



Getting the Most out of your VSAM Data Sets in CICS by Using RLS

Andre Clark – CICS L2 Support

Neal Bohling – RLS Development

Session 17257



SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

Copyright (c) 2015 by SHARE Inc. Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nc-sa/3.0/>



Agenda

- Current Non-RLS Functionality
- Introduction to Record Level Sharing
- RLS Performance Comparisons
- Setup (DFSMS and CICS)
- Details about How it Works
 - Logging and RLS
 - Locking and RLS
 - RLS and Recoverable Datasets
 - RLS Quiesce
- Debugging
 - RLS Tips
 - CICS Tips
 - FCRVWAIT Debugging Example
- Summary

Local Shared Resources (LSR) Today

- When a task is updating a record in VSAM LSR it has exclusive control of the entire Control Interval.
 - ▷ All other tasks attempting to reference a record within the same Control Interval will be put into a FCXCSUSP (File Control Exclusive Control Conflict) until the first task completes the update.
 - ▷ Consider a dataset that has a 32K Control Interval size and 50 byte fixed records. 655 records can fit into the Control Interval. Consequently, there is a good chance another transaction will be locked out until the update has completed.

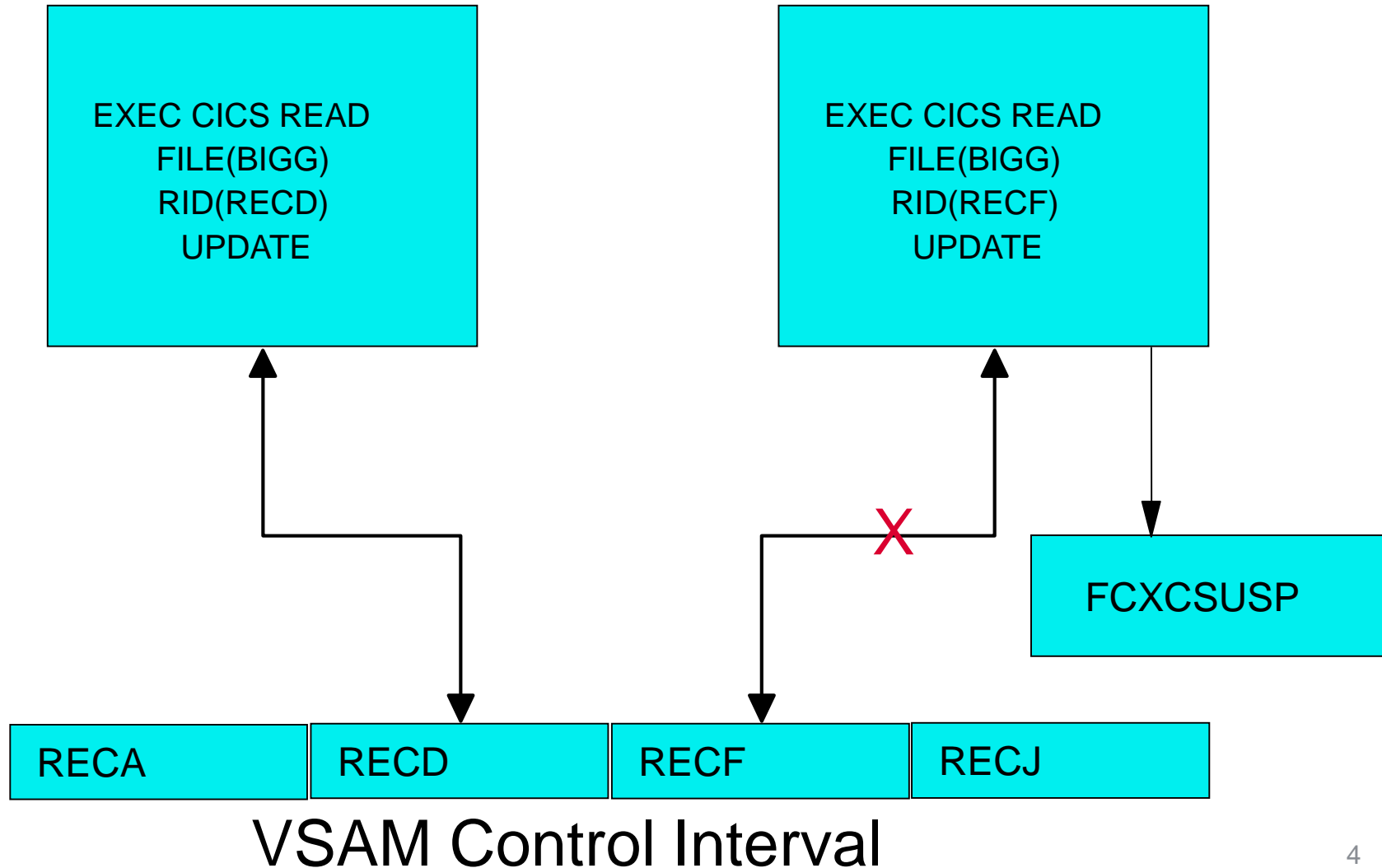
- When a task is reading or browsing a record in VSAM LSR it has shared control of the entire Control Interval.
 - ▷ If another task wants to update a record in the Control Interval it will be put into a FCXCSUSP until all tasks are through reading or browsing.

Local Shared Resources (LSR)

VSAM Record Sharing Non-RLS

Transaction 00001

Transaction 00002



VSAM Functionality Today

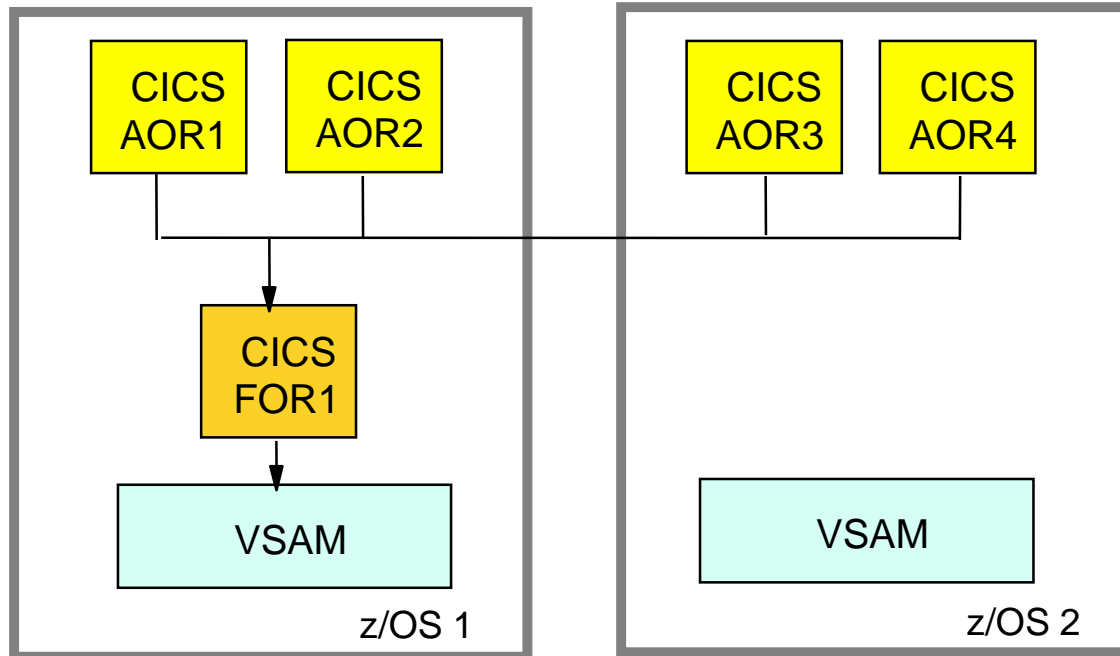
- Sharing of datasets across one MVS image
 - ▶ CrossRegion Shareoptions 1 allows any number of readers OR only one user for read and write processing. VSAM ensures complete data integrity.
 - ▶ CrossRegion Shareoptions 2 allows any number of readers AND one user for write and read processing. VSAM ensures read/write integrity for one user. Read integrity is not guaranteed for other readers.
 - ▶ CrossRegion Shareoptions 3 and 4 allow sharing by any number of readers and writers. Users are responsible for read and write integrity. Improper Enqueue and Dequeue macro usage may cause dataset damage.
- Sharing of datasets across multiple MVS systems
 - ▶ CrossSystem Shareoptions 3 and 4 allow sharing by any number of readers and writers. Users are responsible for read and write integrity. Improper Reserve and Release macro usage may cause dataset damage.
- Global Resource Serialization (GRS) or equivalent must be setup for sharing across multiple systems.

File Sharing Options in CICS Today

- **Function Shipping to File Owning Region (FOR)**
 - ▶ LU62 (SNA) connection from different MVS, or within the same MVS
 - ▶ MRO connection within the same MVS
 - ▶ MRO/XCF connection within the same Sysplex
- **External CICS Interface (EXCI)**
 - ▶ Allows applications running in a non-CICS environment to access CICS application programs.
- **Shared Data Tables**
 - ▶ Uses MVS cross-memory services to share a file between different CICS regions in the same MVS image.
- **Coupling Facility Data Tables**
 - ▶ Sharing of working data within a Sysplex with update integrity.
 - ▶ Keylength restriction of 16 bytes.
 - ▶ Cannot be forward recovered.
 - ▶ Does not survive the loss of the Coupling Facility structure in which it resides.

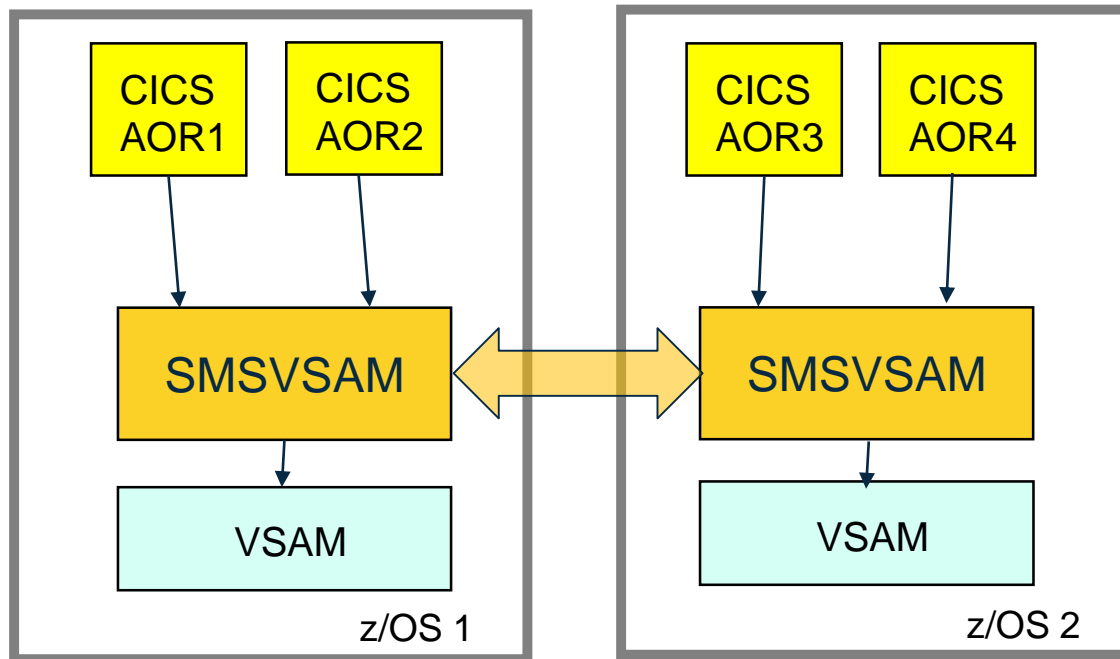
CICS Function Shipping Before VSAM RLS

AOR = Application Owning Region
FOR = File Owning Region



- ◆ CICS FOR is a single point of failure

CICS Using VSAM Record Level Sharing



SMSVSAM acts as PLEX-wide middle man to ensure serialization and buffer updates.

