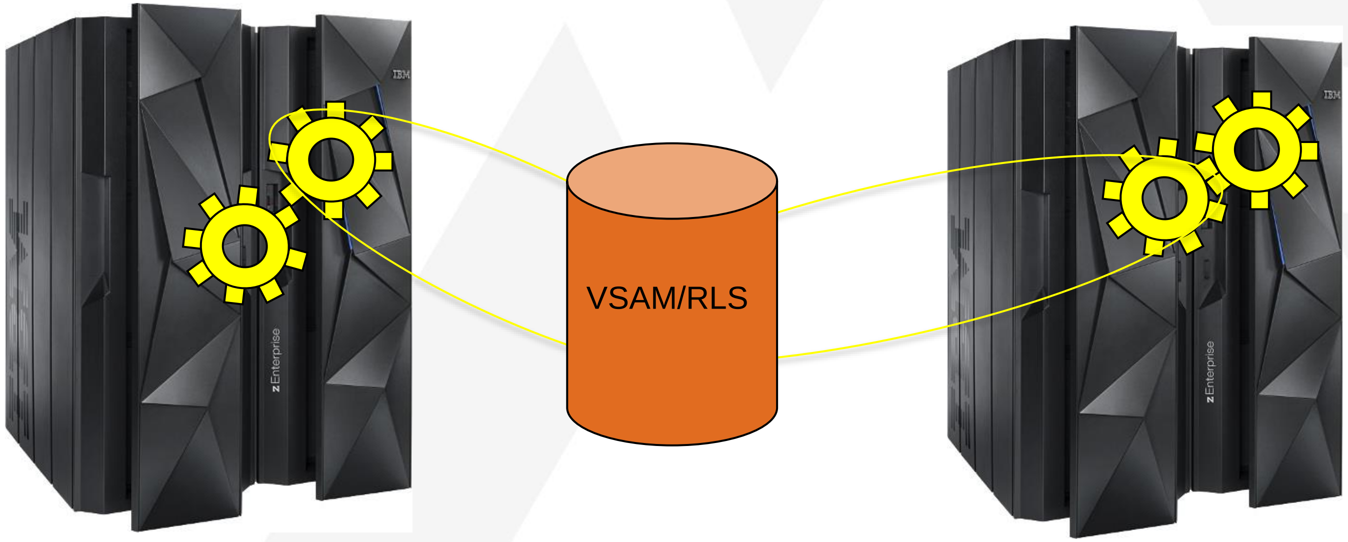
Sysplex Data Sharing

What You Need to Know for Implementing RLS and TVS



Terri Menendez
STSM
zSystems, IBM Corp
terriam@us.ibm.com

August 2024

Agenda

- RLS Enhancements
- RLS Introduction
- RLS SYSPLEX Example
- RLS Buffering
- RLS Caching
- RLS Physical I/O
- RLS Locking
- RLS Data Sets
- Monitoring
- SMF Reporting
- RLS SYSPLEX Example – Multiple Exploiters
- RLS Performance Enhancements
- Customer Benchmarks / Performance Data

- TVS Introduction
- RLS/TVS SYSPLEX Example
- TVS Parameters
- TVS Commands
- TVS Commits/Backouts
- TVS Auto Commits
- IDCAMS Commands
- Forward Recovery
- Peer Recovery
- References

RLS Enhancements

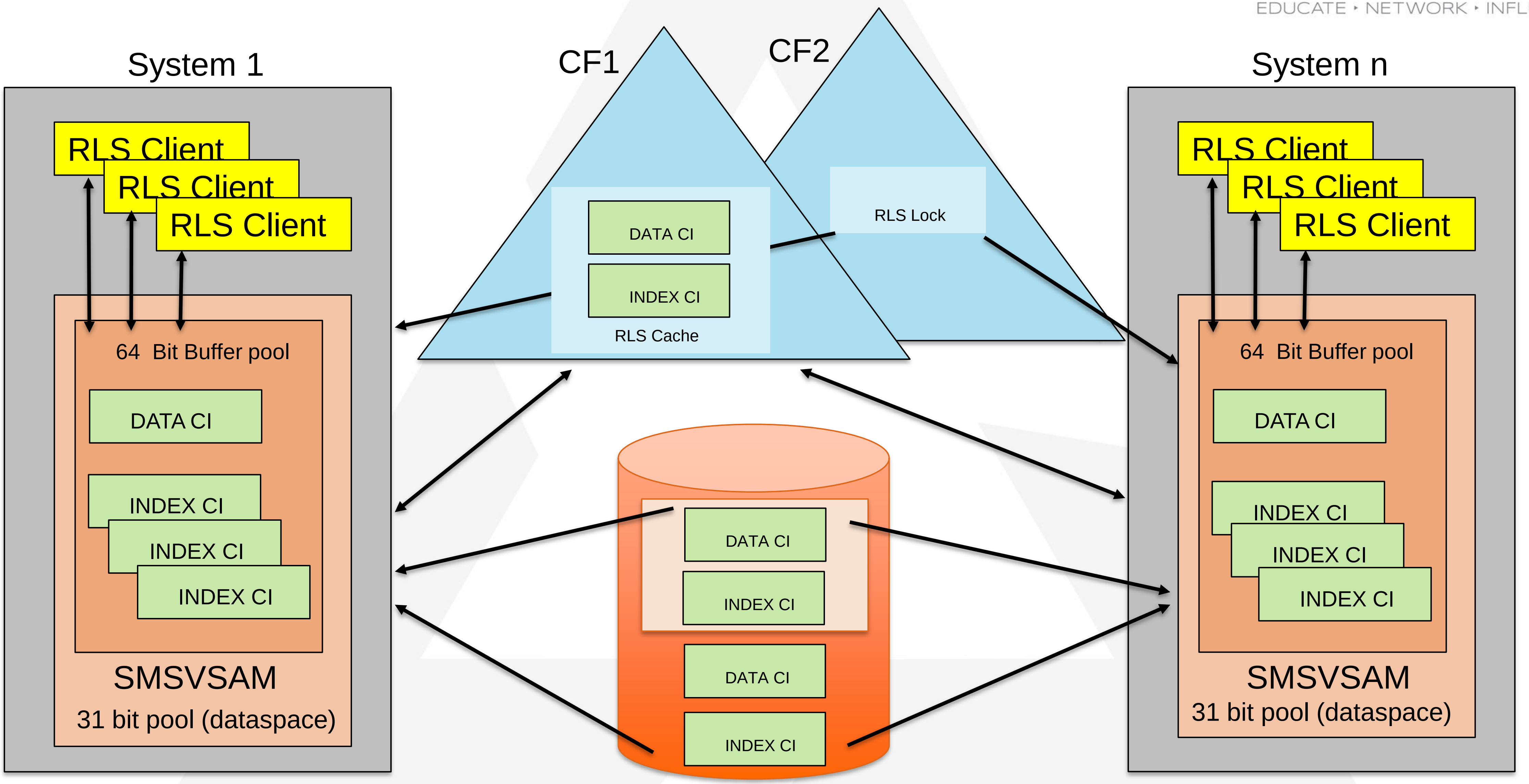
- CA Reclaim (remove needs to reorg KSDSs)
- RLS for Catalog (removes SYSIGGV2 enq)
- RLS for AMS
- Omegamon RLS Support
- SHCDS LISTSTAT (sysplex wide stats while DS open)
- Access Method Encryption
- VSAM / VSAM RLS zHyperlink™ Exploitation
- KSDS RLS Index Record Locking (removes CI split lock)
- Primary / Secondary Extent Reduction

- VERIFY RECOVER
- TVS Auto-Commits
- RLS/TVS CDC/IIDR
- RLS Upgrade Locking   (removes AIX upgrade lock)
- EzNoSLQ NoSQL Key:Value database
  - Native APIs
  - C APIs
  - Java APIs (OA64018)
  - Python APIs (OA66418)

VSAM RLS Introduction

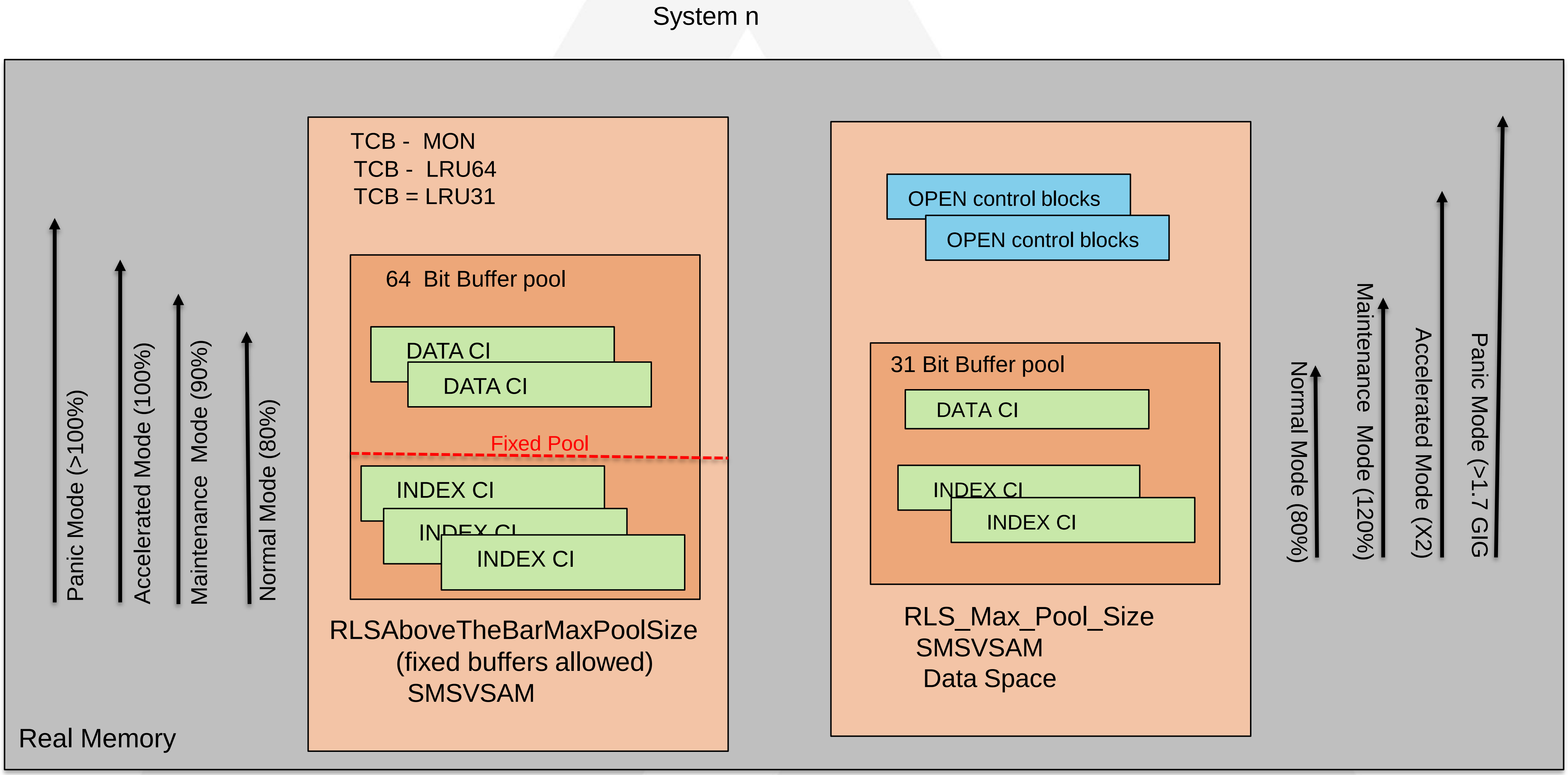
- VSAM RLS provides full data sharing to your existing VSAM files (KSDS, ESDS,(V)RRDS), in a parallel SYSPLEX.
- Allows for high availability (HA) by allowing data sets to be shared cross LPARs/CECs via structures in the Coupling Facility (CF).
- Allows for high scalability by allowing many address spaces and LPARs to share the same files.
- Provides record level serialization, 64 bit buffering, and global caching for better performance.

