23539K | 23447K | 0 | 0.86 | 0.000 | 78,442 | 25,291 | | N/A |

## zFLASH - Other Uses

DBAR

- zFLASH can expand the scope of the storage available on the CBU machine
- Ideal for DBAR tests
- Provides a survival mechanism for an actual disaster event till decision is made to return or upgrade.

Altering the Philosophy of memory allocation

- Increasing the active use of memory

Hardware Failure events

## Hardware Failures

Problem Recovery

In the event of a system check stop – example Book Failure

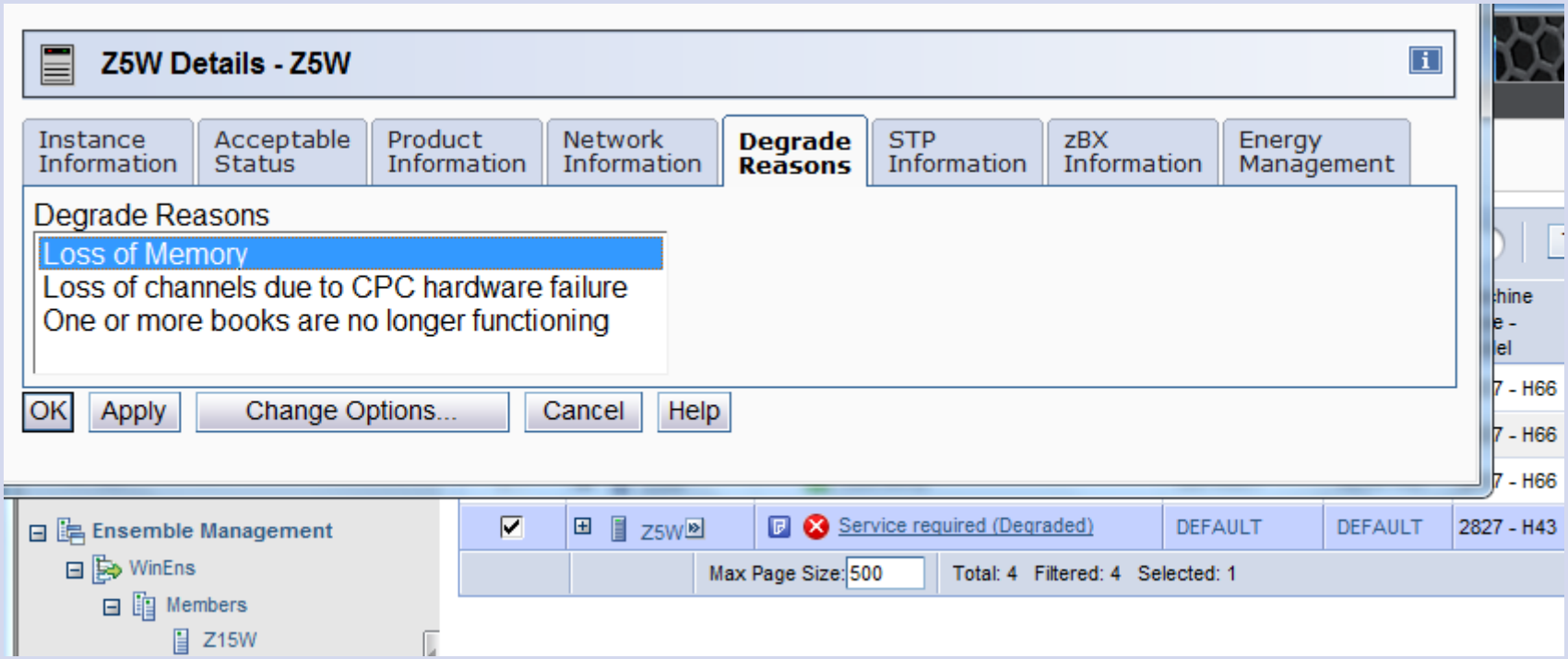
– Whole system is affected - machine down

– Depending on the type of failure a processor book may become "fenced offline"

– Resources on the fenced book are unavailable
- Processors, Memory, I/O interconnect (coupling links if present)
- LPAR activations will likely fail at some point