Introduction to Record Level Sharing (RLS)

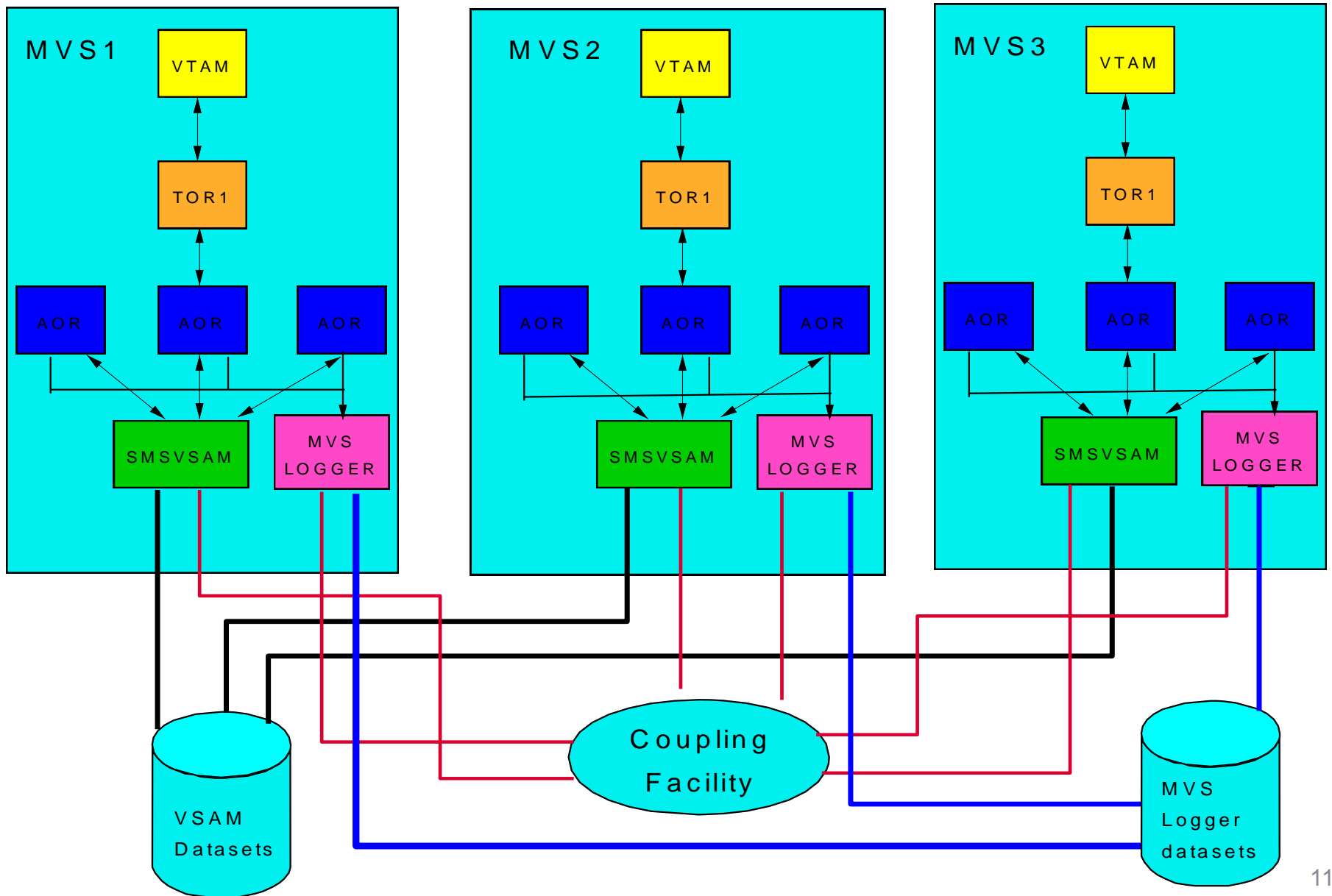
- Provides full VSAM data sharing across a SYSPLEX
- Locking is done at the Record Level instead of Data-Set Level
- SMSVSAM uses large buffer pools and caching for maximum cross-system read speed
- RLS Mode is determined at OPEN time

- RLS Access mode is available for all VSAM files managed by CICS File Control (DFHCSD restriction is now removed)
 - VSAM data sets NOT managed by CICS File Control are not supported in RLS mode
 - Transient Data Intrapartition Data Set (DFHINTRA)
 - Temporary Storage Data Set (DFHTEMP)
 - Local and Global catalogs (DFHLCD and DFHGCD)

Introduction to Record Level Sharing

- Once the dataset is opened in RLS mode within CICS
 - ▶ Record locking only, no Control Interval locking
 - ▶ All access to the dataset must be in RLS mode within CICS
 - VSAM allows readonly shareoption 2 opens with OW25251 applied.
 - ▶ Dataset has to be quiesced throughout the Sysplex before switching to non-RLS
 - Every CICS region in the Sysplex that has the RLS dataset open can close the Dataset to switch to non-RLS - DS can then be opened in RLS or non-RLS mode
 - One CICS region in the Sysplex can issue SET DSNAME() QUIESCED command - Dataset can only be opened in non-RLS mode until UNQUIESCE is issued
- VSAM is responsible for the locking
- CICS is responsible for the logging
 - ▶ System log for backout
 - ▶ Forward recovery logs
 - ▶ Automatic journals

Parallel Sysplex with RLS



Non-RLS vs RLS Summary

- **Using Non-RLS**
 - Serialization at the data set level causing contention
 - Under certain SHAREOPTIONS, no serialization at all
 - Function Shipping provides more granular serialization, but opens up single point of failure
 - No PLEX-wide buffer sharing (caching)
- **Using Record Level Sharing (RLS)**
 - Full PLEX-wide serialization at the record level
 - No non-RLS access allowed; serialization enforced
 - Data is buffered and cached to maximize performance
 - No single point of failure

RLS Performance Comparisons

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

RLS Performance

- IBM Redpaper **IBM CICS Performance Series: CICS and VSAM RLS**
 - <http://www.redbooks.ibm.com/abstracts/redp4905.html>
- **Sysplex configuration**
 - The sysplex consisted of two LPARs, each with three dedicated CPs on an IBM 2097-763 Model E64. There were two integrated CFs with three shared CPs each.
- **CICS configurations**
 - CICS Transaction Server Version 4.1 was used, running on z/OS Version 1.11. Network simulators were used to drive the workload at approximately the same transaction rates for each scenario. All transactions entered the TOR and were routed to the AOR, where file requests were either function-shipped or accessed locally as LSR or RLS files.
- **The workload consisted of COBOL applications with little business logic. The applications were not defined as threadsafe. The workload had the following characteristics:**
 - Average of six file requests per transaction
 - CICS SIT parameter MROLRM=YES
 - CICS file requests: 69% Read, 10% Read for Update, 9% Update, 11% Add, 1% Delete
 - CICS SIT parameter FCQRONLY=YES
 - LOG(UNDO) attribute specified on the VSAM Cluster definitions
 - The RLS buffer pool size, CF caches, and so on, were sized to avoid any major performance constraints in the system.

Scenario One: Local LSR Access in AOR

- Table 1 displays the results for the first scenario involving a TOR and AOR where all of the files were accessed locally in the AORs by means of LSR
- Using the data recorded here, the average CPU time per transaction across the five intervals can be calculated to be 0.34 ms.

Table 1 Local LSR access in the AOR

ETR	TOR%	AOR%	TOT CPU%	CPU/TRAN	RESP TIME
423.46	3.79	11.35	15.14	0.357 ms	7 ms
531.61	4.72	14.07	18.79	0.353 ms	7 ms
702.08	6.14	18.24	24.38	0.347 ms	7 ms
1041.40	9.03	27.04	36.07	0.346 ms	7 ms

Scenario Two: MRO Function Shipping

- Table 2 provides the results for the second scenario, which introduced an FOR and function-shipped all of the file requests through MRO
- After migrating to MRO function shipping, the average CPU time per transaction has now gone up to 0.47 ms.

Table 2 MRO Function Shipping

ETR	TOR%	AOR%	FOR%	TOT CPU%	CPU/TRAN	RESP TIME
427.41	4.09	10.68	6.42	21.19	0.495 ms	11 ms
536.87	5.05	13.34	7.79	26.18	0.487 ms	11 ms
711.31	6.51	17.36	9.91	33.78	0.474 ms	11 ms
1065.26	9.52	25.57	14.67	49.76	0.467 ms	11 ms
1426.78	12.56	33.54	19.04	65.14	0.456 ms	11 ms

Scenario Three: XCF Function Shipping

- Table 3 shows the results of the third scenario, which moved the FOR to another LPAR in the sysplex so that all of the file requests could be function-shipped across XCF using a coupling facility.
- By moving the FOR to another LPAR and then function shipping using the coupling facility, CPU usage was introduced in the XCFAS address spaces on each LPAR. Now the average CPU per transaction has gone up to 0.75 ms.

