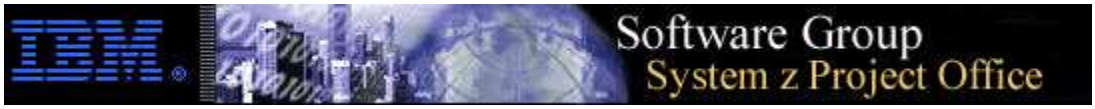
w3 COMP IBM.

Kettner's Corner #11 - zTidBitz@CPO (VSAM)



John Kettner is a member of the Z CPO team and teaches classes on z/OS fundamentals to customers and IBMers. John will share his expertise sharing his answers on key questions and topics on Z.

Check out the System z Project office Website: http://w3.ibm.com/sales/competition/compdlib.nsf/weball/9CF22F2F65E01639872571 OpenDocument

06/01/07 - #11 zTidBit@CPO (VSAM RLS)

Technical Rating: ***** (HIGH): 6+ years mainframe**
for this doc. ** (MEDIUM): 2- 6 years mainframe
(see **BOLD**) * (LOW): 1 - 2 year(s) mainframe

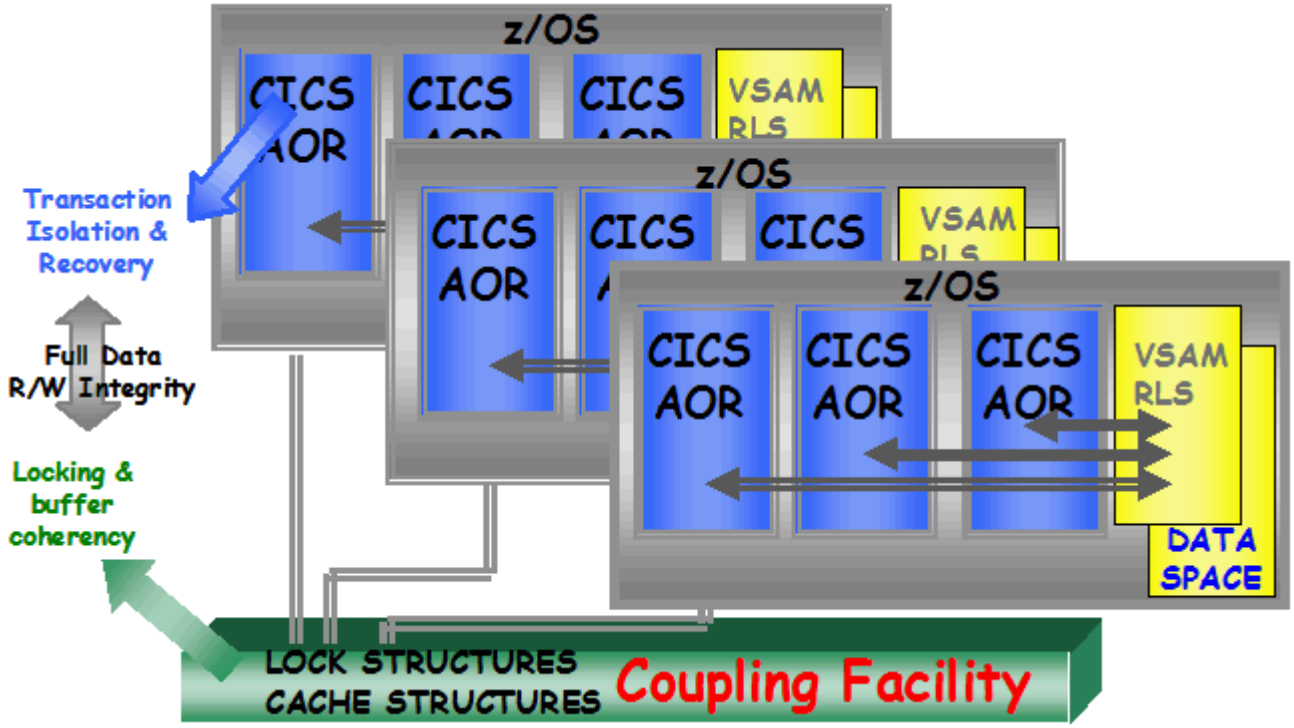
I recently did a SWGCPO teleconference on Virtual Storage Access Method (VSAM) and I want to re-emphasize an aspect of z/OS. VSAM is the most integrated access method within z/OS and serves as the foundation for many program products and database managers. Our customers use VSAM as a major data depository for thousands of terabytes to hold business information. One of the *best* features of this access method is record level sharing **(RLS)**. Recall that z/OS is a SHARE-EVERYTHING architecture and this overview is one highlight using its design. RLS allows concurrent access in a sysplex (s) to VSAM data sets at the record level allowing *any number* of users to share VSAM file structures < also known as

Spheres >. This implementation requires no change to applications.
Note – The major difference between RLS and non-RLS mode is where z/OS places the VSAM Control Block Structures.

Customer data exist in VSAM files accessed by thousands of applications. Once of the most popular uses of VSAM is within CICS. This large volume of data and its application foundation generated the motivation for VSAM RLS. In order to exploit the all inclusive value of this functionality across CICS systems is the use of a Coupling Facility (CF) which is the implementation for data sharing. The transaction-execution model enables the integrity for data sharing. Therefore, three data managers that execute on z/OS and provides transactional recovery for their data are: Information Management System Database (IMS-DB), DATABASE2 (DB2) and CICS VSAM.