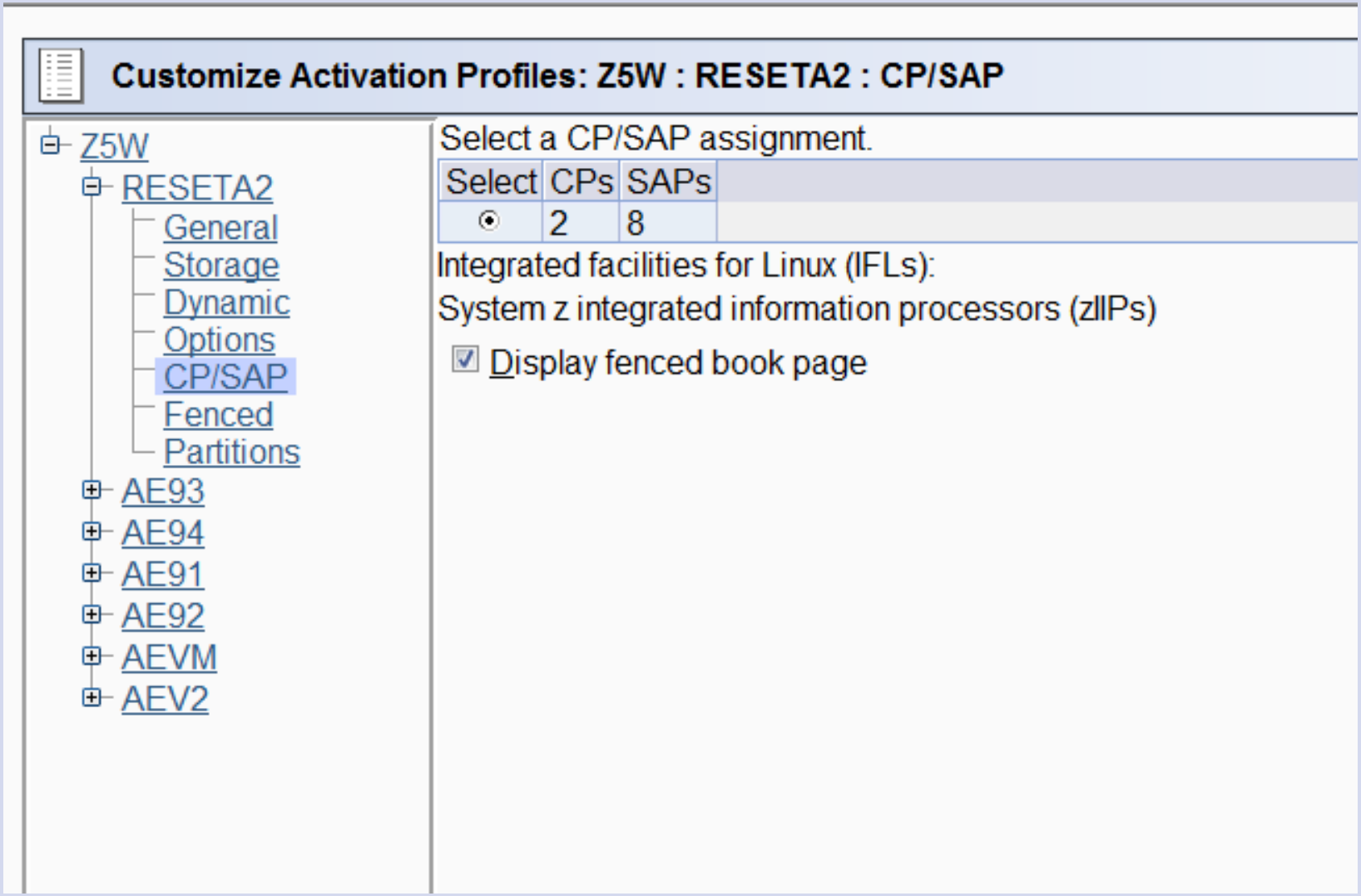
# The Evolution of Managing Real Storage and zFLASH's Impact