Table 3 XCF Function Shipping

ETR	TOR%	AOR%	XCFAS%	FOR%	XCFAS%	TOT CPU%	CPU/TRAN	RESP TIME
426.75	3.93	13.43	3.13	8.30	3.54	32.33	0.757 ms	12 ms
539.29	4.93	16.90	4.05	10.41	4.42	40.71	0.754 ms	12 ms
720.57	6.52	22.49	5.48	13.75	5.92	54.16	0.751 ms	12 ms
1072.64	9.69	33.42	8.17	20.04	8.92	80.24	0.748 ms	12 ms
1435.46	13.09	44.80	11.20	27.06	11.98	108.13	0.753 ms	12 ms

Scenario Four: RLS Uncommitted Read

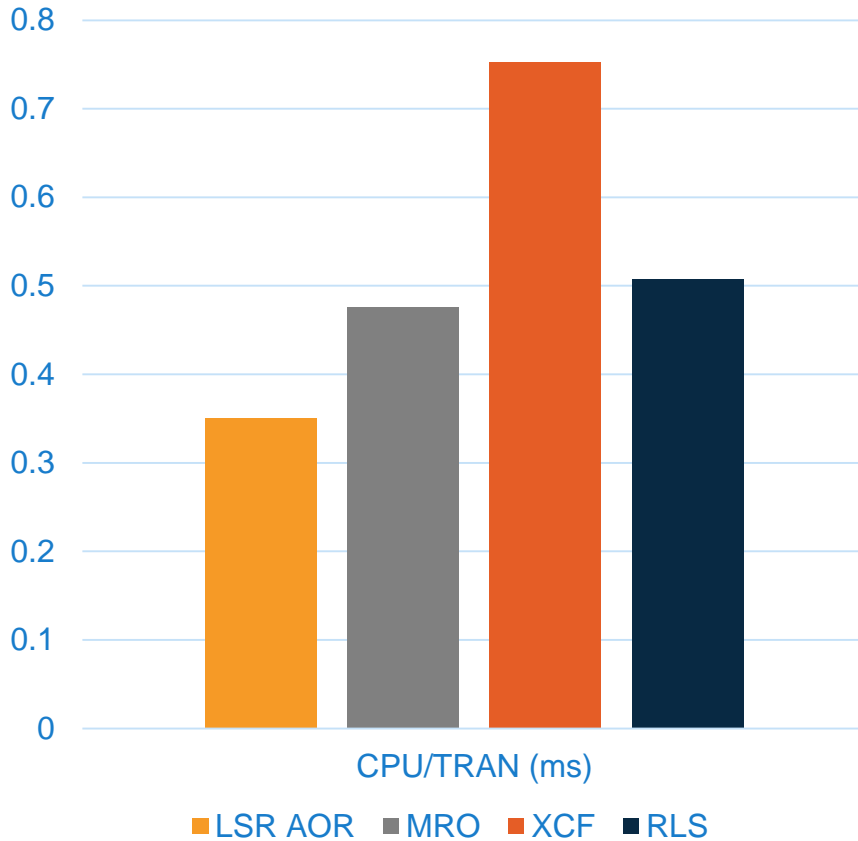
- The results for the fourth scenario are shown in Table 4. The FOR was removed and all of the files in the AORs were redefined as RLS files, meaning that all file requests were now RLS accesses through the SMSVSAM address space.
- This has the same characteristics as LSR files in that it is possible to read a record that has been read for update by another task, which could change it with a rewrite.
- In this scenario, the FOR was removed from the configuration and all of the files in the AORs were defined as RLS with Uncommitted Read (UR) integrity. If we were comparing this to XCF function shipping, then, as the data shows, the CPU per transaction has gone down to an average of 0.49 ms.

Table 4 RLS Uncommitted Read

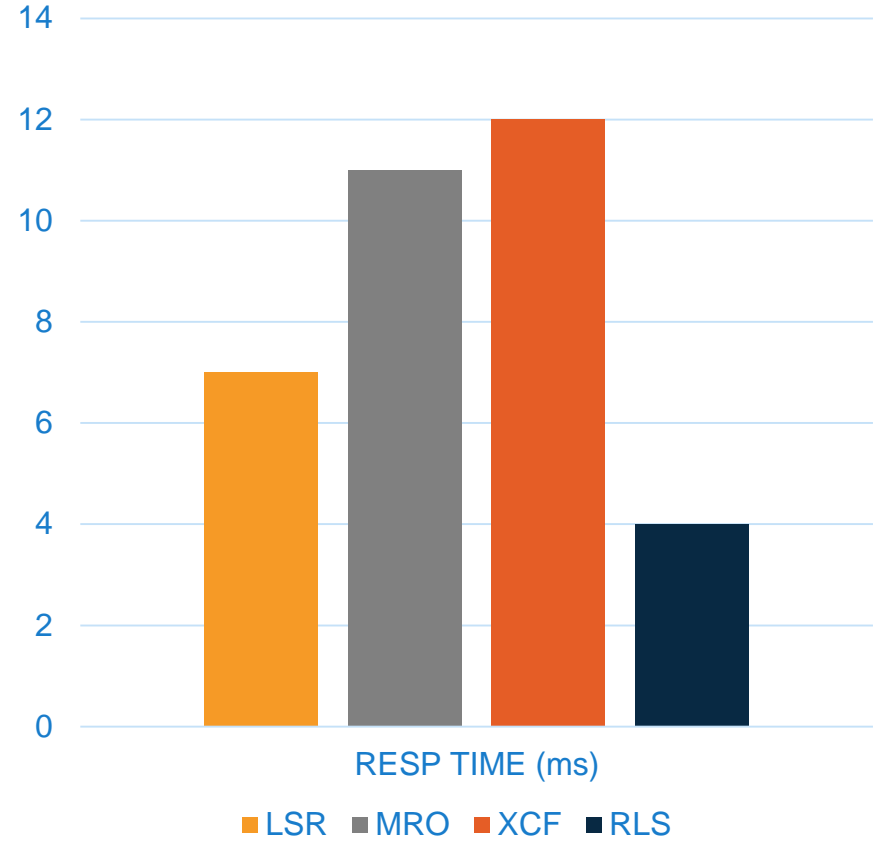
ETR	TOR%	AOR%	SMSVSAM%	TOT CPU%	CPU/TRAN	RESP TIME
423.50	3.93	17.40	0.48	21.87	0.515 ms	4 ms
532.29	4.83	21.46	0.55	26.84	0.504 ms	4 ms
703.15	6.30	28.08	0.66	35.04	0.498 ms	4 ms
1042.72	9.12	40.91	0.90	50.93	0.499 ms	4 ms
1378.79	12.07	54.11	1.17	67.35	0.488 ms	4 ms

Comparisons

CPU per Transaction



Response Time per Transaction



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Setting up DFSMS for RLS

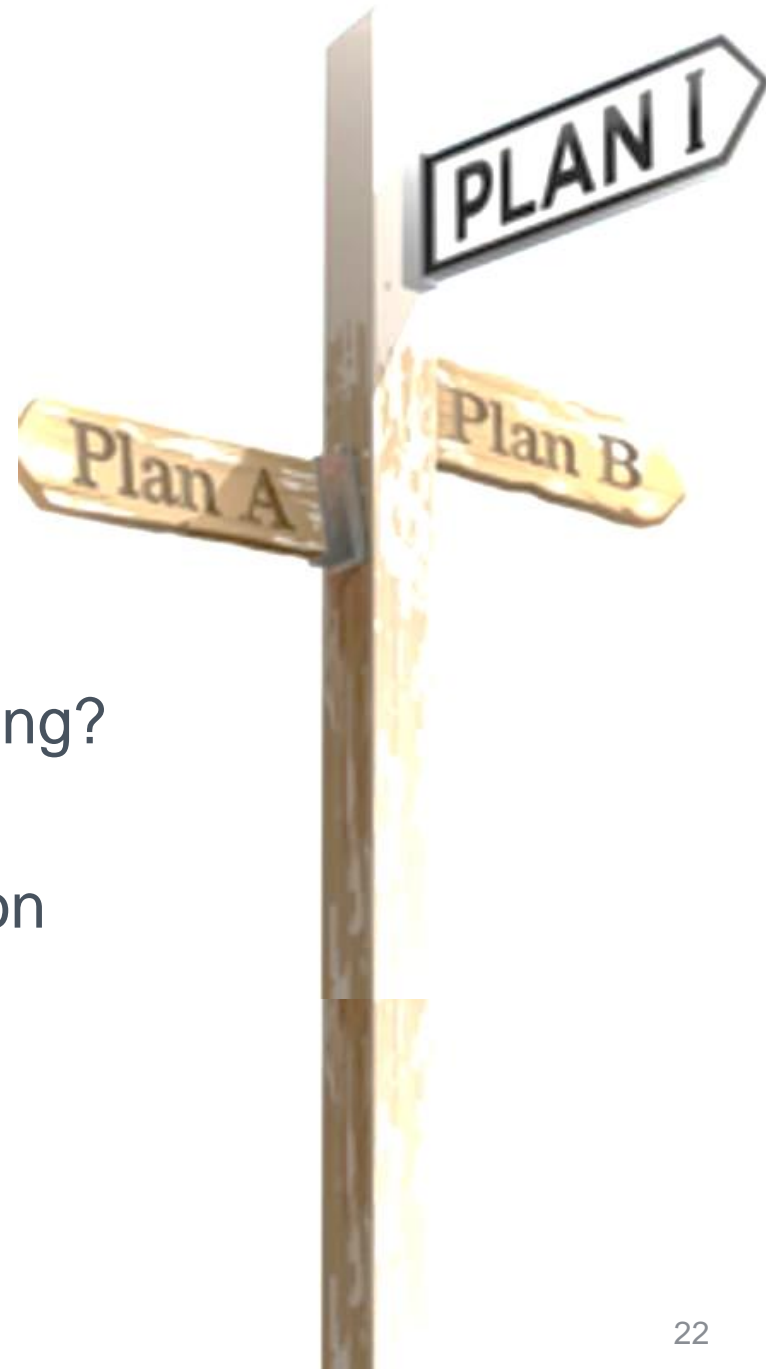
Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Step #1

PLAN

Consider:

- How much data will be in RLS?
- How much cross-system data sharing?
- How big should the buffers be?
- Want separation between production and development?



Overview of RLS Setup

SHCDS

- Define Share Control Data Sets

CF

- Define CFRM Policy
- Define Lock Structures
- Define Cache Structures

SMS

- Update IGDSMSxx
- Update SMS Configuration
- Update ACS Routines

Share Control Data Sets

- Maintains Integrity in case of SMSVSAM or CF failure
- Holds information related to subsystems, data sets, and locking
- **Two ACTIVE, at least one SPARE**
 - Linear VSAM, named SYS1.DFPSHCDS.Vvolser
 - CISIZE(4096), SHR(3,3), single volume
- Note: SHCDS must be accessible to all systems running SMSVSAM, and should not be shared outside the plex.