NOTE: RLS itself does not provide transactional recovery. But, when used through CICS, applications < then > have the transactional recovery functions for VSAM files. CICS infrastructure provides logging and recovery. RLS provides locking and sysplex-wide concurrent shared-data access.

In the past, VSAM requests were executed within the application caller's address space. Now, the implementation of RLS is to use a continually running 'server' address space. RLS requests submitted from any address space on a z/OS image (LPAR) are handled by this server using a shared buffer pool resident within a data space (special form of address space). RLS logic is implemented by programs running in a VSAM RLS Server and provides the cross-memory access from the caller's address space to the server's address space.



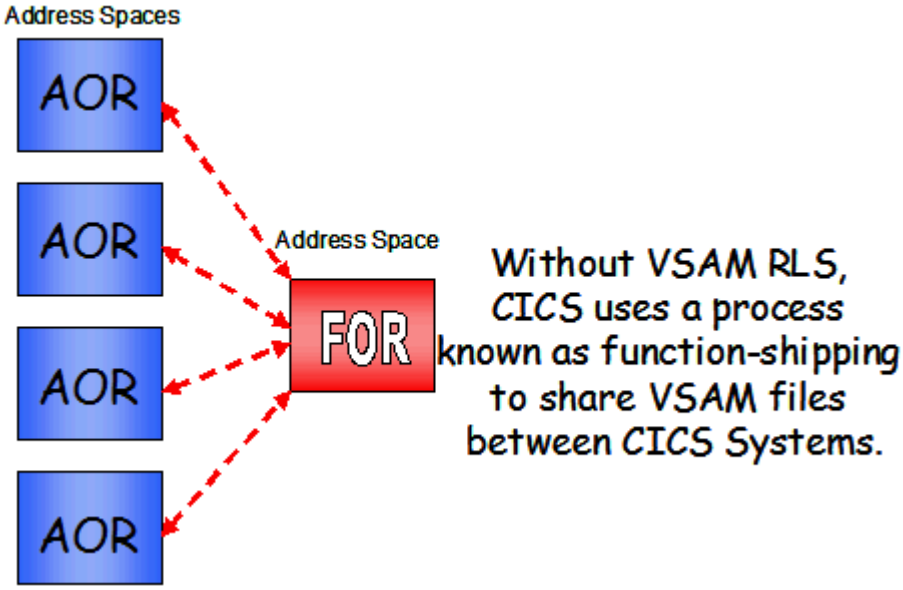
Part two of the RLS implementation is in order to meet the availability requirements of a shared server, functional recovery was placed in all elements of RLS processing. Cancellation or failure of any thread (task) under which an RLS request is executing results in the functional recovery performing clean up to avoid interference with the execution of other RLS requests under other threads, tasks or address spaces*.

One instance of RLS server is resident in each z/OS within the sysplex. The multiple instances of the RLS server are aware of the existence of one another through the use of a shared control dataset, the cross-system coupling facility (XCF) messaging and access to the coupling facility lock and cache structures.

Alternatives of not using RLS is using either NSR (non-shared resources) or LSR (local shared resources) access modes. But, this provides serialization at the control interval (CI) level. Having the access at a control interval you will lock up many records < depending on the CI size > even through

you likely need only one. These alternatives will likely cause more lock contention and throughput delays.These access modes existed prior to RLS and continue to exist. This non-RLS function is provided by a local buffer manager. The buffer manager permits only one valid copy of a control interval to reside within the bufferpool. By obtaining exclusive use of a buffer, the owner has exclusive use of the control interval resident within the buffer. Therefore, there are now three choices: NSR, LSR and RLS.

As a side note without VSAM RLS, CICS uses a process known as function-shipping to share VSAM file structures between CICS Systems. Function shipping is implemented by using a multi-region operation (MRO) environment. Files resided in a completely separate address space in a file-owning region (FOR) and served as the focal point for all VSAM access. When a CICS transaction < business logic > executes in an Application Owning Region (AOR) the request for data is then shipped off to the FOR while the AOR awaits a reply. This design implementation is very expensive in resource consumption. It can also incur a performance bottleneck as well as potentially a single point of failure.



In summary - RLS uses z/OS' coupling

technology to provide record level data sharing. Although RLS provides locking and buffer coherency to achieve data integrity for shared data, it does not provide the application level or transaction level isolation of backout recovery. The transactional recovery functions are provided by CICS which has been extended to support RLS. Therefore, CICS and VSAM RLS provide the benefits of z/OS' Coupling Facility to CICS VSAM Applications.

*As honorary mention, there is an extension use of RLS within DFSMStvs (Transactional VSAM) where VSAM will provide locking, two-phase commit, and logging facilities that allow multiple batch update jobs to run concurrently with CICS access to the same data sets while maintaining integrity and recoverability.

If you have question for John, send him an email at: John H Kettner/New York/IBM.

For a direct link to the Z Project Office Website Click Here:http://w3.ibm.com/sales/competition/compdlib.nsf/weball/9CF22F2F65E0163987OpenDocument

---

Document details

Date: Jun. 1, 2007

Content owner: John H Kettner

For IBMers only. Not intended for business partners or customers.

Report handling guidelines

Content

provider:
IBM SWG Competitive Project Office

Feedback

Tag and save this page to your dogear favorites.

What is dogear ?