## Fenced Book - Service Message



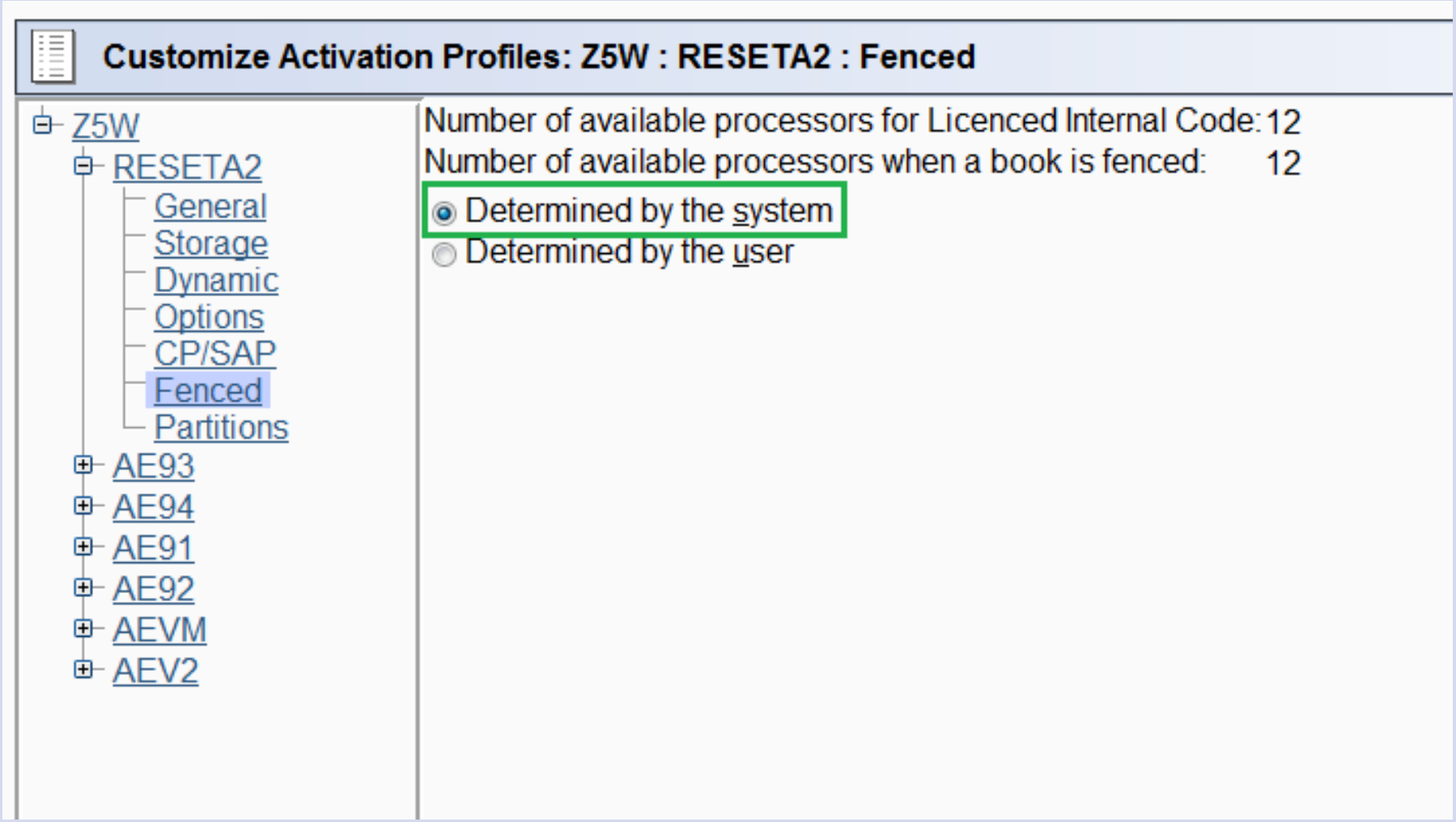
# The Evolution of Managing Real Storage and zFLASH's Impact