Coupling Facility Structures

- Define CFRM policy with one of more:

Lock Structures

- IGWLOCK00 (required)
- At least 20MB per system
- Any number of secondary lock structures

Cache Structures

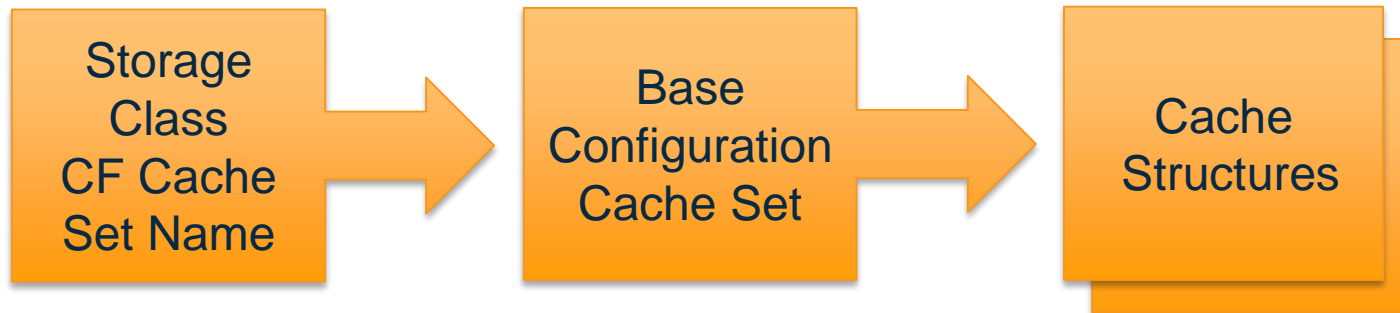
- Any number of structures
- At least 30MB
- Ideally, large enough to hold all buffers

IGDSMSxx Changes

- Add / Update the following keywords
 - RLSINIT(YES) – starts RLS
 - RLS_MAXCFFEATURELEVEL(A)
 - RLS_MAX_POOL_SIZE(xxx) - buffer sizes
 - RLSABOVETHEBARMAXPOOLSIZE(ALL,xxxx)

Update SMS Configuration

- Base Configuration
 - One or more Cache Sets
 - If using MLS, one or more Lock Sets
- Data Class
 - CF Cache Value (ALL | NONE | UPDATESONLY | DIRONLY)
- Storage Class
 - Specify CF Cache Set name



Finishing Up

- Update your ACS routines – use new DC/SC
- Define / Alter your VSAM data sets with LOG() parm
 - NONE – RLS eligible, but logs nothing, non-recoverable
 - UNDO – Logs backward recovery, recoverable
 - ALL – Logs forward and backward recovery, recoverable
- Start SMSVSAM via V SMS,SMSVSAM,ACTIVE
 - You'll see IGW414I if everything is set up

Useful Commands

- SETSMS RLSINIT(YES) – change SMS parm on the fly
- D SMS,SMSVSAM[,ALL] – show status
- V SMS,SMSVSAM,ACTIVE – starts SMSVSAM
- V SMS,SMSVSAM,TERMINATESERVER – clean stop
- FORCE SMSVSAM,ARM – quick stop
- D SMS,SMSVSAM,QUIESCE – shows quiesce activity

CICS SETUP FOR RLS

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

CICS Setup for RLS

- New SIT parameters
 - CSDRLS (YES/NO)
 - Specifies if DFHCSD is to be opened in RLS mode
 - Default is NO
 - CSDINTEG (Uncommitted/Consistent/Repeatable)
 - Level of read integrity if DFHCSD is opened in RLS mode
 - Default is Uncommitted
 - FTIMOUT
 - Timeout interval for RLS requests
 - DTIMOUT still takes precedence
 - Default is 30 Seconds
 - RLS (YES/NO)
 - Specifies CICS supports RLS
 - Default is NO
 - QUIESTIM
 - Timeout value for quiescing a RLS dataset
 - Default is 240 Seconds
 - NONRLSRECOV
 - Specifies whether CICS uses the recovery options of the VSAM catalog or the FILE resource within CICS for non-RLS files, including the CSD.

CICS Setup for RLS

- New File definition parameters
 - ▶ READINTEG - Level of read integrity
 - UNCOMMITTED - No read integrity
 - CONSISTENT - Data has been committed
 - REPEATABLE - Multiple reads within the same unit of work give the same results
 - Default is UNCOMMITTED
 - ▶ RLSACCESS - If CICS is to open the dataset in RLS mode
 - Default is NO
- Changed File definition parameters
 - ▶ RECOVERY and FWDRECOVLOG are only meaningful for non-RLS files

Logging and RLS within CICS

- CICS performs all logging and journaling
- CICS invokes MVS Logger for all logging
 - ▶ System Logs (DFHLOG and DFHSHUNT)
 - ▶ Forward Recovery Logs
 - Managed at the dataset level
 - Logstream name is obtained from VSAM catalog
 - ▶ Autojournal
 - Managed at file level
 - Different files pointing to the same dataset can autojournal to different streams

Logging and RLS within CICS

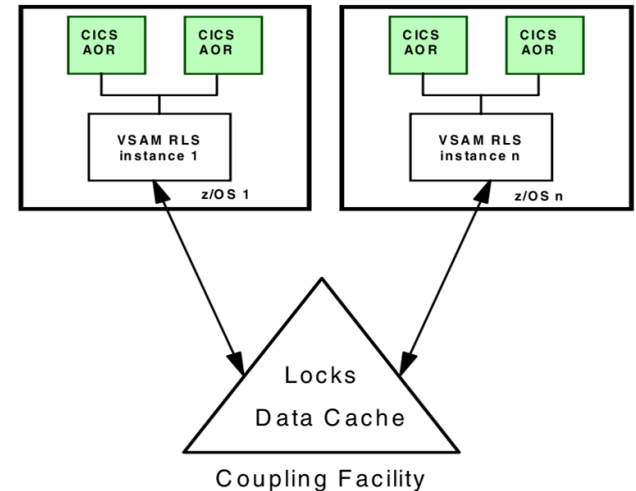
- System Logs (DFHLOG and DFHSHUNT) are used for backout of transactions. Each CICS region will have their own set of System Logs. Consequently, these may be DASD-Only or Coupling Facility logstreams.
- Forward Recovery Logs for RLS are used by products such as CICS/VR to forward recover datasets. These can also be DASD-Only or Coupling Facility logstreams. However, if they are DASD-Only the RLS dataset cannot be opened in more than One MVS in the Sysplex.
 - ▶ If the RLS dataset is opened with a Forward Recovery Log that is DASD-Only in MVS1, an Open on the same dataset from MVS2 will receive:
 - +DFHLG0784 The MVS logger has returned an error during IXGCONN CONNECT for log stream USASSC1.IYNX9.DFHJ05. You cannot connect to a DASDONLY log stream that is already connected to another MVS image. MVS logger codes: X'00000008' X'000008E2'.**
 - ▶ Since a Coupling Facility is a requirement of RLS, it is recommended that the forward Recovery logstreams are also defined to the Coupling Facility.
- DFHLGLOG is used by CICS/VR to tie-up opens and closes to a recoverable dataset. For RLS, it is recommended that all CICS regions in the Sysplex share the same DFHLGLOG. Therefore, it should be defined as a Coupling Facility logstream.

Locking in CICS Before RLS

- Locking in CICS before RLS
 - ▶ VSAM acquires a Control Interval lock while an update request is active
 - ▶ After the update is complete CICS will acquire a record lock if the dataset was defined as recoverable
 - ▶ Record locks are held until the transaction issues an explicit or implied SYNCPOINT
 - ▶ Record locks are not known by other CICS regions
 - ▶ Record locks are not known by VSAM

Locking with RLS

- Locks are managed by SMSVSAM
- Locks are Sysplex wide
- Locks are at the record level
 - ▶ A named record is locked
 - ▶ Lock is owned by a named UOW and a named CICS
- Update requests are queued until the lock is released
- For non-recoverable requests the lock is released at the end of the request
- For recoverable requests the lock is released at the end of a syncpoint



Types of RLS locks

- Exclusive Non-Recoverable Resource
 - ▶ Single owner Active lock held until end of request
 - ▶ Used to serialize updates to file resources
 - ▶ Owned by only one transaction at a time
 - ▶ Other transactions will wait for the lock

- Exclusive Recoverable Resource
 - ▶ Same as above except:
 - Held until syncpoint or end of task
 - Will be converted to a Retained lock for any failure

- Shared
 - ▶ Used to support read integrity
 - ▶ May be owned by several transactions
 - ▶ Transactions requesting the lock exclusive will wait

Deadlock and Timeout

- Deadlocks may now be between transactions in different CICS regions
 - ▶ For RLS files, VSAM detects
 - Deadlocks between RLS requests
 - Timeouts caused by locks held for an excessive time
 - Timeouts caused by waits between RLS and other resource managers
 - ▶ Timeout value based on DTIMOUT or FTIMEOUT DFHSIT parameter
 - DTIMOUT takes precedence
 - ▶ VSAM passes back diagnostic information that CICS uses to produce messages
 - DFHFC0164 IYNX9 URLS 00049 TC16 USASSC1. A request has timed out waiting for an RLS lock. There are 1 transactions holding this lock.
 - DFHFC0165 IYNX9 URLS 00049 TC16 USASSC1. Transaction URLS (00037) unit of work X'B61572D0751E8440' running in job IYNXU with applid IYNXU in MVS MV21 holds exclusive lock on key X'D9C5C3F1' in data set USASSC1.CICSR530.RLSRECV causing true contention.