RLS SYSPLEX Example



RLS Buffering

RLS Buffer Pools



RLS Buffering - LRU

RLS 64 bit LRU - Modes

The LRU for the 64 bit buffer pool operates in four modes:

- **Normal Mode** - Total 64 bit pool size is less than 80% of RLSAboveTheBarMaxPoolSize.
- **Maintenance Mode** - Total 64 bit pool size is greater than 80% and less than 90% of RLSAboveTheBarMaxPoolSize.
- **Accelerated Mode** - Total 64 bit pool size is greater than 90% and less than 100% of RLSAboveTheBarMaxPoolSize.
- **Panic Mode** - Total 64 bit pool size is greater than 100% of RLSAboveTheBarMaxPoolSize

RLS Buffering – LRU (cont.)

The LRU will release buffers as follows:

- **Normal Mode** - Buffers stay indefinitely in normal mode.
- **Maintenance Mode** - Buffers 60 minutes or older will be released.
- **Accelerated Mode** - Buffers 30 minutes are older will be released. Requests for new buffers will first be stolen. If there are no buffers to steal a new get block will be done.
- **Panic Mode** - Buffers 5 minutes are older will be released. Requests for new buffers will first be stolen. If there are no buffers to steal, the request will sleep until LRU runs.

RLS Buffering Parameters

SYS1.PARMLIB(IGDSMSxx):

- ○ **RLS_MAX_POOL_SIZE**(100/nnnn) Where nnnn = (10 to 9999), anything over 1500 is treated as a maximum of 1728M.
- ○ **RLSAboveTheBarMaxPoolSize**(sysname/ALL,0/nnnn) Where nnnn is either 0 (default), or 500M to 2,000,000M. Minimum value 500M.
- ○ **RLSFixedPoolSize**(sysname/ALL,0/nnnn) Where nnnn is either 0 to 80% of real storage

- ❑ Only one RLS_Max_Pool_Size for all lpars
- ❑ RLSAboveTheBarMaxPoolSize and RLSFixedPoolSize can have individual sizes for each LPAR.
  - ➢ Caution: Buffers are fixed on a first come first serve basis. For example, if the first data set opened and accessed has a 4K CISIZE, then the fixed buffers will be 4K in size for the life of this SMSVSAM instance.
  - ➢ Fixed buffers can provide a significant performance improvement since the RSM pin/unpin calls are avoided.

RLS Buffering Parameters (Cont.)

❑ Pool size values are a goal for which the RLS Least Recently Used (LRU) manager tries to maintain. If more buffers are required at any given time, the pool may temporarily exceed the values set.

❑ Total amount of buffer pools should not exceed amount of real storage. A paged out buffer is freed by the LRU.

❑ Data sets must set RLSAboveTheBar(YES) in the DATACLAS to use the above the bar pool. NO is the default.

Sizing the Pools:

❑ For CICS or other LSR users, start with a pool size = sum of LSR pools per lpar.

❑ For catalog, start with VLF MAXVIRT * 4096 per lpar or based on total catalog size.

❑ For exploiters with no existing buffering (ie HSM), consider starting with buffering the index CIs, and 20% of the data CIs (refer to High Used RBAs in the catalog).

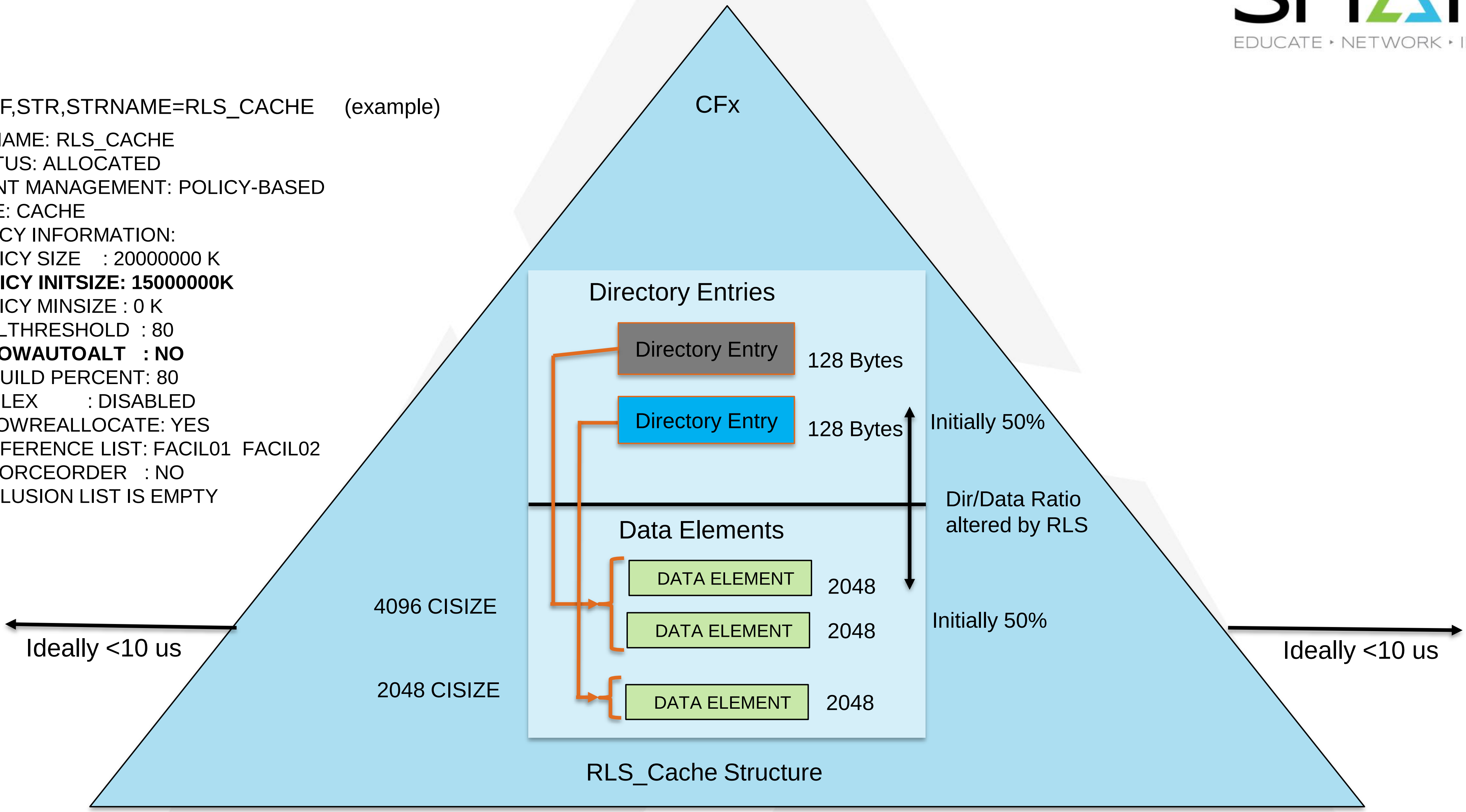
❑ Buffering success is measured by BMF hit rate, average elapse and CPU time, and LRU mode.

❑I nternal IBM testing with 128 GIG above the bar pool.

RLS CF Caching

RLS Cache Structure

```
D XCF,STR,STRNAME=RLS_CACHE     (example)
STRNAME: RLS_CACHE
 STATUS: ALLOCATED
 EVENT MANAGEMENT: POLICY-BASED
 TYPE: CACHE
 POLICY INFORMATION:
  POLICY SIZE     : 20000000 K
  POLICY INITSIZE: 15000000K
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: 80
  DUPLEX         : DISABLED
  ALLOWREALLOCATE: YES
  PREFERENCE LIST: FACIL01  FACIL02
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY
```



CFx

Directory Entries

Directory Entry 128 Bytes

Directory Entry 128 Bytes

Initially 50%

Dir/Data Ratio altered by RLS

Data Elements

DATA ELEMENT 2048

DATA ELEMENT 2048

DATA ELEMENT 2048

Initially 50%

4096 CISIZE

2048 CISIZE

Ideally <10 us

Ideally <10 us

RLS_Cache Structure

RLS Caching

SYS1.PARMLIB(IGDSMSxx):

- **RLS_MaxCFFeatureLevel**(Z/A) Where feature level A allows CISIZE >4K to be cached.

- RLS uses a store-thru-cache design, serialized by the CF CASTOUT lock.
- Each SMSVSAM registers interest with directory entries when reading a CI, then invalidates the entries on writing the (XI).
- Entries are reclaimed when a cache is “full” and will invalidate the associated buffers.
- Data elements can be scaled back via DATACLAS RLSCFCACHE(ALL/UPDATESONLY/NONE/DIRONLY)
- RLS caches cannot be duplexed.
- RLS caches will automatically rebuild for CF issues. A new structure will be allocated empty.

CISIZE	Dir Entry	Data Elements
2K	1	1
4K	1	2
6K	1	3
8K	1	4
10K	1	5
12K	1	6
14K	1	7
16K	1	8
18K	1	9
20K	1	10
22K	1	11
24K	1	12
26K	1	13
28K	1	14
30K	1	15
32K	1	16

RLS Cache Sizing

- **The sum of all the RLS cache structures should equal the sum of all the RLS buffer pools across the sysplex.**

- RLS BMF False Invalids and CF Reclaims for XI indicate the cache is too small relative to the size of the buffer pools.

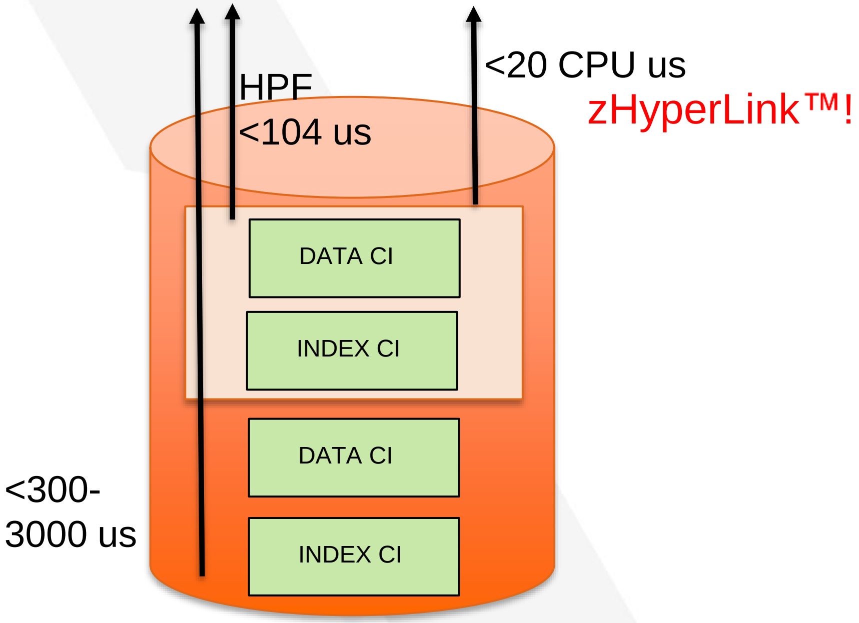
- Splitting data sets with same performance requirements across multiple cache structures may lead to cache balancing problems.