## Fenced Book - Checkbox

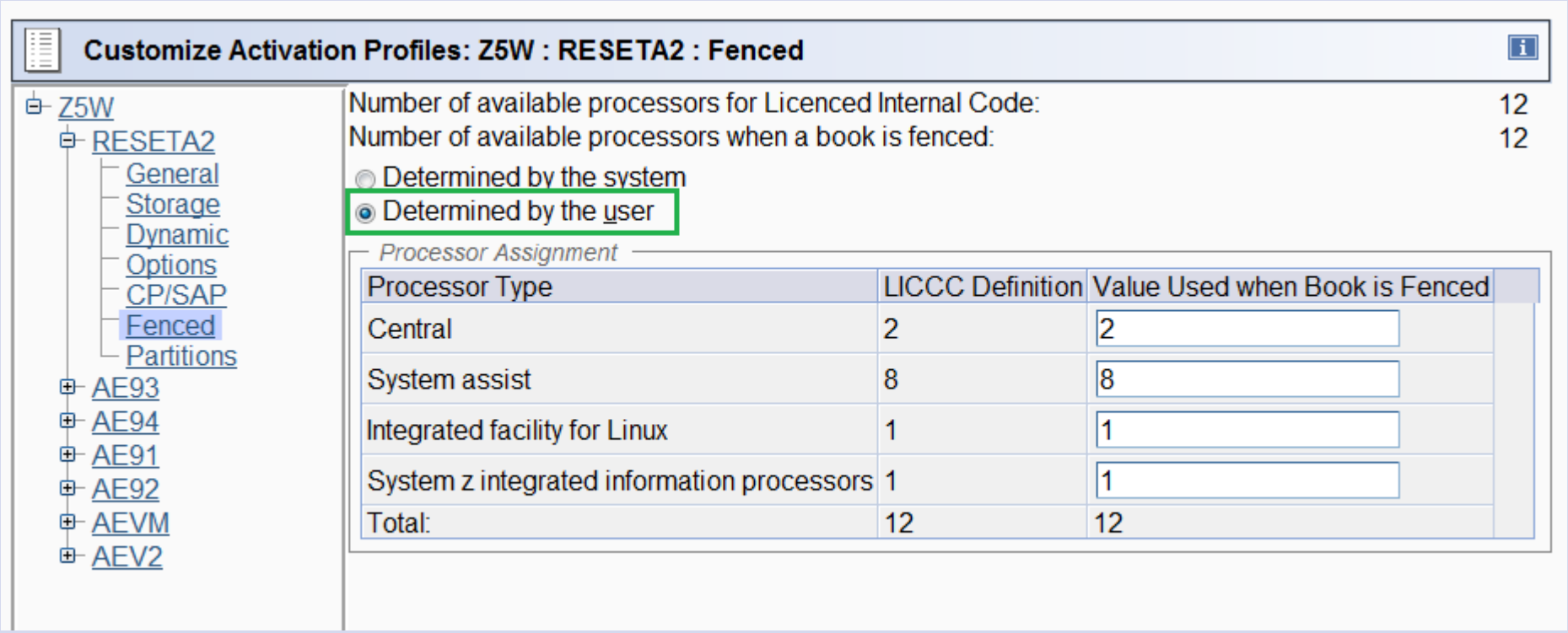


# The Evolution of Managing Real Storage and zFLASH's Impact

## Fenced Book box