RLS and Recoverable Datasets

- Recoverable datasets participate in syncpoint protocols
- Changes made by a unit of work are all either committed forward or committed backward
- Recovery attributes for RLS have been moved to the VSAM ICF catalog
 - ▶ LOG(UNDEFINED | NONE | UNDO | ALL)
 - ▶ LOGSTREAMID(log_stream_name)
 - ▶ BWO(UNDEFINED | TYPECICS | TYPEIMS | TYPEOTHER)
- CICS will check for inconsistencies in catalog definitions at open time
- FCT defined recovery attributes are not used for RLS files

RLS and Sharing Non-Recoverable Datasets

- Non-Recoverable data sets are those defined with LOG(NONE)
- Any number of CICS systems can concurrently update a non-recoverable RLS dataset
- Any number of batch jobs can concurrently update a non-recoverable RLS dataset

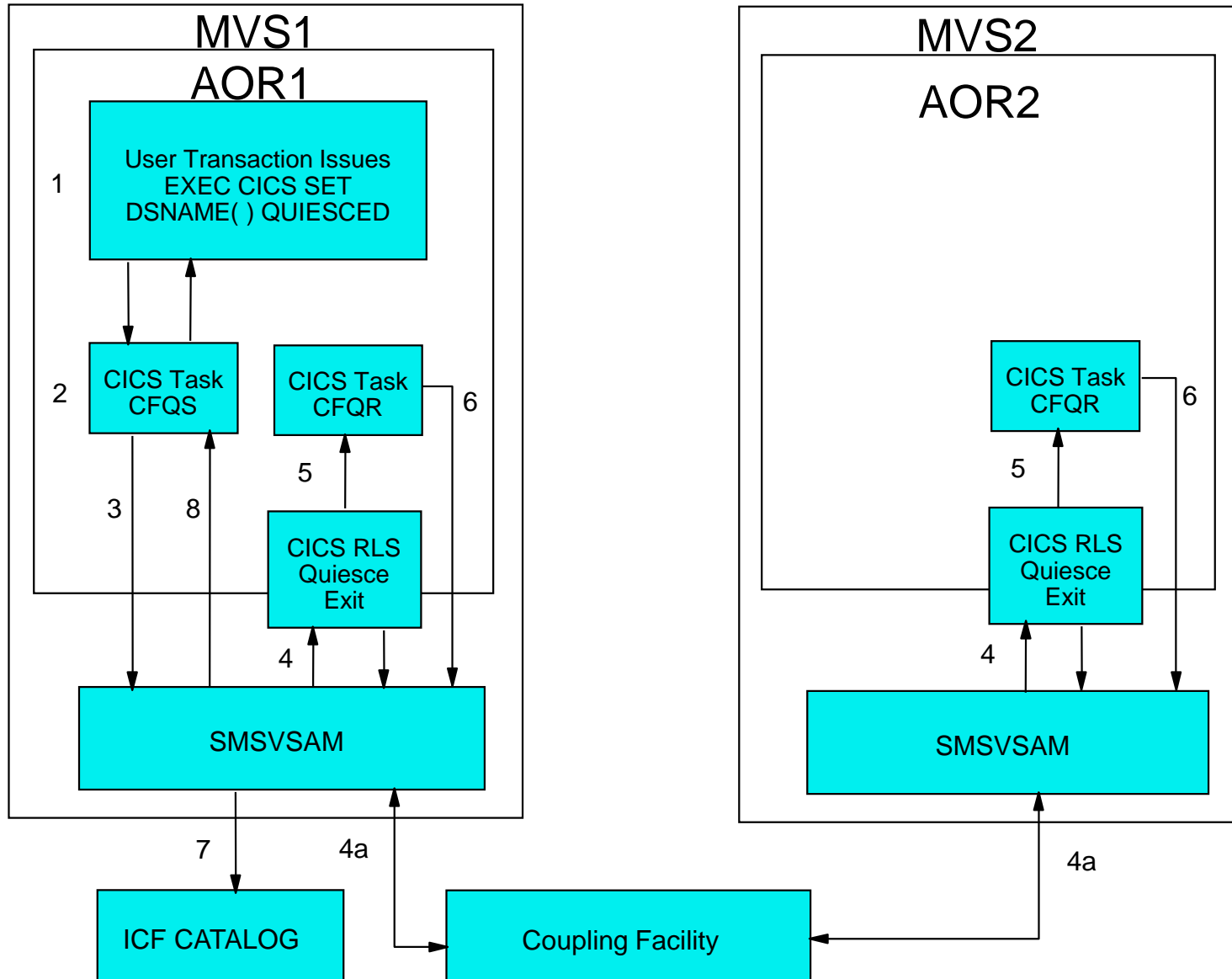
RLS and Sharing Recoverable Datasets

- Recoverable data sets are defined with LOG(UNDO | ALL)
- Any number of CICS systems can concurrently update a recoverable RLS dataset.
- Any number of batch jobs can concurrently read a recoverable RLS dataset
- A batch job may not open a recoverable dataset in RLS mode for update
- To update a recoverable dataset from a batch job, you must first quiesce RLS activity and update the dataset in non-RLS mode.
- Transactional VSAM allows batch to concurrently share a recoverable RLS dataset with CICS

RLS Quiesce

- Closes all RLS ACBs in a Sysplex for a dataset
 - ▶ Quiesce request is propagated to all regions with an open ACB
 - ▶ All CICS regions within the Sysplex close the file
 - ▶ After close is complete only non-RLS opens are allowed
 - Dataset can be Unquiesced to allow RLS open
 - ▶ A bit is set in the VSAM ICF Catalog to indicate the dataset is quiesced
- Two long running tasks in each CICS region that has RLS capability
 - ▶ CFQS - Quiesce Send
 - ▶ CFQR - Quiesce Receive

CICS RLS Quiesce Operation

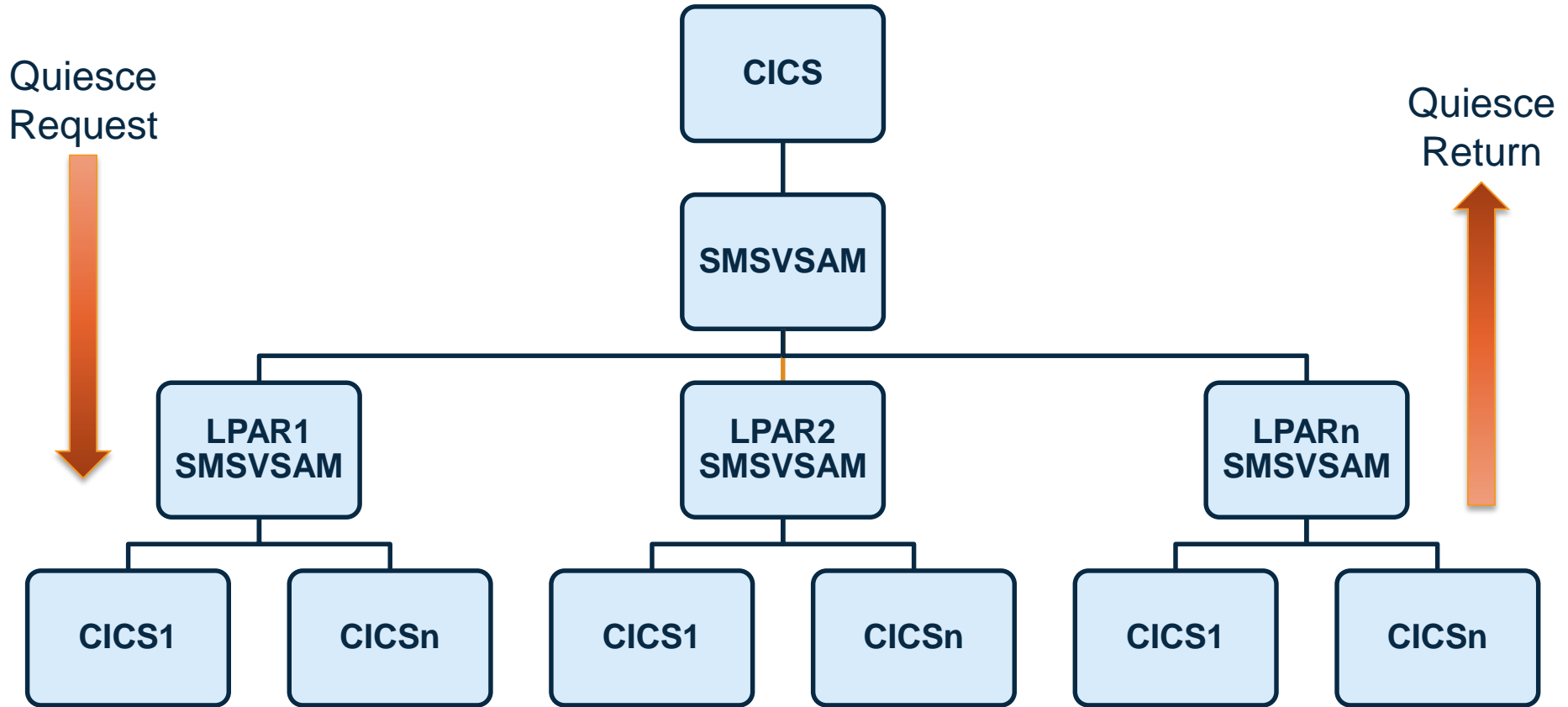


CICS RLS Quiesce Operation

1. An authorized program issues EXEC CICS SET DSNAME() QUIESCED command.
If the command specifies the BUSY(NOWAIT) option, all phases of the quiesce operation are asynchronous, and the user application program continues processing.
If the command specifies BUSY(WAIT) control is not returned to the user application until SMSVSAM replies to the CFQS task that the quiesce function has completed, or failed.
2. CICS File Control invokes the CFQS transaction to send the quiesce request to SMSVSAM.
3. The long-running CICS CFQS task passes the quiesce request to SMSVSAM.
4. SMSVSAM drives the CICS RLS quiesce exit program of all the CICS regions that have open RLS ACBs for the specified dataset. This also applies to the CICS region that initiated the request if it has open RLS ACBs for the dataset.
 - 4a. SMSVSAM uses the coupling facility to propagate the quiesce request to the other SMSVSAMs in the Sysplex.
5. The CICS RLS quiesce exit program schedules a CICS region task (CFQR) to perform the required quiesce actions in that CICS region.
6. When CICS has closed all open RLS ACBs for the dataset, CICS issues the quiesce completed notification direct to SMSVSAM.
7. When all CICS regions have replied that the dataset has been quiesced SMSVSAM sets the quiesced flag in the ICF catalog. This prevents any files being opened in RLS mode for the dataset, but allows non-RLS open requests.
8. SMSVSAM returns to the CICS region that initiated the quiesce request. If BUSY(WAIT) was specified on the request CFQS will resume the waiting application program.