- Different applications should have their own cache structures.

RLS Physical I/O Considerations

RLS Physical I/O

- ❑ RLS always writes the Records (CIs) to disk.
- ❑ The CF castout lock serializes the write to the CF cache and to disk from each system. Readers wait for the castout lock.
- ❑ RLS supports compression, striping, and data set encryption.
- ❑ RLS supports the new zHyperlink technology for reads only.
- ❑ CI contention is measure by “REDOs” and CASTOUT lock contention/retries.

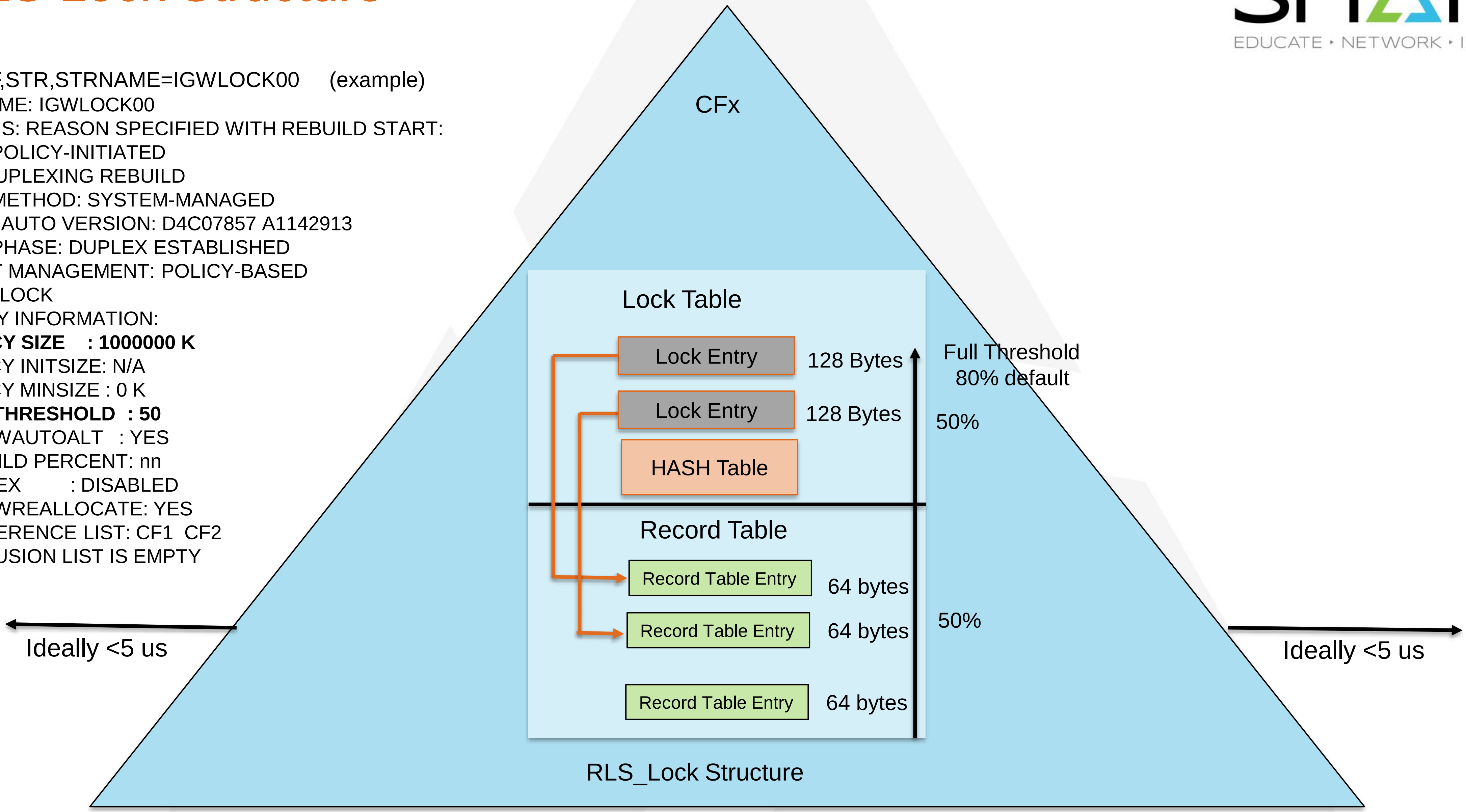


Example response times

RLS Locking Considerations

RLS Lock Structure

```
 D XCF,STR,STRNAME=IGWLOCK00     (example)
STRNAME: IGWLOCK00
 STATUS: REASON SPECIFIED WITH REBUILD START:
           POLICY-INITIATED
         DUPLEXING REBUILD
           METHOD: SYSTEM-MANAGED
             AUTO VERSION: D4C07857 A1142913
           PHASE: DUPLEX ESTABLISHED
 EVENT MANAGEMENT: POLICY-BASED
 TYPE: LOCK
 POLICY INFORMATION:
  POLICY SIZE    : 1000000 K
  POLICY INITSIZE: N/A
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 50
  ALLOWAUTOALT   : YES
  REBUILD PERCENT: nn
  DUPLEX         : DISABLED
  ALLOWREALLOCATE: YES
  PREFERENCE LIST: CF1  CF2
  EXCLUSION LIST IS EMPTY
```



RLS Lock Structure Sizing

- Lock_Structure_Size = 10M * number_of_Systems_in_sysplex * Lock_entry_Size
  - Lock_entry_Size (depends on the CFRM MAXSYSTEM value):
    - MAXSYSTEM <= 7 Lock_entry_size = 2
    - MAXSYSTEM >= 8 & <24 Lock_entry_size = 4
    - MAXSYSTEM >=24 & <=32 Lock_entry_size = 8

- Example: MAXSYSTEM = 23 and 8 systems in sysplex
    - IGWLOCK00 = 10M * 8 * 4 = 320M

RLS Locking

- ❑ RLS has many types of locks:
  - ○ Record level
  - ○ Index level
  - ○ Data set level
  - ○ LPAR level
  - ○ Sysplex level

- ❑ Two types of lock management:
  - ○ Global – lock requests from different lpars.
  - ○ Local - lock requests from same system.

- ❑ Two types of lock contention:
  - ○ True - two or more requests want the same lock. Usually, can be managed by the user.
  - ○ False - two or more requests want different locks but hash to the same lock entry. Indication of a too small lock structure.

- ❑ Assigned data sets to different lock structures for different applications.

- ❑ All locks have SMF42 fields providing statistics on contention.

RLS Lock Rate Display

- ***D SMS,CFLS (Coupling Facility Lock Structure)***

```
J80      12152 15:30:56.26   D SMS,CFLS
IGW320I 15:30:56 Display SMS,CFLS(IGWLOCK00 )
PRIMARY STRUCTURE:IGWLOCK00 VERSION:C99DC09480021972 SIZE:200704K
RECORD TABLE ENTRIES:434612 USED:984
SECONDARY STRUCTURE:IGWLOCK00 VERSION:C9A3EFBCF6FC3610
SIZE:200704K
RECORD TABLE ENTRIES:434612 USED:984
LOCK STRUCTURE MODE: DUPLEXED  STATUS: ENABLE
System   Interval   LockRate   ContRate  FContRate   WaitQLen
J80      1 Minute     1239.1      0.065      0.042       0.33
J80       1 Hour       373.9      0.099      0.038       0.04
J80       8 Hour  ---------- ---------- ---------- ----------- --------
J80        1 Day  ---------- ---------- ---------- ------------ -------
  (13)   1 Minute      344.9      0.029      0.014       0.10
  (13)    1 Hour       108.4      0.041      0.015       0.03
```

Ideal contention levels are less 0.5%

RLS Data Set Considerations

Data Set Considerations

- All VSAM Record Type Data sets:
  - Maximum size with DATACLAS Extended Format (EF) and Extended Addressibility (EA). Based on CI size, for example:
    - A CI size of 4 KB yields a maximum data set size of 16 TB
    - A CI size of 32 KB yields a maximum data set size of 128 TB
  - Maximum number of volumes per data set: 59
  - Maximum number of extents per volume 123.
  - Maximum number of extents with DATACLAS option Space Constraint Relief . . . Y :
    - 59 * 123 = 7257
      - Reduce space option in DATACLAS Reduce Space Up To percentage to for extends
      - CISIZEs which span tracks: 14K, 20K, 22K, 28K, 30K, AND 32K
  - May want to key range very large data sets to avoid performance and recovery issues.

Data Set Considerations

- KSDSs
  - **Most scalable type of VSAM** data set when using random inserts.
  - **Avoid adding to the end of a KSDS (ie ascending keys) when possible.**
  - Avoid a high volume of inserts into an empty data set. Add/delete dummy keys to prime free space.
  - Consider larger CISIZEs for data sets with high inserts / erases to avoid CI/CA splits/reclaims.
  - Consider smaller CISIZEs for high updates (no record length change) to avoid CI REDO’s and CASTOUT lock contention.
  - Consider compressing records (DATACLAS COMPRESSION=Y) especially for large RECORDSIZEs. May also reduce CI splits/reclaims by allowing for more records per CI.
  - Use CA Reclaim to improve performance, reduce space, and avoid reorganizations for fragmentation (applies to data sets with ERASEs).
  - Reduce the need to extend data sets by providing adequate primary space.
  - CI split/reclaims can occur concurrently in different CAs. Serialized by Component_1 Class 4 (Index) locks.
  - CA split/reclaim are serial for the data set. Controlled by the Component_1 Class1 (DIWA) lock.
  - Reduce index levels when possible, through the use of CA Reclaim, more records per CI.

Data Set Considerations

- ❑ ESDSs
  - ○ Not a scalable data set for high inserts, since inserts must be at the end of the data set.
  - ○ Consider sub-dividing poor performing ESDSs.
  - ○ Inserts are serial and are controlled by the Component_2 lock.

- ❑ Alternate Indexes
  - ○ Prior to z/OS 2.3 inserts, erases, and updates with record length changes were serial and controlled by the Component_2 (Upgrade) lock.
  - ○ With z/OS 2.2 lower level serialization (record locks and REDOs) are used to allow concurrent writes.