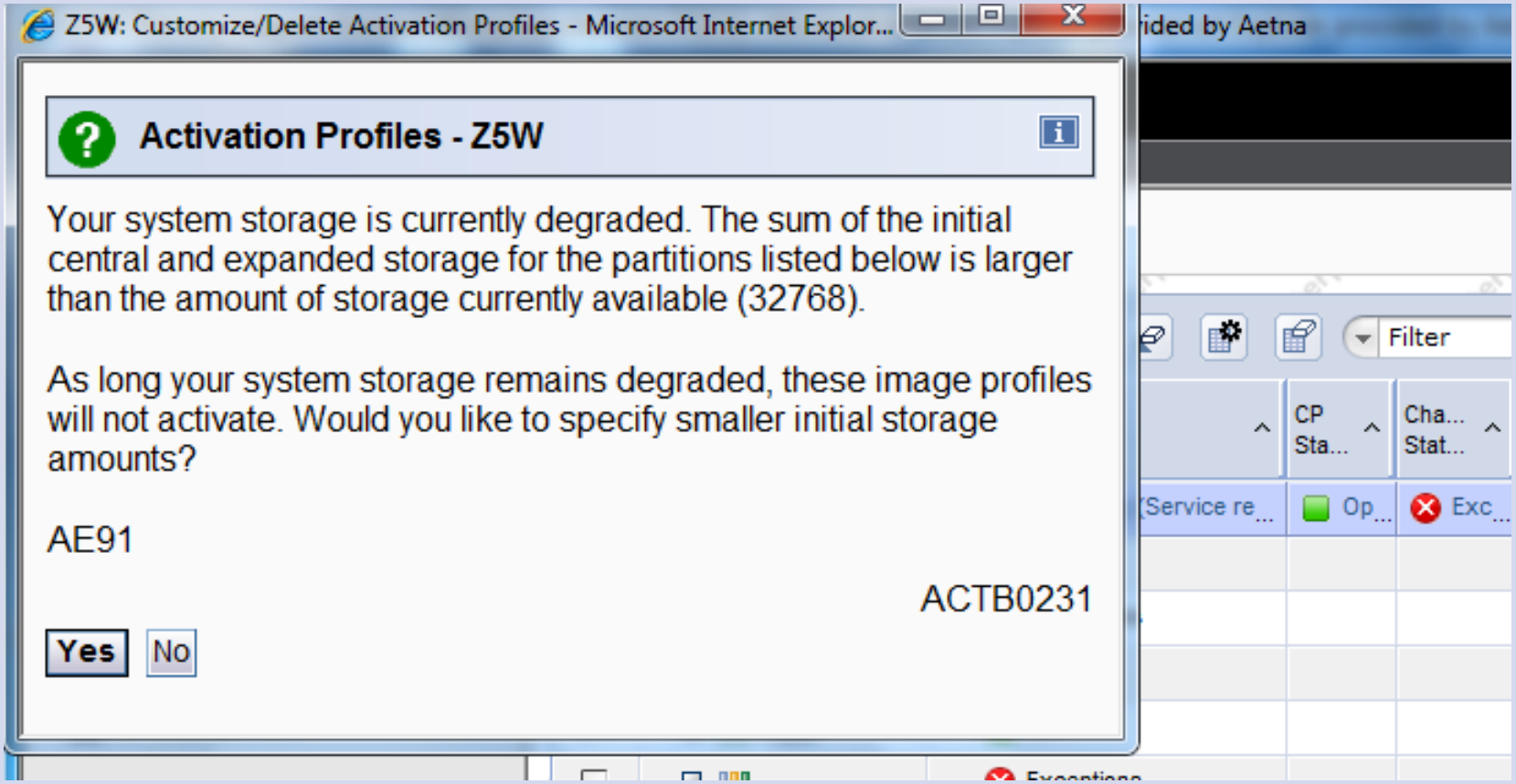


# Fenced Book - USER View



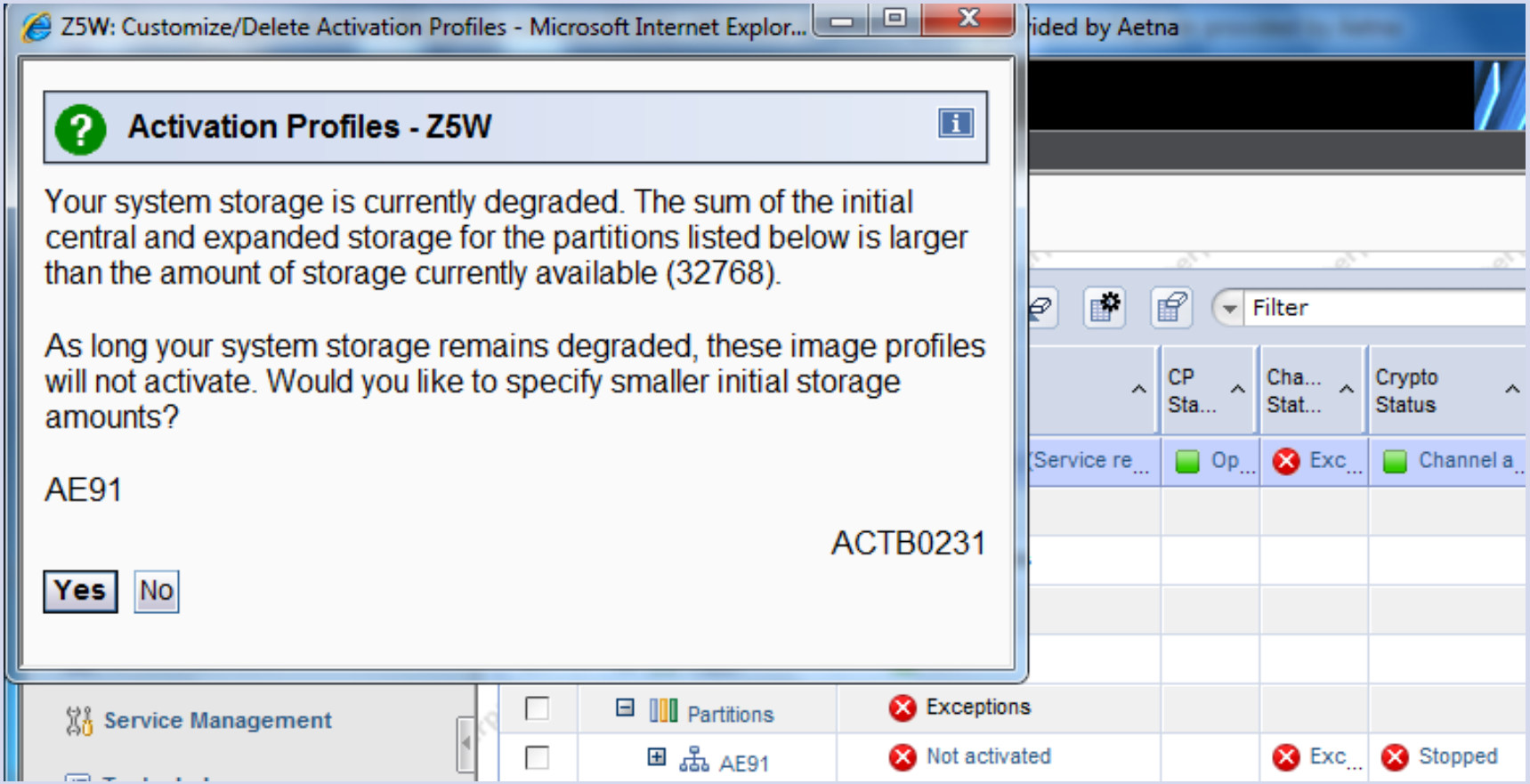
# The Evolution of Managing Real Storage and zFLASH's Impact