Debugging Displays / Hints / Tips

Quiesce Pattern



Quiesce Tips

- QUIESCE for close only involves regions with DS open
- UNQUIESCE involves ALL registered regions
- To check for unresponsive regions, use:

D SMS,SMSVSAM,QUIESCE

IGW540I 22.56.56 DISPLAY SMS,SMSVSAM,QUIESCE

SPHERE NAME: MY.VSAM.SPHERE

SYSTEM NAME: SYS1 START TIME: 22.10.24 TOTAL ELAPSE TIME: 00.46.32<<

PARTICIPATING SUB-SYSTEM STATUS:	SCHEDULED:	COMPLETED:	ELAPSE:
SUB-SYSTEM NAME: CICSA	22.10.24	22.10.24	00.00.00
SUB-SYSTEM NAME: CICSE	22.10.24	00.00.00	00.46.32<<
SUB-SYSTEM NAME: CICSB	22.10.24	22.10.24	00.00.00
SUB-SYSTEM NAME: CICSC	22.10.24	22.10.24	00.00.00
SUB-SYSTEM NAME: CICS D	22.10.24	22.10.24	00.00.00
SUB-SYSTEM NAME: CICS F	22.10.24	22.10.24	00.00.00

Retained / Lost Locks

- Used to preserve serialization in case of error
- Use **SHCDS LISTSUBSYS** and **LISTSUBSYS**
- **UNDER Normal Circumstances, CICS knows how to clean up**

RETAINED LOCKS

- If a subsystem or UOW finishes without releasing locks
- Use INQUIRE DSNAME() RETLOCKS
- Use INQUIRE UOWDSNFAIL to find UOW that failed and why
- Use SET DSNAME to address failed UOW – retry or purge

LOST LOCKS

- Marked at the Data Set Level
- Occurs when SMSVSAM and CF fail (all lock information is lost)
- CICS will normally handle lost locks automatically:
- Use INQUIRE DSNAME() LOSTLOCKS to identify data sets
- To abandon locks:
 - COLD START the region
 - SET DSNAME RESETLOCKS

Buffering Considerations

- Pool sizes are a GOAL, not a LIMIT
- LRU reclaims buffers differently above/below the bar:
 - Below: starts reclaiming buffers at **120%**, Panic at 200%
 - Above: starts reclaiming at **90%**, Panic at 100%
- Implications:
 - Recommendation: set below-the-bar at 850mb or lower
 - If you'll need more than 1GB, use above-the-bar
- Too few buffers results in thrashing / false invalids / high CPU

Measurements

- SMF42 records show RLS stats:
 - Subtype 15 – statistics at Storage Class level
 - Subtype 16 – statistics at Data Set Level (if enabled)
 - Enable via V SMS,MONDS(dsn),ON
 - Subtype 17 – locking statistics at system / plex level
 - Subtype 18 – cache statistics
 - Subtype 19 – Buffering / LRU Statistics
- RMF Monitor III provides same info
- Omegamon XE also supports RLS

RMF Monitor III - RLSSC



RMF V2R1 VSAM RLS Activity - CAPTKEN1

Line 1 of 5

Samples: 120 Systems: 1 Date: 03/08/14 Time: 17.28.00 Range: 120 Sec

< 2GB / > 2GB
 LRU Status : Good / Good
 Contention % : 0.0 / 0.0
 False Cont % : 0.0 / 0.0

Stor Class	Access	Resp Time	----- Rate	Read BMF%	----- CF%	----- DASD%	----- Valid%	BMF False	----- Inv%	Write Rate
RLSSC1										
Below 2GB	DIR	0.000	0.00	0.0	0.0	0.0	0.0	0.00		0.00
	SEQ	0.000	0.00	0.0	0.0	0.0	0.0	0.00		0.00
Above 2GB	DIR	0.001	2303	99.4	0.0	0.6	100	0.00		1158
	SEQ	0.000	0.00	0.0	0.0	0.0	0.0	0.00		0.00

RMF Mon III - LRU

RMF V2R1 VSAM LRU Overview - CAPTKEN1 Line 1 of 3

Samples: 100 Systems: 1 Date: 03/08/14 Time: 15.23.20 Range: 100 S

MVS System	Avg CPU Time	- Buffer Goal	Size High	- Accl %	Reclaim %	----- BMF%	Read CF%	----- DASD%
SYSTEM1								
Below 2GB	<.001	100M	10M	0.0	0.0	0.0	0.0	0.0
Above 2GB	<.001	1000M	328M	0.0	0.0	99.4	0.0	0.6

Additional Tips

- You can use separate lock structures and caches to keep development and production workload separate (sort of)
- If you plan to also use RLS for CATALOG, use different storage classes – helps with monitoring and separation
- ESDS and RLS tends to be slow
 - All updates lock at Add-to-End lock
 - SMF42 subtype 15 field SMF42FOU (component 2 contention)
- Be mindful with READINTEG and RECOVERABLE
 - CONSISTENT cannot read if another UOW holds the lock
 - REPEATABLE will hold locks until SYNCPOINT
 - Can hold up other work if not careful

Useful Health Checks

- **VSAMRLS_CFLS_FALSE_CONTENTION**
 - Will start hitting if your lock structure is too small
 - Caused by Hash Conflicts in the lock table
- **VSAMRLS_DIAG_CONTENTION**
 - Notes internal SMSVSAM contention
 - Recommend that you set PARM(FILTER(5))
- **VSAMRLS QUIESCE_STATUS**
 - Helps identify unresponsive regions
 - Has configurable threshold

Using IGWFPMAN

- Unsupported Diagnostic Tool
- **VERBX IGWFPMAN 'F(xx) JOBNAME(SMSVSAM)'**
- **SMLS** – System Managed Locking System
 - Shows all active lock requests
- **AS** – Address Space Report
 - Shows all active TCBs and the module call flow
 - Also shows latch contention
- **POOLS** – pools report
 - Shows pool size and usage statistics
 - Useful for examining buffer sizes
- **BMF** – Buffer Management Facility
 - Shows buffer sizes, goals, and which data sets are buffered
 - Also shows LRU mode

Example Debugging FCRVWAIT

Debugging an FCRVWAIT

- If you have a task waiting on resource type FCRVWAIT, it means the task is waiting within VSAM for I/O to take place, or is waiting for a record lock.
- A wait on resource type FCRVWAIT occurs when conflicts over shared or exclusive locks are deferred internally by VSAM and not returned as an error condition to CICS. Conflicts which cause an FCRVWAIT wait are:
 - A task issues a file control READ UPDATE request for a record, for which:
 - Another task already holds an exclusive lock
 - One or more tasks hold a shared lock.
 - A task issues a file control READ request with CONSISTENT or REPEATABLE integrity for a record, for which:
 - Another task already holds an exclusive lock.
 - Another task is waiting for an exclusive lock because one or more tasks may already have a shared lock, or another task has an exclusive lock.
- Waiting on this resource can occur only for files accessed in RLS mode.

Client Scenario:

- Client called support center with FCRVWAIT on RLS file RLSTEST on LPAR MV20 CICS APPLID IYNX5
- Client does not know if they are waiting for I/O or if there is already an owner of the lock needed



Gathering Documentation for hangs

- Hang situations with CICS and RLS require dumps of the following spaces
 - All SMSVSAM address spaces and dataspace on the SysPlex
 - Master address spaces
 - All XCF address spaces and dataspace on the Sysplex
 - All CICS address spaces on the SysPlex
(On this SysPlex all CICS region names start with IYNX)
 - DUMP COMM=(CICS & SMSVSAM HANG)
 - R nn, JOBNAME=(SMSVSAM, XCFAS, *MASTER*, IYNX*), CONT
 - R nn, DSPNAME=('XCFAS'.*, 'SMSVSAM'.*), CONT
 - R nn, SDATA=(COUPLE, ALLNUC, LPA, LSQA, PSA, RGN, SQA, SUM, TRT, CSA, GRSQ, XESDATA), CONT
 - R nn, REMOTE=(SYSLIST=*('XCFAS', 'SMSVSAM', 'IYNX*', '*MASTER*'), DSPNAME, SDATA), END

NOTE: Refer to MVS System Commands for dump parameters

SYSLOG after COMM DUMP was issued on MV23

MV23 19:30:13.98 IEA045I AN SVC DUMP HAS STARTED AT TIME=19.30.13 DATE=01/27/2015 981
FOR ASIDS(0001,0009,0006,003C,003B,0040,0041,0043,0042,0047,0038,0044,003A)

MV3C 19:30:14.56 IEA045I AN SVC DUMP HAS STARTED AT TIME=19.30.14 DATE=01/27/2015 662
FOR ASIDS(0005,0009,0006,0001)

MV24 19:30:14.56 IEA045I AN SVC DUMP HAS STARTED AT TIME=19.30.14 DATE=01/27/2015 698
FOR ASIDS(0005,0009,0006,0001,003B)

MV21 19:30:14.56 IEA045I AN SVC DUMP HAS STARTED AT TIME=19.30.14 DATE=01/27/2015 102
FOR ASIDS(0005,0009,0006,0001)

MV20 19:30:14.56 IEA045I AN SVC DUMP HAS STARTED AT TIME=19.30.14 DATE=01/27/2015 867
FOR ASIDS(0005,0009,0006,0001,0041,012D,0148,0164)

SYSLOG after COMM DUMP was issued on MV23

MV24 19:30:29.14 IEA611I COMPLETE DUMP ON DUMP.MV24.DUMPSRV.D150127.T193013.S00188
DUMPID=188 REQUESTED BY JOB (DUMPSRV) FOR ASIDS(0005,0009,0006,0001,003B,0021)
REMOTE DUMP FOR SYSNAME: MV23
INCIDENT TOKEN: SYSPLEX1 MV23 01/27/2015 19:30:13

MV21 19:30:29.70 IEA611I COMPLETE DUMP ON DUMP.MV21.DUMPSRV.D150127.T193013.S00010
DUMPID=010 REQUESTED BY JOB (DUMPSRV) FOR ASIDS(0005,0009,0006,0001)
REMOTE DUMP FOR SYSNAME: MV23
INCIDENT TOKEN: SYSPLEX1 MV23 01/27/2015 19:30:13

MV20 19:30:33.04 IEA611I COMPLETE DUMP ON DUMP.MV20.DUMPSRV.D150127.T193013.S04122
DUMPID=122 REQUESTED BY JOB (DUMPSRV) FOR ASIDS(0005,0009,0006,0001,0041,012D,0148,0164,001E)
REMOTE DUMP FOR SYSNAME: MV23
INCIDENT TOKEN: SYSPLEX1 MV23 01/27/2015 19:30:13

MV3C 19:30:39.74 IEA611I COMPLETE DUMP ON DUMP.MV3C.DUMPSRV.D150127.T193013.S01062
DUMPID=062 REQUESTED BY JOB (DUMPSRV)
FOR ASIDS(0005,0009,0006,0001)
REMOTE DUMP FOR SYSNAME: MV23
INCIDENT TOKEN: SYSPLEX1 MV23 01/27/2015 19:30:13

MV23 19:30:50.31 IEA611I COMPLETE DUMP ON DUMP.MV23.#MASTER#.D150127.T193013.S00073
DUMPID=073 REQUESTED BY JOB (*MASTER*)
FOR ASIDS(0001,0009,0006,003C,003B,0040,0041,0043,0042,0047,0038,0044,
003A,001E)
REMOTE DUMPS REQUESTED
INCIDENT TOKEN: SYSPLEX1 MV23 01/27/2015 19:30:13

MV20 DUMP: ST SYS

SYSTEM STATUS:

Nucleus member name: IEANUC01

I/O configuration data:

IODF data set name: SYS1.IODF03

IODF configuration ID: PLX1

EDT ID: P1

Sysplex name: SYSPLEX1

TIME OF DAY CLOCK: CE6B7918 AC1D818B 01/27/2015 **19:30:15.721432 local**

TIME OF DAY CLOCK: CE6B7918 AC1D818B 01/27/2015 **19:30:15.721432 GMT**

Program Producing Dump: SVCDUMP

Program Requesting Dump: IEAVTSDT

Incident token: SYSPLEX1 MV23 01/27/2015 19:30:13.986860 GMT

Note time dump was taken

IYNX5 on MV20: VERBX DFHPD670 'DS=3'