RLS Monitoring

RLS Monitors

- ❑ RMF Monitor III
- ❑ IBM Tivoli Omegamon XE for Storage (see IBM website for more info)
- ❑ IBM Tivoli Omegamon XE for CICS (see IBM website for more info)

RMF RLS Displays

```
                RMF III Sysplex Report Selection Menu
Selection ===>

Enter selection number or command for desired report.

 Sysplex Reports
    1 SYSSUM   Sysplex performance summary          (SUM)
    2 SYSRTD   Response time distribution           (RTD)
    3 SYSWKM   Work Manager delays                  (WKM)
    4 SYSENQ   Sysplex-wide Enqueue delays          (ES)
       .
    7 CFACT    Coupling Facility activity           (CA)
       .
   10 RLSSC    VSAM RLS activity by storage class   (RLS)
   11 RLSDS    VSAM RLS activity by data set        (RLD)
   12 RLSLRU   VSAM LRU overview                    (RLL)
```

RMF CF Structure Detail Display

SHARE
EDUCATE ‣ NETWORK ‣ INFLUENCE

RMF CF Structure Detail:

```
-------------------------------------------------------------------------------------------------------------
                              COUPLING  FACILITY  USAGE  SUMMARY
-------------------------------------------------------------------------------------------------------------
                                       STRUCTURE SUMMARY
-------------------------------------------------------------------------------------------------------------

                                                    % OF          % OF   % OF    AVG    LST/DIR    DATA        LOCK     DIR REC/
        STRUCTURE                           ALLOC   CF       #    ALL    CF      REQ/   ENTRIES    ELEMENTS   ENTRIES   DIR REC
 TYPE   NAME            STATUS CHG          SIZE    STOR     REQ  REQ    UTIL    SEC    TOT/CUR    TOT/CUR              TOT/CUR  XI'S

CACHE  RLSCACH1  ACTIVE            7G     19.8     299236 19.8    n/a     332.48  2049K      3506K        n/a        40K

                                                                                   2049K      3171K                     2K

CACHE  RLSCACH2  ACTIVE            7G     19.8     145452  8.2    n/a     162.32  2049K      3506K        n/a        23K
                                                                                   2049K      3171K                     0K
```

Note: RMF CFDETAIL must be active: F RMF,MODIFY III,CFDETAIL

RLS RMF Displays

F RMF MONITOR III SYSPLEX **RLS Activity by Storage Class** (sysplex wide)

```
ERB3BUF          RMF V1R8   VSAM RLS Activity  - SYSPLEX       Line 1 of 14
Command ===>                                        Scroll ===> CSR

LRU Status     :  Good
Contention % :      0.0
False Cont % :      0.0

StorClas     Access      Resp    -------------Read----------------- ------ BMF -----------  Write
                         Time    Rate   BMF% CF%  DASD% Valid% False Inv%  rate
STORCLAS1
Below 2G      DIR       0.000      0.00      0.0    0.0    0.00       0.0        0.00       0.00
              SEQ       0.000      0.00      0.0    0.0    0.00       0.0        0.00       0.00
Above 2G      DIR       0.000    84299   100.0    0.0    0.00       100        0.00       0.00
              SEQ       0.000      0.00      0.0    0.0    0.00        0.0        0.00       0.00
```

RLS RMF Displays (cont.)

F RMF MONITOR III SYSPLEX **RLS Activity by Data Set** (sysplex wide)

```
ERB3BUF          RMF V1R8   VSAM RLS Activity  - SYSPLEX        Line 1 of 14
Command ===>                                          Scroll ===> CSR

LRU Status      :  Good
Contention % :      0.0
False Cont % :      0.0

Sphere/DS  Access Resp    -------------Read---------------- ------ BMF -----------  Write
                  Time    Rate   BMF% CF%  DASD% Valid% False Inv%  rate
USERCAT1
Below 2G     DIR      0.000  2438    100    0.0     0.0       100       0.00      0.00
             SEQ      0.000  0.00    0.0    0.0     0.0       0.0       0.00      0.00
Above 2G     DIR      0.000  0.00    00.0   0.0     00.0      0.0       0.00      0.00
             SEQ      0.000  0.00    0.0    0.0     0.0       0.0       0.00      0.00
USERCAT1.INDEX
Below 2G     DIR      0.00   7314    100    0.0     0.0       100       0.00      0.00
             SEQ      0.000  0.00    0.0    0.0     0.0       0.0       0.00      0.00
Above 2G     DIR      0.000  0.00    00.0   0.0     00.0      0.0       0.00      0.00
             SEQ      0.000  0.00    0.0    0.0     0.0       0.0       0.00      0.00
```

RLS RMF Displays (cont.)

RMF MONITOR III **SYSPLEX VSAM LRU Overview**

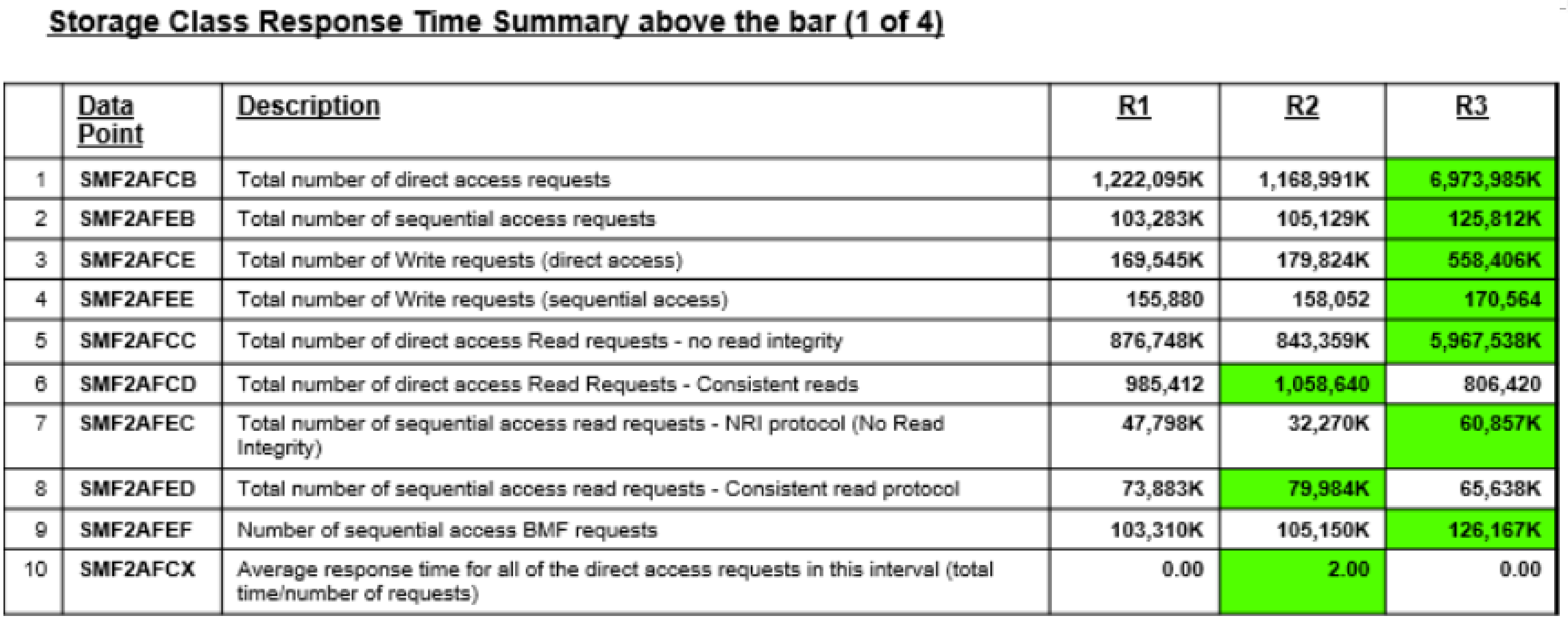
```
Samples: 59     Systems: 1    Date: 07/27/18  Time: 12.38.50  Range: 10    Se

MVS         Avg CPU  - Buffer Size -  Accel  Reclaim  ------ Read -----
System       Time        Goal   High     %        %          BMF%   CF%  DASD%

SYS1
 Below 2GB   0.1147   850M   952M   100      0.0        85.0  10.0       5.0
 Above 2GB   0.112       20G    17G    0.0     0.0        100      0.0      0.0

SYS2
 Below 2GB   0.1147   850M   610M  43.0      0.0        89.0     9.0       2.0
 Above 2GB   0.112       20G    16G    0.0     0.0        100      0.0      0.0
```

RLS SMF Reporting

RLS SMF Reporting

- SMF 42 Subtypes 15, 16, 17, 18, 19
  - **Subytpe 15** - RLS statistics by Storage Class
  - **Subtype 16** - RLS statistics by Data set
    - Must use V SMS,MONDS(spherename),ON to collect subtype 16 statistics.
    - Must turn on data set collection when using RMF III:
      - F III,VSAMRLS(ADD(DSNAME.**))
  - **Subtype 17** - RLS locking statistics
  - **Subtype 18** - RLS caching statistics
  - **Subtype 19** - BMF statistics
- Note: Only one system in the sysplex collects the SMF 42 records. The system collecting the records is displayed in the D SMS,SMSVSAM operator command.

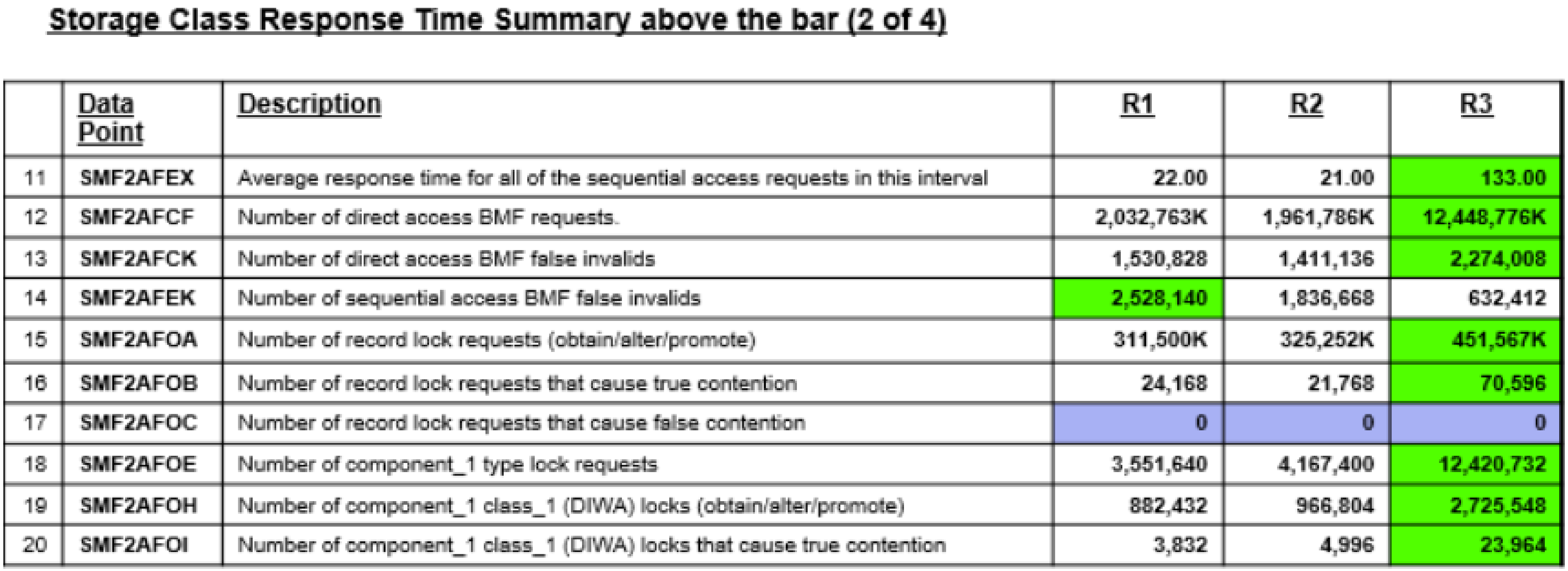
RLS SMF Reporting



Storage Class Response Time Summary above the bar (1 of 4)