## Fenced Book - Activation Msg

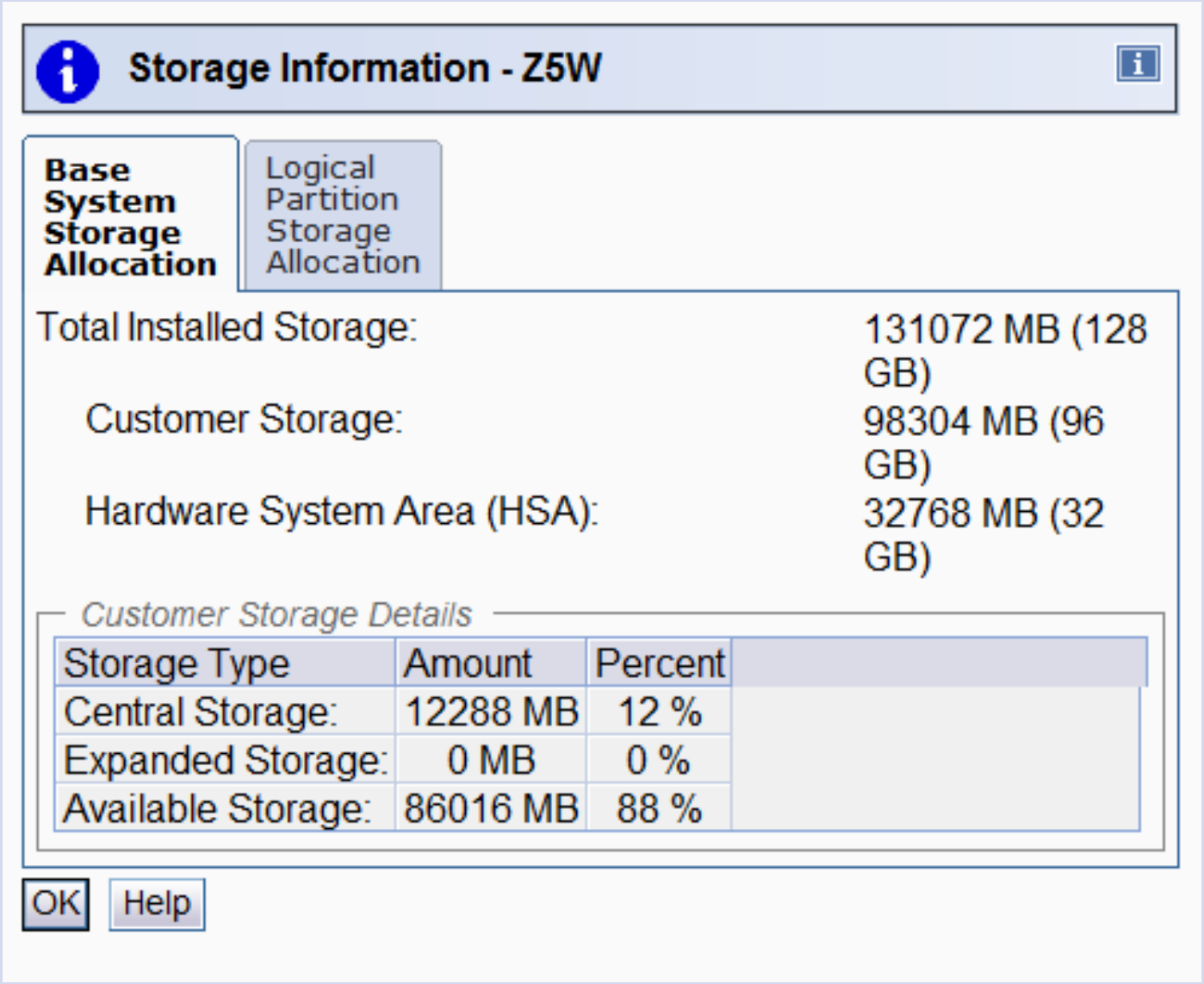


# The Evolution of Managing Real Storage and zFLASH's Impact

## Fenced Book - Activation Issue

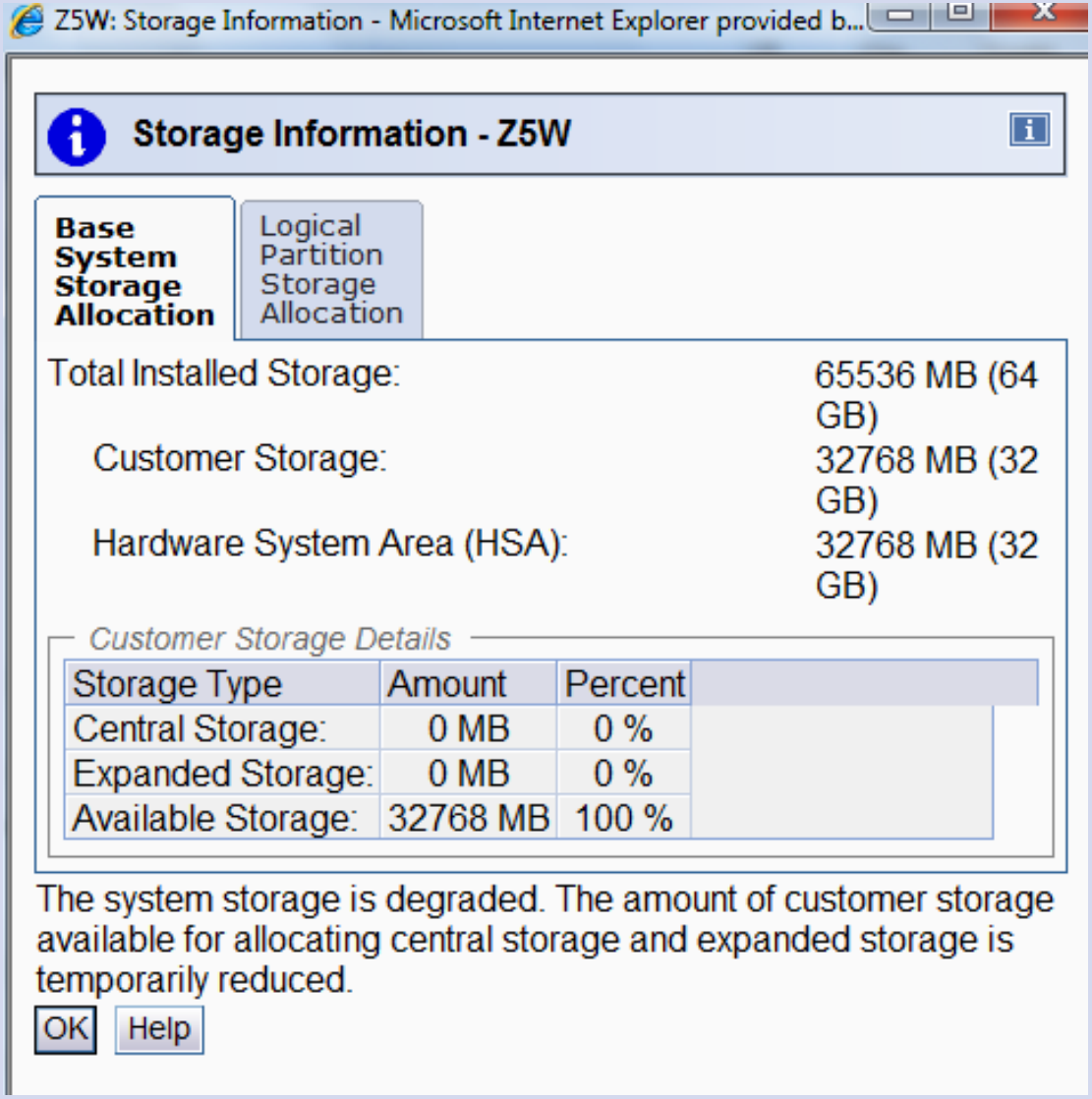


## Fenced Book - Storage originally



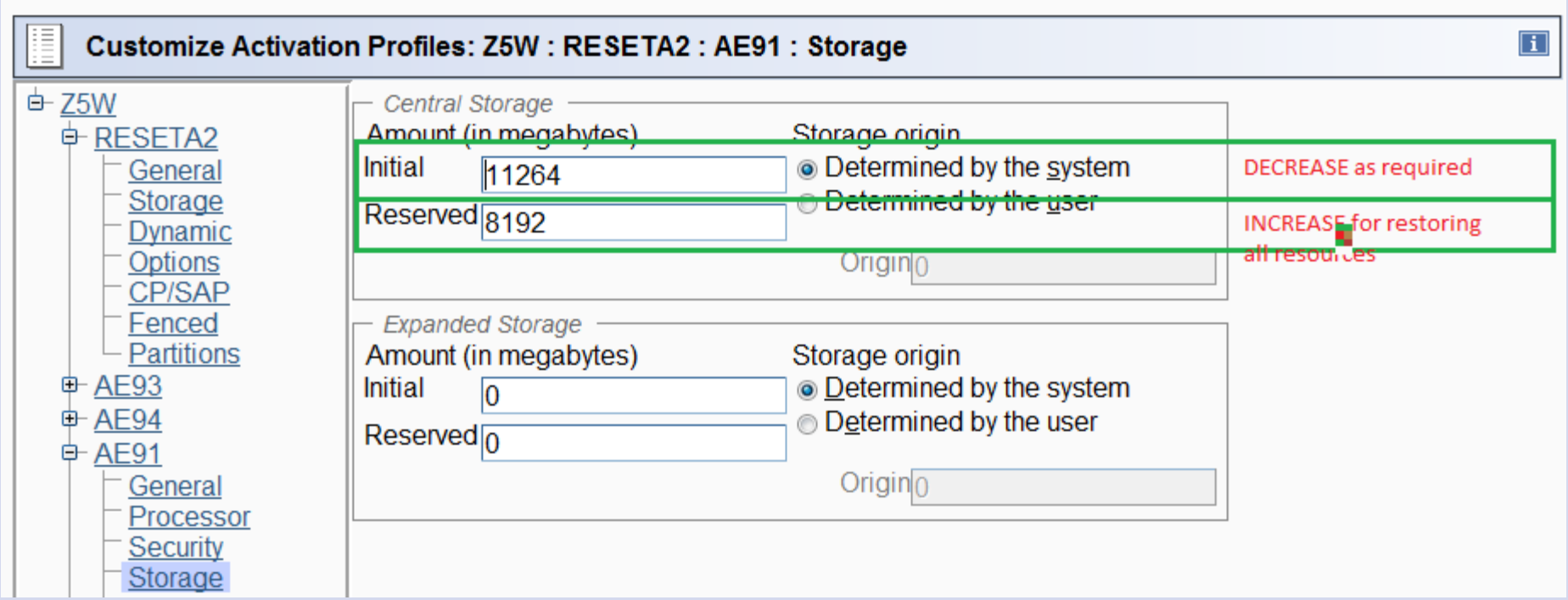
# The Evolution of Managing Real Storage and zFLASH's Impact

## Fenced Book - Storage degraded



# The Evolution of Managing Real Storage and zFLASH's Impact

## Fenced Book -



## Fenced Book - Actions

- POR and come up with a fenced book
- Make Processor adjustments if needed
- Determine LPAR priority list
- Re-adjust available storage for activation of LPARs (don't forget the RESERVE memory for non-disruptive restoration of temporarily reduced storage allocation)
- zFLASH may make the difference between coming up with a subset of your LPARs and having the full list of LPARs running that tolerate the memory reduction due to zFLASH Paging

# Thanks

Session 16180