DS_TOKEN	KE_TASK	T	S	F	P	RESOURCE TYPE	RESOURCE_NAME	W	TIME OF SUSPEND	TIMEOUT DUE	DTA (DSTSK)	AD ATTACHER TOKEN	MD	SUSPAREA	XM_TXN_TOKEN
01080005	1FB58800	S	S	N	N	IS_SCHED	IS_SCHDQ	S	18:46:24.684	-	1FAA7680	XM 21188B00	QR	1FA35A10	21188B0000000039C
01120003	31D48000	S	R								1FAA7E00	AP 00041900	CQ		
01840003	1FAD5630	S	S	N	N	SODOMAIN	SO_NOWORK	M	19:17:47.906	-	1FAF4380	XM 1FB09500	SL	1FB2E730	1FB095000000004C
01860003	1FAD5000	S	S	N	N	EPECQEMT	EPSUSPND	M	18:46:20.532	-	1FAF4500	XM 1FB09700	EP	1FB2C078	1FB0970000000005C
018E0003	1FB58100	S	S	N	N	RRMSEXIT	RESYNC	M	18:46:19.468	-	1FAF4B00	1FEB2BC0	QR	1FB57100	
02060003	211DA100	S	S	N	N	FCCFQS		M	19:26:23.544	19:31:23.544	31D40500	XM 1FB09100	QR	211C2290	1FB0910000000024C
020C0003	21191800	S	S	N	N	FCCFQR		M	18:46:21.237	-	31D40980	XM 1FB0A500	QR	211BD828	1FB0A50000000025C
020E0003	211CA100	S	S	N	N	KCCOMPAT	SINGLE	W	18:46:24.515	-	31D40B00	XM 1FB0A700	QR	000478B8	1FB0A70000000030C
02120001	1F928000	S	S	N	N	SMSYSTEM		S	19:26:20.922	19:31:20.922	31D40E00	SM 00000002	QR	1FA35C20	
02800003	211B2800	S	S	N	N	CSNC	MROQUEUE	M	18:46:23.141	-	31DAA080	XM 1FB0AB00	QR	2155503C	1FB0AB0000000026C
0D000001	21131100	N	S	P	N	ICWAIT		S	19:30:13.706	-	31D51080	XM 1FB0AD00	QR	31D51080	1FB0AD0000000029C
0D020015	210FC100	N	S	N	N	FCRVWAIT	RLSTEST	M	19:28:12.423	-	31D51200	XM 1FB09D00	QR	211C22F0	1FB09D0000000059C
0D040009	210FC800	N	S	P	N	ZCLOWAIT	DFHZARQ1	S	19:28:14.764	-	31D51380	XM 1FB09300	QR	31D51380	1FB0930000000054C

Wait Type

File Name

Timeout Due

ECB

Task Number

FCRVWAIT – What we know so far

- Task Number 00059 issued a File Control Request on CICS Region IYNX5 which is running on MVS LPAR MV20
- Task Number 00059 is in an FCRVWAIT against file RLSTEST and does not have DTIMOUT set
- Task Number 00059 is waiting for ECB 211C22F0 to be posted

IYNX5 on MV20: VERBX DFHPD670 'FCP'

==FCP: CURRENT FILE ACTIVITY

Key for Current File Activity :

RPL : RPL address
Task : Owing task number
Type : Type of command = BROWSE,DELETE,READ,WRITE,READ_U(Read for Update)
RPLOPT : RPL option codes (first 3 bytes)
VSWARESP: RPL feedback word
Hold : Holding position - YES or NO
I/F : VSAM request is currently active - YES or NO
X/C : Request failed with exclusive control conflict - YES or NO
Wait : Tasks are in exclusive control conflict waiting on this RPL - YES or blank
XCRPL : RPL this activity is waiting on in exclusive control conflict
XCTSK : Task number this activity is waiting on in exclusive control conflict
Request : Vsam Request Byte = GET,PUT,CHECK,POINT,END,ERASE,WRTBUF(Write Buffer)
RIDFLD : Key, RBA or RRN

Filename	RPL	Task	Tran	Type	RPLOPT	VSWARESP	Hold	I/F	X/C	Wait	XCRPL	XCTSK	Request	RIDFLD
RLSTEST	211C22E8	00059	AFCV	READ_U	408680	010800B9	YES	YES	NO	NO	-	-	GET	C1C1C1C1
---- FILE REQUESTS TABLE END ----														

IYNX5 on MV20: VERBX DFHPD670 'FCP'

FIND RLSTEST

==FCP: FILE CONTROL TABLE ENTRIES

Key for FCTE summary table :

ACC : Access type, VSAM or BDAM
 TYPE : File type, KSDS, ESDS, RRDS, VRRDS
 MODE : File mode, PATH, AIX=Alternate index, BASE
 LSR : LSR pool num
 REM : Remote file ?
 SLG : Use system log ?
 SREQS : Servreq settings, R=Read, U=Update,
 A=Add, D=Delete, B=Browse,
 STATUS : File status, OPEN=OPEN, CLOS=CLOSE,
 OING=OPENING, CING=CLOSING,
 ENA=ENABLED, DIS=DISABLED, UNE=UNENABLED
 JID : Journal ID
 DSNB-OJB : Address of the object DSNB
 DSNB-BAS : Address of the base DSNB
 FR : Forward recovery ?
 FRL : Forward recovery log ID
 TIME OPEN : Time file opened (store clock value)

ADDRESS	FILENAME	ACC	TYPE	MODE	RLS	LSR	REM	SLG	SREQS	STATUS	JID	DSNB-OBJ	DSNB-BAS	FR	FRL	TIME OPEN	
211D1030	IYNX5	VSAM			NO	1	NO	YES	RUADB	CLOS	ENA	0	211E54D0	00000000	NO	0	
211D1248	IYNX5RLS	VSAM			YES	1	NO	NO	RUADB	CLOS	ENA	0	211E55F8	00000000	NO	0	
211D1AA8	REMLRS	VSAM			NO	1	YES	NO	R	CLOS	ENA	0	00000000	00000000	NO	0	
211D1CC0	RLSBASE	VSAM			YES	1	NO	NO	RUADB	CLOS	ENA	0	211E5848	00000000	NO	0	
21545030	RLSFILE	VSAM			YES	1	NO	NO	RUADB	CLOS	ENA	0	211E55F8	00000000	NO	0	
21545248	RLSPATH	VSAM			YES	1	NO	NO	RUADB	CLOS	ENA	0	211E5970	00000000	NO	0	
21545460	RLSREADO	VSAM			NO	1	NO	NO	R B	CLOS	ENA	0	211E55F8	00000000	NO	0	
21545678	RLSTEST	VSAM	KSDS	BASE	YES	1	NO	NO	RUADB	OPEN	ENA	0	211E5A98	211E5A98	NO	0	27/01/15 18:46:51

IYNX5 on MV20: VERBX DFHPD670 'FCP'

FIND 211E5A98

==FCP: DATASET NAME BLOCKS

Key for DSNB summary table :

TYP : File type as set in DSNB
VVC : Validated against VSAM catalog ?
FR : Forward recovery ?
LOG : Backout logging ?
RAI : Recovery attributes initialised flag
KLN : Key length
KPO : Key position
FRL-TKN : Forward recovery log token
LSR : LSR pool num
ACBO : Number of ACBs open
ACBOU : Number of ACBs open for update
AFC : Attributes originate from VSAM catalog ?
ARO : Attributes set on RLS open ?
BRO : **BCB has RLS ACBs open ?**
BNO : BCB has non RLS ACBs open ?

ADDRESS	DATASET-NAME	TYP	VVC	FR	LOG	RAI	KLN	KPOS	FRL-TKN	LSR	ACBO	ACBOU	AFC	ARO	BRO	BNO	RCO
211E5280	USASSC1.BIGG.KSDS		NO	NO	NO	OFF	0	0	00000000	0	0	0	NO	NO	NO	NO	NO
211E5158	USASSC1.CICSR670.BASE		NO	NO	NO	OFF	0	0	00000000	0	0	0	NO	NO	NO	NO	NO
211E53A8	USASSC1.CICSR670.ESDS		NO	NO	NO	OFF	0	0	00000000	0	0	0	NO	NO	NO	NO	NO
211E5720	USASSC1.CICSR670.PATH		NO	NO	NO	OFF	0	0	00000000	0	0	0	NO	NO	NO	NO	NO
211E5848	USASSC1.CICSR680.RLSBASE		NO	NO	NO	OFF	0	0	00000000	0	0	0	NO	NO	NO	NO	NO
211E5970	USASSC1.CICSR680.RLSPATH		NO	NO	NO	OFF	0	0	00000000	0	0	0	NO	NO	NO	NO	NO
211E5A98	USASSC1.IYNXVRLS.FILE	K	YES	NO	NO	ON	4	0	00000000	0	1	1	YES	YES	YES	NO	NO
211E5030	USASSC1.IYNX5.DFHCS	K	YES	NO	NO	OFF	22	0	00000000	0	0	0	NO	NO	NO	NO	NO
211E54D0	USASSC1.IYNX5.KSDS		NO	NO	NO	OFF	0	0	00000000	0	0	0	NO	NO	NO	NO	NO
211E55F8	USASSC1.IYNX5RLS.FILE		NO	NO	NO	OFF	0	0	00000000	0	0	0	NO	NO	NO	NO	NO
211E5BC0	USASSC5.CICSR420.RRDS		NO	NO	NO	OFF	0	0	00000000	0	0	0	NO	NO	NO	NO	NO

FCRVWAIT – What we know so far

- Task Number 00059 issued a Read Update File Control Request using RPL 211C22E8 KEY C1C1C1C1 on CICS Region IYNX5 using which is running on MVS LPAR MV20
- Task Number 00059 is in an FCRVWAIT against file RLSTEST Dataset Name USASSC1.IYNXV.RLSFILE and does not have DTIMOUT set
- Task Number 00059 is waiting for ECB 211C22F0 to be posted

RLS side of FCVRWAIT

- Delays in SMSVSAM can be from a variety of sources:
 - Cross-system contention on lock (lock contention)
 - Same-system contention on lock (lock contention)
 - Waiting on split (split lock contention)
 - Buffering contention, or various other internal latch issues
- Two ways to approach this:
 - IDCAMS SHCDS commands
 - Dump and use IPCS diagnostic tools

Using SHCDS

- IDCAMS LISTSUBSYS(ALL)

SHCDS LISTSUBSYS(ALL)