	Data Point	Description	R1	R2	R3
1	SMF2AFCB	Total number of direct access requests	1,222,095K	1,168,991K	6,973,985K
2	SMF2AFEB	Total number of sequential access requests	103,283K	105,129K	125,812K
3	SMF2AFCE	Total number of Write requests (direct access)	169,545K	179,824K	558,406K
4	SMF2AFEE	Total number of Write requests (sequential access)	155,880	158,052	170,564
5	SMF2AFCC	Total number of direct access Read requests - no read integrity	876,748K	843,359K	5,967,538K
6	SMF2AFCD	Total number of direct access Read Requests - Consistent reads	985,412	1,058,640	806,420
7	SMF2AFEC	Total number of sequential access read requests - NRI protocol (No Read Integrity)	47,798K	32,270K	60,857K
8	SMF2AFED	Total number of sequential access read requests - Consistent read protocol	73,883K	79,984K	65,638K
9	SMF2AFEF	Number of sequential access BMF requests	103,310K	105,150K	126,167K
10	SMF2AFCX	Average response time for all of the direct access requests in this interval (total time/number of requests)	0.00	2.00	0.00

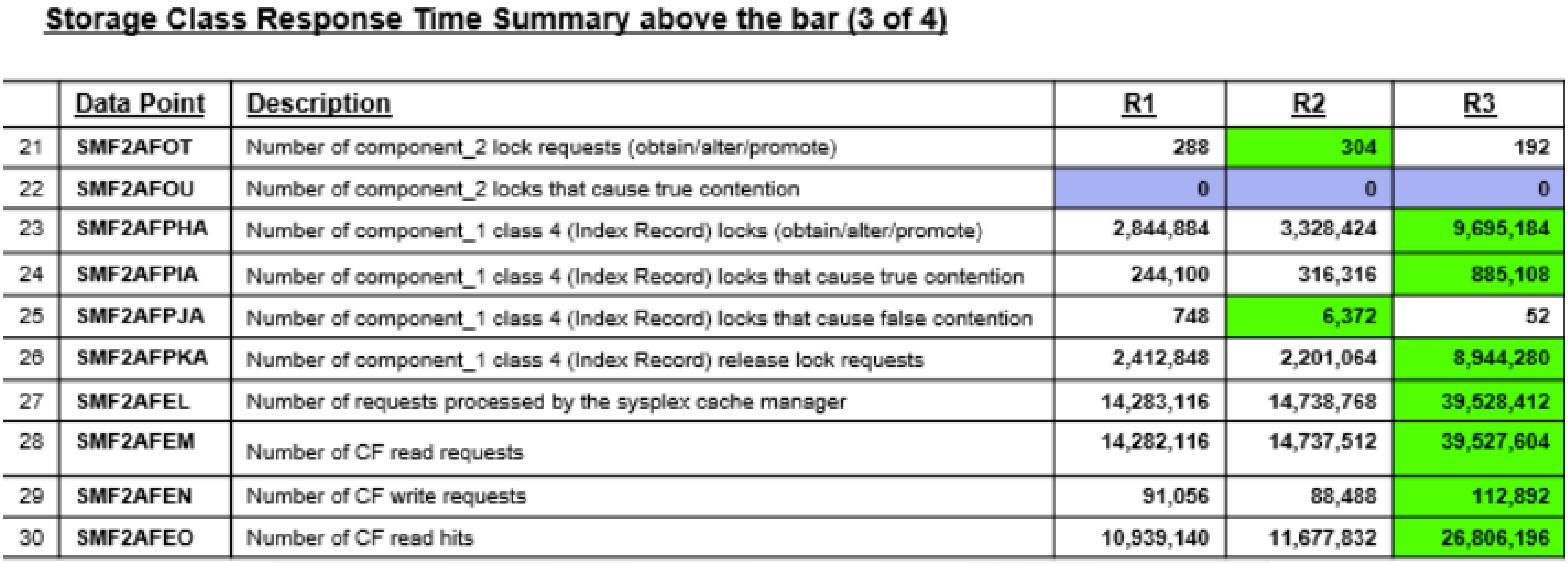
RLS SMF Reporting



Storage Class Response Time Summary above the bar (2 of 4)

	Data Point	Description	R1	R2	R3
11	SMF2AFEX	Average response time for all of the sequential access requests in this interval	22.00	21.00	133.00
12	SMF2AFCF	Number of direct access BMF requests.	2,032,763K	1,961,786K	12,448,776K
13	SMF2AFCK	Number of direct access BMF false invalids	1,530,828	1,411,136	2,274,008
14	SMF2AFEK	Number of sequential access BMF false invalids	2,528,140	1,836,668	632,412
15	SMF2AFOA	Number of record lock requests (obtain/alter/promote)	311,500K	325,252K	451,567K
16	SMF2AFOB	Number of record lock requests that cause true contention	24,168	21,768	70,596
17	SMF2AFOC	Number of record lock requests that cause false contention	0	0	0
18	SMF2AFOE	Number of component_1 type lock requests	3,551,640	4,167,400	12,420,732
19	SMF2AFOH	Number of component_1 class_1 (DIWA) locks (obtain/alter/promote)	882,432	966,804	2,725,548
20	SMF2AFOI	Number of component_1 class_1 (DIWA) locks that cause true contention	3,832	4,996	23,964

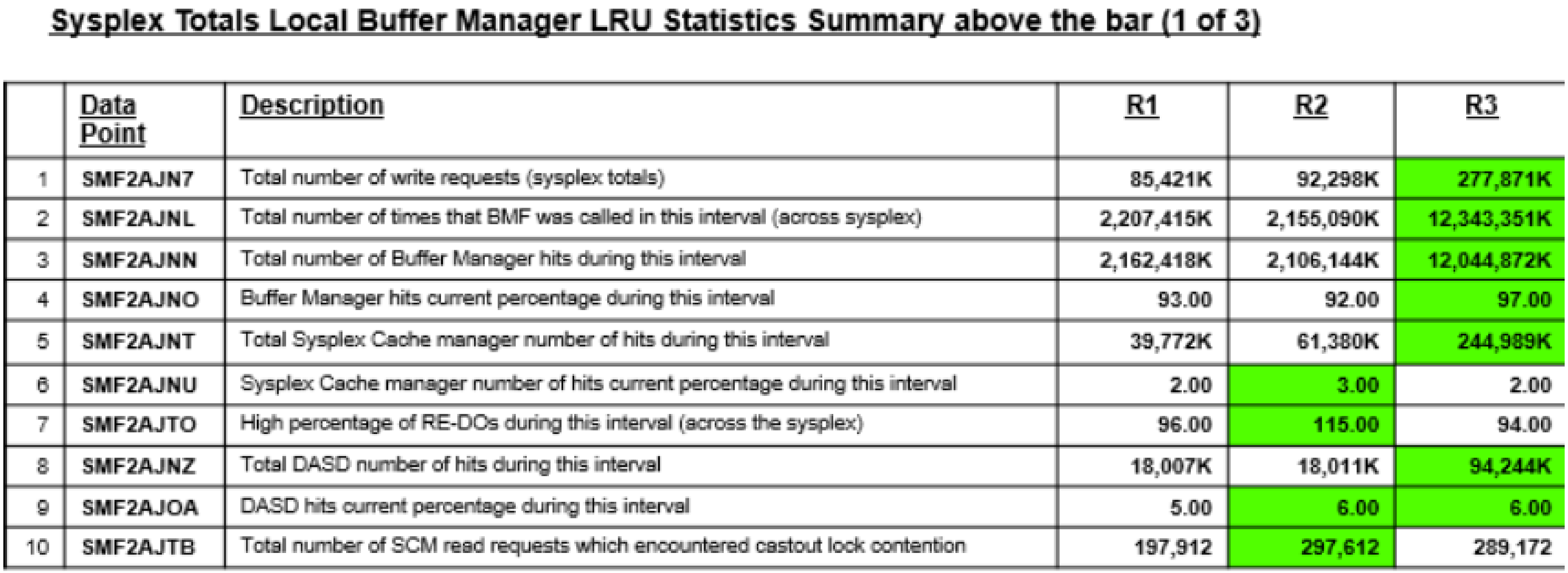
RLS SMF Reporting



Storage Class Response Time Summary above the bar (3 of 4)

	Data Point	Description	R1	R2	R3
21	SMF2AFOT	Number of component_2 lock requests (obtain/alter/promote)	288	304	192
22	SMF2AFOU	Number of component_2 locks that cause true contention	0	0	0
23	SMF2AFPHA	Number of component_1 class 4 (Index Record) locks (obtain/alter/promote)	2,844,884	3,328,424	9,695,184
24	SMF2AFPIA	Number of component_1 class 4 (Index Record) locks that cause true contention	244,100	316,316	885,108
25	SMF2AFPJA	Number of component_1 class 4 (Index Record) locks that cause false contention	748	6,372	52
26	SMF2AFPKA	Number of component_1 class 4 (Index Record) release lock requests	2,412,848	2,201,064	8,944,280
27	SMF2AFEL	Number of requests processed by the sysplex cache manager	14,283,116	14,738,768	39,528,412
28	SMF2AFEM	Number of CF read requests	14,282,116	14,737,512	39,527,604
29	SMF2AFEN	Number of CF write requests	91,056	88,488	112,892
30	SMF2AFEO	Number of CF read hits	10,939,140	11,677,832	26,806,196

