Sharing the Wealth: VSAM Record-Level Sharing

BY STEVE PRYOR

Not long ago, it seemed that the future of the large mainframe environment was in doubt. Legions of smaller, less expensive, distributed computers would replace complex, costly mainframes, along with the expensive systems programming talent required to maintain them. Like most doomsday scenarios, however, this one has failed to materialize. Despite explosive growth in the number and power of smaller systems, the bulk of vital corporate data, particularly for larger companies, remains on the mainframe. Indeed, faced with what has turned out to be the very high cost of installing, maintaining, and controlling widely distributed hardware and software, corporations have begun reemphasizing the role of the mainframe.

One reason for this move back to a centralized environment is the maturity and functionality available in the large systems environment. While distributed systems still struggle with issues of backup, recovery, and accessibility to data, each new release of DFSMS provides new features that extend the already robust reliability, availability, and security that is the hallmark of the mainframe environment. For the storage administrator, each new release of DFSMS also brings increased capacity and complexity that must be effectively managed.

It has been estimated that more than half of all data stored on mainframes is VSAM. Accordingly, recent operating system releases have concentrated on improving the performance of VSAM files and making access to VSAM data easier and more reliable. This month's column will discuss some of the new features of DFSMS/MVS that relate to VSAM data and how these features impact the storage administrator's job.

VSAM RLS

For many VSAM-based applications, sharing data across multiple systems has been a difficult problem. Applications that need to maintain integrity while allowing multiple systems to update VSAM files have had to resort to complex enqueueing schemes to prevent data from being corrupted, since there has not been any ability within MVS to provide cross-system serialization at the record level. Until now, that is.

DFSMS/MVS version 1.3 introduced VSAM Record-Level Sharing (RLS), which uses the coupling facility hardware to allow VSAM data to be accessed for update from multiple systems. The coupling facility is used as a store-through cache where records accessed for update can be stored and where locks can be maintained to ensure integrity.

VSAM RLS itself is merely an access method that can be implemented by a JCL parameter (RLS=) or in the ACB (MACRF=RLS). To take full advantage of VSAM RLS, the application (usually CICS) must provide control of the logical units of work that perform multiple concurrent updates and provide logging facilities to allow recovery processing. For VSAM RLS data, the SHAREOPTION values are ignored, and many different tasks, running in many different regions, can update the same data without danger of corruption, since data is locked at the record level, not at the control-interval level.

VSAM RLS is subject to a number of restrictions, as is the case with most new MVS features. For example, RLS datasets must be SMS-managed. VSAM RLS datasets may be in key-sequenced, entry-sequenced, or relative-record format. Clusters accessed with RLS may not be defined with IMBED (although REPLICATE is allowed) and may not use control interval access. For KSDS clusters, addressed access (RPL OPTCD=ADR) is also not allowed.

VSAM RLS is implemented by a new SMSVSAM address space that must be started on each system which takes part in the sysplex. The SMSVSAM address spaces contain the VSAM control blocks and coordinate the sharing of VSAM RLS data across systems. Each SMSVSAM address space is also linked to two data spaces, one that is used for data buffering, and the other for activity monitoring and writing SMF data. For RLS files, the ACB buffering options, such as BUFND, BUFNI, BUFSP, and STRNO are ignored, as are requests for control blocks in common (CBIC) and user buffering (UBF). Instead, all of the buffering is performed in the SMSVSAM address space and its associated data spaces. Only the ACB, RPLs, and EXLSTs are left in the user address space.

Along with its new possibilities for applications, VSAM RLS brings with it a number of complexities for the storage administrator. Since each VSAM RLS dataset must be SMS-managed, the storage administrator must ensure that the ACS routines assign appropriate storage classes. The SMS base configuration definition and CFRM (coupling facility) policy may need to be changed to specify CF cache structures. Storage classes that are to be assigned to VSAM RLS datasets must contain the CF cache structure names. A CF lock structure with the fixed name of IGWLOCK00 must also be defined to support VSAM RLS.

In addition to changes to the SMS base configuration and storage class definitions, changes may need to be made to JCL and IDCAMS control statements or SMS data classes. VSAM datasets that are to be eligible for RLS will need to be assigned LOG ids and LOGSTREAM attributes that indicate whether they are recoverable and whether forward recovery logging should be performed. This can be done either through the IDCAMS DEFINE and ALTER control statements, or for DFSMS 1.4

(OS/390 version 2.4) through the equivalent SMS data class attributes.

IMPLEMENTING VSAM RECORD-LEVEL SHARING

To implement VSAM record-level sharing, Sharing Control Data Sets (SHCDSs) must be created. These datasets are used to hold record locking information and prevent data from being corrupted during a failure of the SMSVSAM address space or coupling facility lock structure. At least two SHCDSs must be defined along with at least one spare, and they must be defined with exactly the same space allocation characteristics. The SHCDSs have a name of the form SYS1.DFPSHCDS.qualifier.Vvolser, and thus are placed on specific shared-DASD volumes (preferably ones without any RESERVE activity) via a guaranteed-space storage class.

The SHCDSs can be manipulated (activated, deactivated, added, deleted) via new parameters on the DISPLAY SMS and VARY SMS console commands. DISPLAY SMS, SHCDS, for example, will display the status of all SHCDSs. The storage administrator must also be familiar with other functions of the to the VARY SMS commands, such as VARY SMS,SMSVSAM which can be used to start or stop the SMSVSAM address space and control its connectivity to the coupling facility.

New parameters in the IGDSMSxx member of SYS1.PARMLIB are also of concern to the storage administrator. The DEADLOCK_DETECTION interval determines how long a transaction may wait before the system decides a lock-out is present and terminates it. The RLS_MAX_POOL_SIZE sets the size of the buffer pools on a system.

DFSMS/MVS continues to expand in functionality and complexity. DFSMS version 1.4, distributed with OS/390 version 2.4, extends the reach of VSAM record-level sharing to extended-format (striped) VSAM datasets that are more than 4GB in size. With DFSMS 1.4, DFSMShsm control datasets may also take advantage of VSAM RLS. A number of other extensions to VSAM processing are included in DFSMS 1.4, including the ability for VSAM datasets to occupy a total of up to 255 extents (subject to a limit of 123 extents per component on a single volume). Many other significant changes to VSAM processing are also present in DFSMS 1.4.

New releases of DFSMS/MVS are now appearing at the rate of one every six months. With each new release, the ability of mainframe systems to handle ever larger volumes of data with greater ease, speed, and reliability improves. Along with this improvement comes increased complexity. To succeed in the challenging modern business environment, corporations also need highly knowledgeable storage administrators who are as effective and reliable as the systems they manage. ts

NaSPA member Steve Pryor has more than 15 years of experience in storage management, disaster recovery, software development, and technical support. Steve can be contacted via the Internet at pryor@atlanta.com.