----- LISTING FROM SHCDS ----- IDC0003

SUBSYSTEM NAME	STATUS	RECOVERY NEEDED	LOCKS HELD	LOCKS WAITING	LOCKS RETAINED
IYNXV	ONLINE--ACTIVE	NO		1	0
DATA SETS IN LOST LOCKS-----			0		
DATA SETS IN NON-RLS UPDATE STATE--			0		
TRANSACTION COUNT-----			1		
IYNX5	ONLINE--ACTIVE	NO		0	1
DATA SETS IN LOST LOCKS-----			0		
DATA SETS IN NON-RLS UPDATE STATE--			0		
TRANSACTION COUNT-----			1		
SMSVSAM	BATCH---ACTIVE	NO		0	0
DATA SETS IN LOST LOCKS-----			0		
DATA SETS IN NON-RLS UPDATE STATE--			0		
TRANSACTION COUNT-----			0		
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0					

Using IGWFPMAN

- **VERBX IGWFPMAN 'F(SMLS) JOBNAME(SMSVSAM)'**
- Quick overview of output:
 - SMLS = System Managed Locking Service
 - Arranged in Sections, repeats same information:
 - **Global Wait Q** – “Global Wait Queue Follows”
 - List of all lock waiters in the system
 - Used for deadlock detection
 - **RHT** – Resource Hash Table – “RHT Display”
 - List of all resources, holders, and waiters
 - Used to quickly reference locks

Back to SHCDS

- IDCAMS LISTSUBSYS(ALL)

SHCDS LISTSUBSYS(ALL)

----- LISTING FROM SHCDS ----- IDC0SH03

SUBSYSTEM NAME	STATUS	RECOVERY NEEDED	LOCKS HELD	LOCKS WAITING	LOCKS RETAINED
IYNXV	ONLINE--ACTIVE	NO		1	0
DATA SETS IN LOST LOCKS-----			0		
DATA SETS IN NON-RLS UPDATE STATE--			0		
TRANSACTION COUNT-----			1		
IYNX5	ONLINE--ACTIVE	NO	0	1	0
DATA SETS IN LOST LOCKS-----			0		
DATA SETS IN NON-RLS UPDATE STATE--			0		
TRANSACTION COUNT-----			1		
SMSVSAM	BATCH---ACTIVE	NO	0	0	0
DATA SETS IN LOST LOCKS-----			0		
DATA SETS IN NON-RLS UPDATE STATE--			0		
TRANSACTION COUNT-----			0		

IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

FCVRWAIT - What we know so far

- Task Number 00059 issued a Read Update File Control Request using RPL 211C22E8 KEY C1C1C1C1 on CICS Region IYNX5 using which is running on MVS LPAR MV20
- Task Number 00059 is in an FCRVWAIT against file RLSTEST Dataset Name USASSC1.IYNXV.RLSFILE and does not have DTIMOUT set
- Task Number 00059 is waiting for ECB 211C22F0 to be posted
- IYNX5 on MV20 is waiting for a lock on KEY=AAAA
- IYNXV on MV23 is holding the lock

IYNXV on MV23: VERBX DFHPD680 'FCP'

==FCP: CURRENT FILE ACTIVITY

Key for Current File Activity :

RPL : RPL address
Task : Owing task number
Type : Type of command = BROWSE,DELETE,READ,WRITE,READ_U(Read for Update)
RPLOPT : RPL option codes (first 3 bytes)
VSWARESP: RPL feedback word
Hold : Holding position - YES or NO
I/F : VSAM request is currently active - YES or NO
X/C : Request failed with exclusive control conflict - YES or NO
Wait : Tasks are in exclusive control conflict waiting on this RPL - YES or blank
XCRPL : RPL this activity is waiting on in exclusive control conflict
XCTSK : Task number this activity is waiting on in exclusive control conflict
Request : Vsam Request Byte = GET,PUT,CHECK,POINT,END,ERASE,WRTBUF(Write Buffer)
RIDFLD : Key, RBA or RRN

Filename	RPL	Task	Tran	Type	RPLOPT	VSWARESP	Hold	I/F	X/C	Wait	XCRPL	XCTSK	Request	RIDFLD
RLSTEST	17ED3C88	00493	AFCV	READ_U	408680	00000000	YES	NO	NO	NO	-	-	GET	C1C1C1C1
---- FILE REQUESTS TABLE END ----														

IYNXV on MV23: VERBX DFHPD680 'DS=3'

DS_TOKEN	KE_TASK	T	S	F	P	RESOURCE TYPE	RESOURCE_NAME	TIME OF SUSPEND	TIMEOUT DUE	DTA (DSTSK)	AD ATTACHER TOKEN	MD	SUSPAREA	XM_TXN_TOKEN
02120001	28740680	S	S	N	N	IN	SMSYSTEM	19:25:43.960	19:30:43.960	28748E00	SM 00000002	QR	1643DA40	
02800001	1651E800	S	S	N	N	-	RRMSEXIT RESYNC	21:21:11.329	-	28795080	167F3FC4	QR	1652A100	
02840001	16323680	S	S	N	N	-	TIEXPIRY DS_NUDGE	19:26:23.281	-	28795380	TI 00610003	QR	1643D980	
02860001	16332000	S	R							28795500	AP 00041800	CQ		
02880003	16482680	S	S	N	N	-	IS_INPUT IS_PROCQ	21:21:14.078	-	28795680	XM 1650AD00	QR	1643DE60	1650AD0000000033C
028A0003	17AE2100	S	S	N	N	-	IS_ERROR IS_ERROQ	21:21:14.054	-	28795800	XM 17B2E100	QR	1643DC80	17B2E10000000034C
028C0003	16532100	S	S	N	N	-	IS_SCHED IS_SCHDQ	21:21:14.070	-	28795980	XM 17B2E900	QR	1643DCB0	17B2E90000000035C
028E0003	17AFD100	S	S	N	N	IN		19:29:45.919	19:30:45.919	28795B00	XM 17B2E700	QR	17EAF210	17B2E70000000036C
0C8A029F	179FB800	N	S	P	N	-	ICWAIT TC20	19:28:11.215	-	291E9800	XM 1650AB00	QR	291E9800	1650AB0000000493C

Wait Type

Time of Suspend

Task Number

FCVRWAIT - What we know

- Task Number 00059 issued a Read Update File Control Request using RPL 211C22E8 KEY C1C1C1C1 on CICS Region IYNX5 using which is running on MVS LPAR MV20
- Task Number 00059 is in an FCRVWAIT against file RLSTEST Dataset Name USASSC1.IYNXV.RLSFILE and does not have DTIMOUT set
- Task Number 00059 is waiting for ECB 211C22F0 to be posted
- IYNX5 on MV20 is waiting for a lock on KEY=AAAA
- IYNXV on MV23 is holding the lock
- Task 00493 on MV23 CICS Region IYNXV issued a Read Update File Control Request using RPL 17ED3CC8 KEY C1C1C1C1 and is currently in an ICWAIT (Interval Control)

Steps to avoid FCRVWAIT

- Code DFHSIT parameter FTIMEOUT to a sensible value
- Code DTIMOUT value in the Transaction Definition
 - DTIMOUT takes precedence over FTIMEOUT
- Recode program to not issue Read Update followed by Delay
- If FTIMEOUT or DTIMOUT were set properly, the task waiting would have received an AFCV Abend

DFHFC0164 27/01/2015 20:53:09 IYNX5 AFCV 00051 TC72 CICSUSER. A request has timed out waiting for an RLS lock. There are 1 transactions or Transactional VSAM units of recovery holding this lock.

DFHFC0165 27/01/2015 20:53:09 IYNX5 AFCV 00051 TC72 CICSUSER. Transaction AFCV (00501) unit of work X'CE6B8B7E5513148D' running in job IYNXV with applid IYNXV in MVS MV23 holds exclusive lock on key X'C1C1C1C1' in data set USASSC1.IYNXVRLS.FILE causing true contention.

DFHDU0203I 27/01/2015 20:53:09 IYNX5 A transaction dump was taken for dumpcode: AFCV, Dumpid: 1/0001.

DFHAC2236 27/01/2015 20:53:09 IYNX5 Transaction AFCV abend AFCV in program AFCVIT term TC72. Updates to local recoverable resources will be backed out.

References

- SMSVSAM:
 - Redbook: *VSAM Demystified*, Chapter 5 (SG24-6105)
 - *DFSMSdfp Storage Administration Reference* (SC26-7402)
 - *DFSMS: Using Data Sets* (SC26-7410)
- CICS
 - Redbook: *CICS and VSAM Record Level Sharing: Implementation Guide* (SG24-4766)
 - CICS Transaction Server Documentation -> Installing -> Setting up the MVS environment for CICS -> Setting up VSAM RLS support (<https://ibm.biz/BdE5vN>)

Trademarks

DFSMSdfp, DFSMSdss, DFSMSshsm, DFSMSrmm, IBM, IMS, MVS, MVS/DFP, MVS/ESA, MVS/SP, MVS/XA, OS/390, SANergy, and SP are trademarks of International Business Machines Corporation in the United States, other countries, or both.

AIX, CICS, DB2, DFSMS/MVS, Parallel Sysplex, OS/390, S/390, Seascope, and z/OS are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Domino, Lotus, Lotus Notes, Notes, and SmartSuite are trademarks or registered trademarks of Lotus

Development Corporation. Tivoli, TME, Tivoli Enterprise are trademarks of Tivoli Systems Inc. in the United States and/or other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

Other company, product, and service names may be trademarks or service marks of others.

Notices & Disclaimers

Copyright © 2015 by International Business Machines Corporation.

No part of this document may be reproduced or transmitted in any form without written permission from IBM Corporation.

Permission is granted to SHARE Inc. to publish this presentation paper in the SHARE Inc. proceedings; IBM retains the right to distribute copies of this presentation to whomever it chooses.

Product information and data has been reviewed for accuracy as of the date of initial publication. Product information and data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or programs(s) described herein at any time without notice.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such such products, programs or services available in all countries in which IBM operates or does business. Consult your local IBM representative or IBM Business Partner for information about the product and services available in your area.

Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectually property rights, may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS"WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein.

Notices & Disclaimers

The performance data contained herein was obtained in a controlled, isolated environment. Actual results that may be obtained in other operating environments may vary significantly. While IBM has reviewed each item for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere.

The responsibility for use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's or user's ability to evaluate and integrate them into their operating environment. Customers or users attempting to adapt these techniques to their own environments do so at their own risk. **IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY.**

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not necessarily tested those products in connection with this publication and cannot confirm the accuracy of 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.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.



Getting the Most out of your VSAM Data Sets in CICS by Using RLS

Andre Clark – CICS L2 Support

Neal Bohling – RLS Development

Session 17257



SHARE is an independent volunteer-run information technology association
that provides **education, professional networking and industry influence.**