RLS SMF Reporting

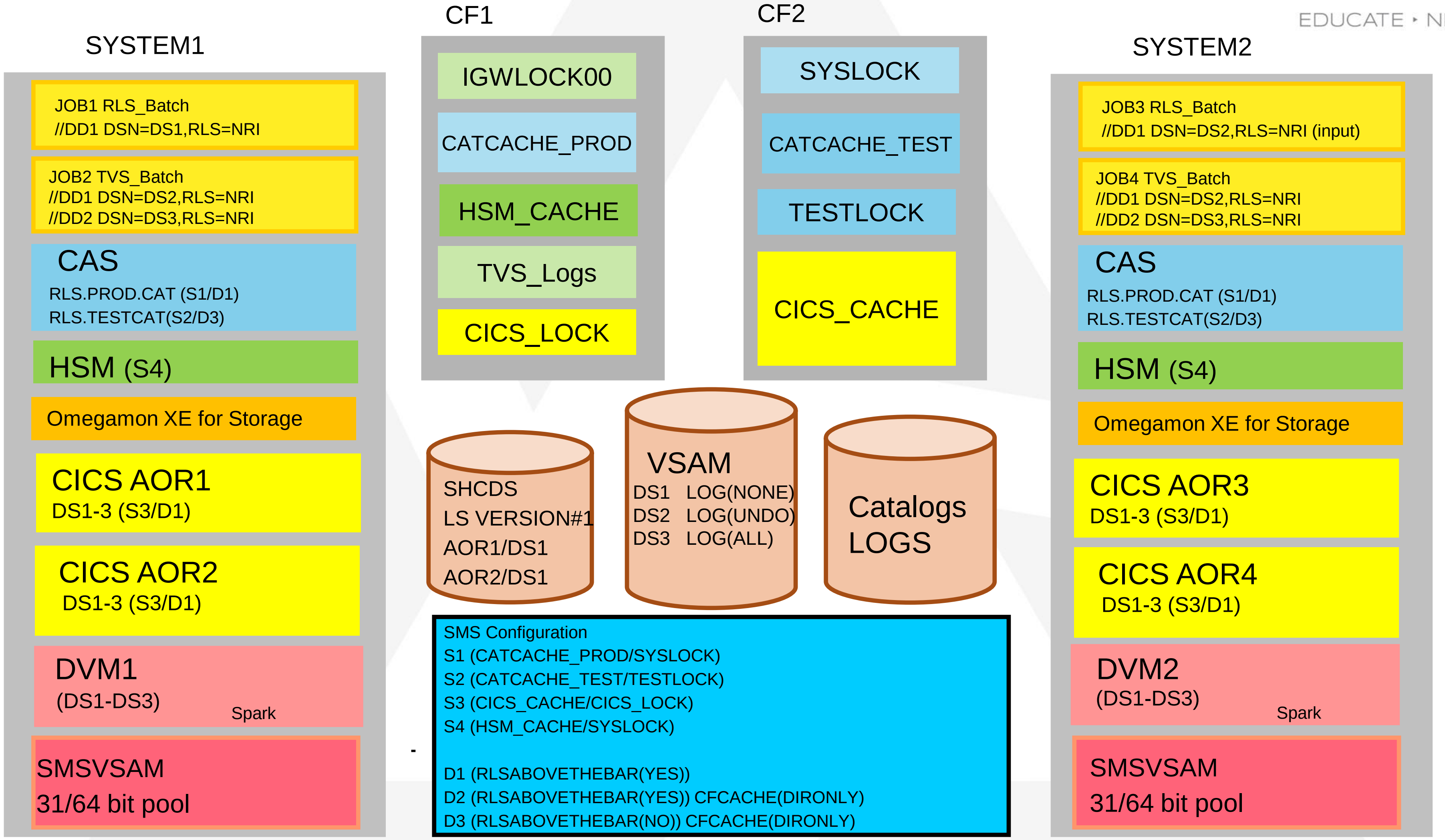


Sysplex Totals Local Buffer Manager LRU Statistics Summary above the bar (1 of 3)

	Data Point	Description	R1	R2	R3
1	SMF2AJN7	Total number of write requests (sysplex totals)	85,421K	92,298K	277,871K
2	SMF2AJNL	Total number of times that BMF was called in this interval (across sysplex)	2,207,415K	2,155,090K	12,343,351K
3	SMF2AJNN	Total number of Buffer Manager hits during this interval	2,162,418K	2,106,144K	12,044,872K
4	SMF2AJNO	Buffer Manager hits current percentage during this interval	93.00	92.00	97.00
5	SMF2AJNT	Total Sysplex Cache manager number of hits during this interval	39,772K	61,380K	244,989K
6	SMF2AJNU	Sysplex Cache manager number of hits current percentage during this interval	2.00	3.00	2.00
7	SMF2AJTO	High percentage of RE-DOs during this interval (across the sysplex)	96.00	115.00	94.00
8	SMF2AJNZ	Total DASD number of hits during this interval	18,007K	18,011K	94,244K
9	SMF2AJOA	DASD hits current percentage during this interval	5.00	6.00	6.00
10	SMF2AJTB	Total number of SCM read requests which encountered castout lock contention	197,912	297,612	289,172

RLS SYSPLEX – Multiple Exploiters

Example RLS Configuration – Multiple Exploiters



SYSTEM1

JOB1 RLS_Batch
//DD1 DSN=DS1,RLS=NRI

JOB2 TVS_Batch
//DD1 DSN=DS2,RLS=NRI
//DD2 DSN=DS3,RLS=NRI

CAS
RLS.PROD.CAT (S1/D1)
RLS.TESTCAT(S2/D3)

HSM (S4)

Omegamon XE for Storage

CICS AOR1
DS1-3 (S3/D1)

CICS AOR2
DS1-3 (S3/D1)

DVM1
(DS1-DS3) Spark

SMSVSAM
31/64 bit pool

CF1

IGWLOCK00

CATCACHE_PROD

HSM_CACHE

TVS_Logs

CICS_LOCK

CF2

SYSLOCK

CATCACHE_TEST

TESTLOCK

CICS_CACHE

SHCDS
LS VERSION#1
AOR1/DS1
AOR2/DS1

VSAM
DS1 LOG(NONE)
DS2 LOG(UNDO)
DS3 LOG(ALL)

Catalogs
LOGS

SMS Configuration
S1 (CATCACHE_PROD/SYSLOCK)
S2 (CATCACHE_TEST/TESTLOCK)
S3 (CICS_CACHE/CICS_LOCK)
S4 (HSM_CACHE/SYSLOCK)

D1 (RLSABOVETHEBAR(YES))
D2 (RLSABOVETHEBAR(YES)) CFCACHE(DIRONLY)
D3 (RLSABOVETHEBAR(NO)) CFCACHE(DIRONLY)

SYSTEM2

JOB3 RLS_Batch
//DD1 DSN=DS2,RLS=NRI (input)

JOB4 TVS_Batch
//DD1 DSN=DS2,RLS=NRI
//DD2 DSN=DS3,RLS=NRI

CAS
RLS.PROD.CAT (S1/D1)
RLS.TESTCAT(S2/D3)

HSM (S4)

Omegamon XE for Storage

CICS AOR3
DS1-3 (S3/D1)

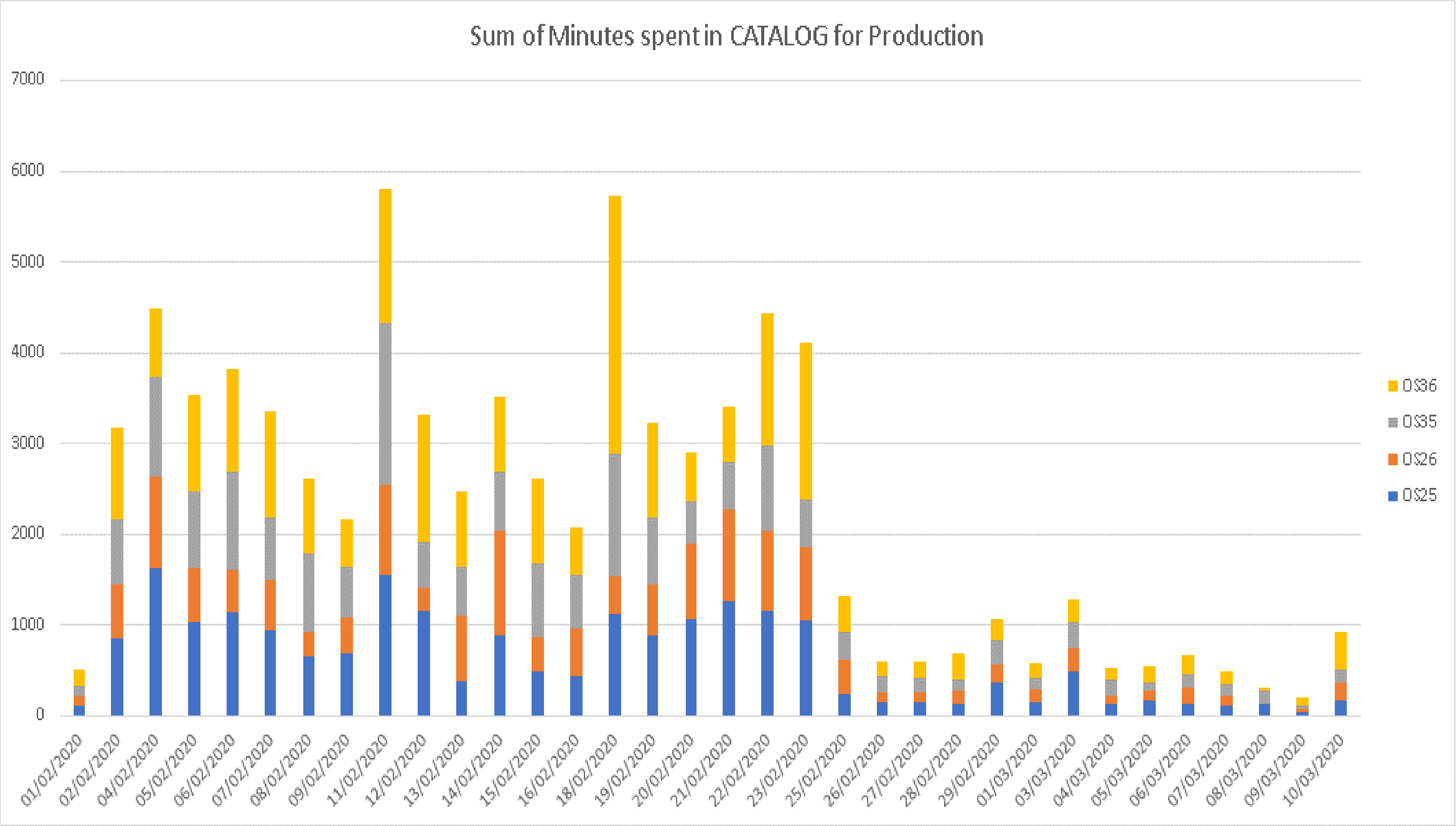
CICS AOR4
DS1-3 (S3/D1)

DVM2
(DS1-DS3) Spark

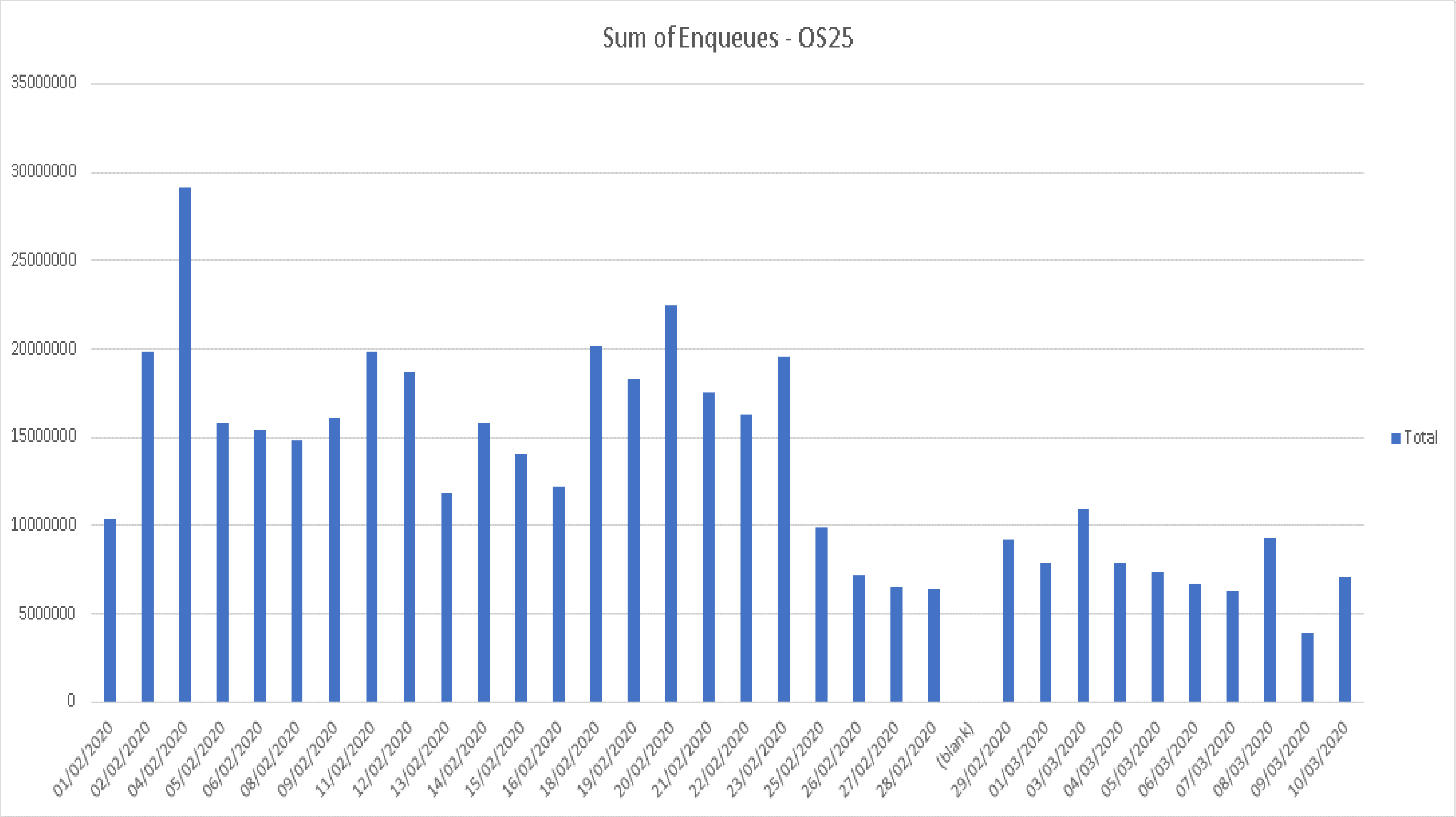
SMSVSAM
31/64 bit pool

RLS Customer Benchmarks
and
Performance Data

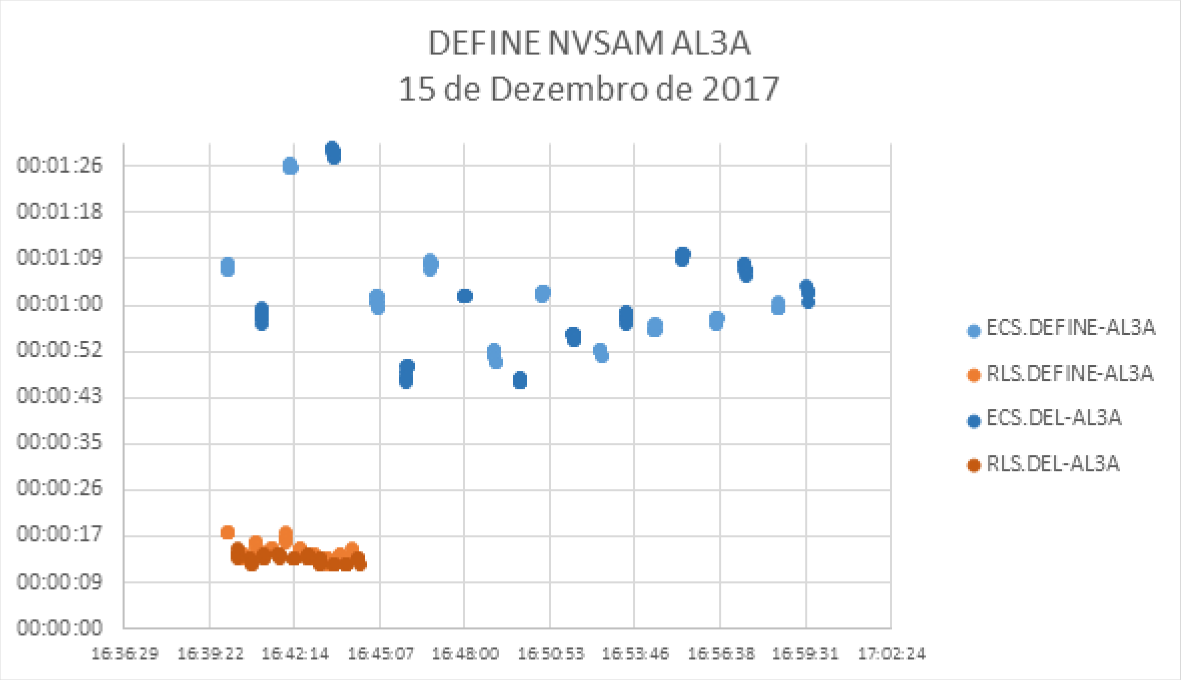
RLS Catalog Customer Benchmark

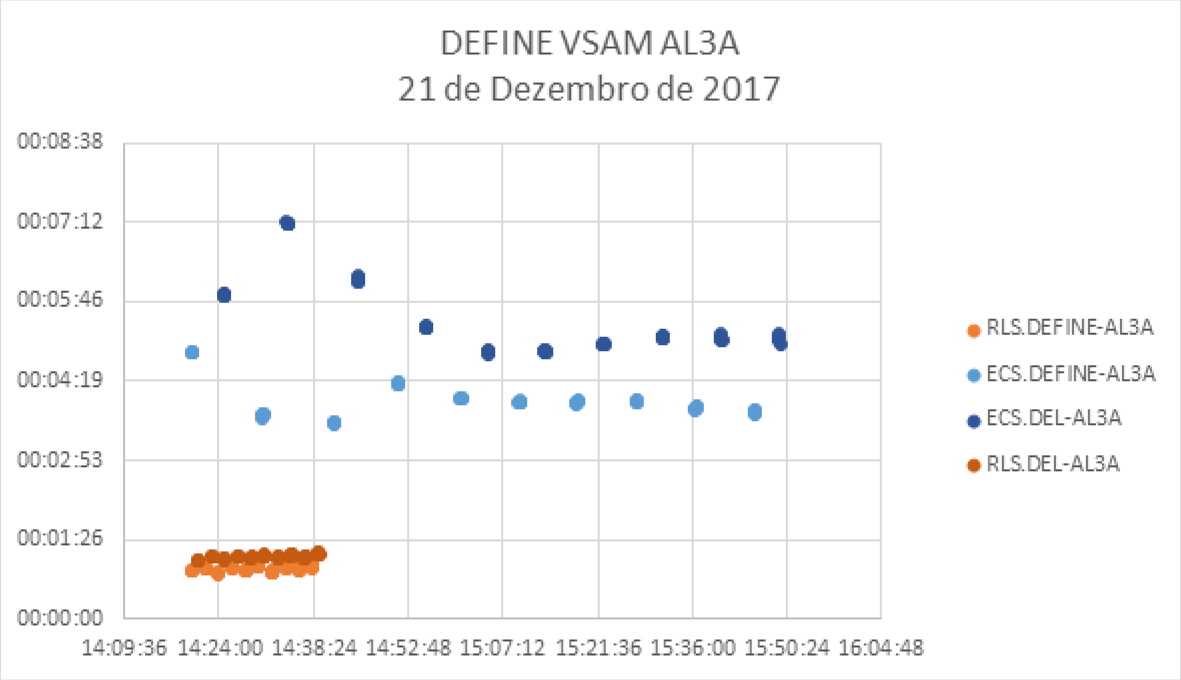


RLS Catalog Customer Benchmark



RLS Catalog Customer Benchmark


DEFINE NVSAM AL3A
15 de Dezembro de 2017


DEFINE VSAM AL3A
21 de Dezembro de 2017

RLS HSM/Catalog Customer Benchmark

SYSPLEX	Metric	NonRLS	RLS	Delta
ALL PLEXES	Elapsed Time HSM Space Mgt	19:48:18	17:22:05	12.3%
ALL PLEXES	Elapse Time HSM SMF202 (Backup/Recall/Migration)	246634342732	97592913560	60.4%
ALL PLEXES	CATSTATX Ave CPU Time	3166	869	72.5%
ALL PLEXES	CATSTATX Ave Elapse Time	12.84	2.52	80.4%

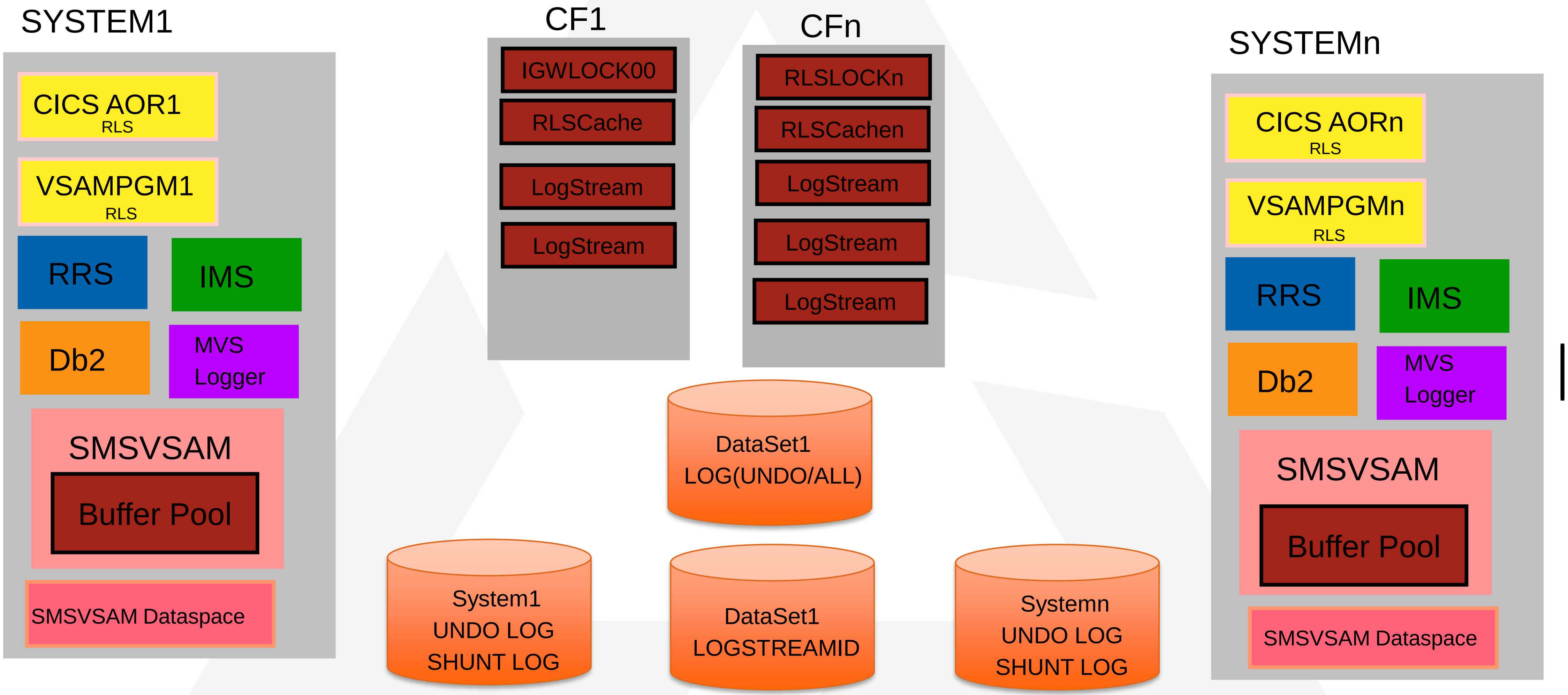
TVS Introduction

TVS Introduction

- Enhanced RLS to provide data recovery capabilities for any application exploiting VSAM RLS with LOG(UNDO/ALL) data sets.
- VSAM RLS data recovery capabilities include:
  - transactional recovery
  - data set recovery
- VSAM RLS becomes a "transactionalized" access method or is now referred to as "Transactional VSAM" (TVS).
- **Now included in the z/OS 3.1 base license.**

RLS/TVS Sysplex Example

RLS/TVS Sysplex Example



RLS/TVS System Parameters

TVS System Parameters

```
SYS1.PARMLIB(IGDSMSxx)
SMS ACDS(acds)
    INTERVAL(nnn|15)
    REVERIFY(YES|NO)
     SYSTEMS(8|32)
     SIZE(nnnnnK|M)
      JOBNAME(jobname|*)
     SELECT(event,event....)
     DSNTYPE(LIBRARY|PDS)
     RLSMAXCFFEATURELEVEL(A|Z)
     RLSINIT(NO|YES)
     CF_TIME(nnn|3600)
     CACHETIME(nnn|3600)
     RLSTMOUT(nnn|0)
     TVSNAME(nnn1,nnn2....)
      TV_START_TYPE(WARM|COLD,WARM|COLD...)
      LOG_OF_LOGS(logstream)
```

```
COMMDS(commds)
DINTERVAL(nnn|150)
ACSDEFAULTS(YES|NO)
TRACE(OFF|ON)
TYPE(ALL|ERROR)
 ASID(asid|*)
 DESELECT(event,event....)
 DSSTIMEOUT(nnn|0)
 RLS_MAX_POOL_SIZE(nnn|100)
SMF_TIME(NO|YES)
BMF_TIME(nnn|3600)
DEADLOCK_DETECTION(iii|15,kkk|4)
SYSNAME(sys1,sys2,...)
MAXLOCKS(max|0,incr|0)
AKP(nnn|1000,nnn|1000)
QTIMEOUT(nnn|300)
```

TVS System Commands

TVS Display TRANVSAM Command

SYSTEM1

```
- 13.19.03 SYSTEM1          d sms,tranvsam
  13.19.04 SYSTEM1          IEE932I 023
  IGW800I 13.19.04 DISPLAY SMS,TRANSACTIONAL VSAM
  DISPLAY SMS,TRANSACTIONAL VSAM - SERVER STATUS
   System       TVSNAME  State     Rrs         #Urs     Start                  AKP    QtimeOut
   --------       --------        ------      -----        --------    ---------                  --------  --------
   SYSTEM1  IGWTV001 ACTIVE REG          0        WARM/WARM      200      400

  DISPLAY SMS,TRANSACTIONAL VSAM - LOGSTREAM STATUS
   LogStreamName                                State        Type           Connect Status
   --------------------------                       ----------       ----------       --------------
   IGWTV001.IGWLOG.SYSLOG            Enabled    UnDoLog      Connected
   IGWTV001.IGWSHUNT.SHUNTLOG  Enabled    ShuntLog     Connected
```

TVS Display URID Command

SYSTEM1

```
  2.33.44 SYSTEM1          d sms,urid(all)
12.33.44 SYSTEM1          IEE932I 337                              C
IGW802I 12.33.44 DISPLAY SMS,URID

TRANSACTIONAL VSAM URID Status For System: SYSTEM1
Urid                                              Ur Status  JobName  StepName # Locks
-------------------------------- ---------- -------- -------- --------
DF686D297DCA70000000000001010000 InFlight    IBMUSERA  LOAD          2
```

TVS Explicit/Implicit Commits/Backouts

Application Requirements - Explicit Commit Example

//ddname DD DSN=Recoverabledatasetname,DISP=SHR,**RLS=NRI**

//step1   EXEC PGM=vsamrlspgm

Begin JOB Step   ------------------------------------ No locks held

OPEN ACB MACRF=(**RLS,OUT**)

(UR1)

GET UPD record 1------------------------------------ Obtain an exclusive lock on record 1

PUT UPD record 1 --------------------------------- Lock on record 1 remains held

GET repeatable read record n------------------- Obtain a shared lock on record n

PUT ADD record n+1------------------------------- Obtain an exclusive lock on record n+1

GET UPD record 2 --------------------------------- Obtain an exclusive lock on record 2

GET UPD record 2 ---------------------------------- Lock on record 2 remains held

Call SRRCMIT -- Commit changes, all locks released .

(UR2)

…

CLOSE

End of JOB Step

Application Requirements - Implicit Commit Example

//ddname DD DSN=Recoverabledatasetname,DISP=SHR,**RLS=NRI**

//step1   EXEC PGM=vsamrlspgm

Begin JOB Step -------------------------------------- No locks held

OPEN ACB MACRF=(**RLS,OUT**)

(UR1)

GET UPD record 1----------------------------------- Obtain an exclusive lock on record 1

PUT UPD record 1 --------------------------------- Lock on record 1 remains held

GET repeatable read record n------------------- Obtain a shared lock on record n

PUT ADD record n+1------------------------------- Obtain an exclusive lock on record n+1

GET UPD record 2 --------------------------------- Obtain an exclusive lock on record 2

GET UPD record 2 ---------------------------------- Lock on record 2 remains held

CLOSE --- All Locks are retained

End of JOB Step (normal)------------------------- Commit changes release all locks

Application Requirements - Explicit Backout Example

//ddname DD DSN=Recoverabledatasetname,DISP=SHR,**RLS=NRI**

//step1   EXEC PGM=vsamrlspgm

Begin JOB Step -- No locks held

OPEN ACB MACRF=(**RLS**,**OUT**)

(UR1)

GET UPD record 1------------------------------------- Obtain an exclusive lock on record 1

PUT UPD record 1 ----------------------------------- Lock on record 1 remains held

GET repeatable read record n-------------------- Obtain a shared lock on record n

PUT ADD record n+1--------------------------------- Obtain an exclusive lock on record n+1

GET UPD record 2 ----------------------------------- Obtain an exclusive lock on record 2

GET UPD record 2 ------------------------------------ Lock on record 2 remains held

Call SRRBACK -- Undo changes, all locks released

(UR2)

…..

CLOSE

End of JOB Step

Application Requirements - Implicit Backout Example

//ddname DD DSN=Recoverabledatasetname,DISP=SHR,**RLS=NRI**

//step1   EXEC PGM=vsamrlspgm

Begin JOB Step -------------------------------------- No locks held

OPEN ACB MACRF=(**RLS,OUT**)

(UR1)

GET UPD record 1---------------------------------- Obtain an exclusive lock on record 1

PUT UPD record 1 -------------------------------- Lock on record 1 remains held

GET repeatable read record n------------------- Obtain a shared lock on record n

PUT ADD record n+1------------------------------- Obtain an exclusive lock on record n+1

GET UPD record 2 -------------------------------- Obtain an exclusive lock on record 2

GET UPD record 2 --------------------------------- Lock on record 2 remains held

-------------------------------------- Cancel ---

End of JOB Step (abnormal) ---------------------- Undo changes release all locks

TVS Auto Commits

TVS Auto Commits Benefits

- Avoids re-design of batch jobs
- Easier transition to Transactional VSAM
- Avoids a job step as a single transaction
  - Avoid locks held (UPDATES, READ CRE)
  - Avoid large undo logs
  - Avoid deadlocks and time-out problems with other CICS applications
- Avoids long backout/recovery times if a job is cancelled

TVS Auto Commits Parameters

- New parameter in the job step JCL
  - //stepname EXEC positional-parm, TVSAMCOM=({minval},{maxval})

- New system level parameter in the IGDSMSxx member of sys1.parmlib
  - TVSAMCOM=({minval},{maxval})

- The JCL overrides the value in IGDSMSxx

Minval: Minimum number of update requests   Maxval: Maximum number of update requests

Transactional VSAM will adjust the commit frequency to a number between *minvalue* and *maxvalue* based on record lock contention analysis for the current unit of recovery

TVSAMCOM Keyword Examples

- **Example 1*: //TVSSTEP01 EXEC PGM=VSAMPGM,TVSAMCOM = (10,10)***
  - Transactional VSAM will issue a commit after 10 record updates have been performed
- **Example 2*: //TVSSTEP02 EXEC PGM=VSAMPGM,TVSAMCOM = (10,100)***
  - Transactional VSAM will dynamically adjust the commit frequency to a number between 10 and 100 based on record lock contention. If no critical record lock contention for the unit of recovery (UR) is found, an auto commit will be issued when 100 record updates have been processed
- **Example 3*: //TVSSTEP03 EXEC PGM=VSAMPGM,TVSAMCOM = (10,0)***
  - Transactional VSAM will issue a commit based on record lock contention analysis for the unit of recovery after 10 records has been updated. If Transactional VSAM does not find a critical record lock contention for the UR a commit will not be issued
- **Example 4*: //TVSSTEP04 EXEC PGM=VSAMPGM,TVSAMCOM = (,)***
  - Transactional VSAM will issue a commit point as soon as the threshold criteria based on record lock contention analysis for the UR is met. If Transactional VSAM does not find a critical record lock contention for the UR a commit will not be issued

IDCAMS Commands

IDCAMS SHCDS Commands

```
IDCAMS SHCDS    {{LISTDS(base_cluster_name) {JOBS}} |
         {LISTSUBSYS(subsystem_name|ALL)} |
         {LISTSUBSYSDS(subsystem_name)} |
         {LISTRECOVERY(base_cluster_name|ALL)} |
                          ***
        LISTSHUNTED {SPHERE(base_cluster_name) |
             URID(urid)              |
             DATA(urid)}}
        {RETRY {SPHERE(base_cluster_name) |
             URID(urid)}}
        {PURGE {SPHERE(base_cluster_name) |
            URID(urid)}}
```

SHCDS Example

```
===>   SHCDS LISTDS('recoverabledataset*')  JOBS

----- LISTING FROM SHCDS ----- IDCSH02 ---------------------------------------------------------------------------------------------------
--

 DATA SET NAME----recoverabledataset

   CACHE STRUCTURE----CACHE01

   LOCK STRUCTURE-----LOCK01

   RETAINED LOCKS---------YES   NON-RLS UPDATE PERMITTED---------NO

   LOST LOCKS--------------NO   PERMIT FIRST TIME----------------NO

   LOCKS NOT BOUND---------NO   FORWARD RECOVERY REQUIRED--------NO

   RECOVERABLE------------YES

                    SHARING SUBSYSTEM STATUS

   SUBSYSTEM   SUBSYSTEM       RETAINED      LOST       NON-RLS UPDATE

   NAME         STATUS           LOCKS         LOCKS     PERMITTED

   ---------     --------------      ----------------     ----------     --------------------------

   IGWTV001     ONLINE--FAILED   YES              NO           NO

JOB NAMES:

    IBMUSERA

 ***
```

Data Set Forward Recovery

The CICSVR utility automates the steps necessary to forward recover a data set (does not include retrying shunted work).

System Failure - Peer Recovery

- With "Peer Recovery", another SMSVSAM instance (peer) performs backouts for the failed URs which were being processed by the failed SMSVSAM instance.
- Peer recovery will restart automatically if Transactional VSAM has been included in the ARM policy.
- If ARM is not used, the following command can be issued immediately on a peer system following a failure:
  - V SMS,TRANSVSAM(tvsname),PEERRECOVERY,ACTIVE

References:

- *DFSMStvs Planning and Operating Guide:* Abstract for DFSMStvs Planning and Operating Guide - IBM Documentation
- *VSAM Demystified:* VSAM Demystified (ibm.com)
- *TVS Administration Guide:* Abstract for DFSMStvs Administration Guide - IBM Documentationb

Questions ???

Your feedback is important!

Submit a session evaluation for each session you attend:

www.share.org/evaluation