Rule DAS625: NSR was used, but a large percent of the access was direct

Finding: CPExpert noticed that Non-shared resources (NSR) was used as the access method for the VSAM data sets listed, but most of the accesses were direct. This finding applies only if SMF Type 42 (Data Set Statistics) information is available[1] in a MXG performance data base

Impact: This can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of applications referencing the VSAM data. The level of impact depends on the number of direct I/O operations.

Discussion: I/O buffers are used by VSAM to read and write control intervals from DASD to virtual storage. Two buffering techniques are used with VSAM data sets that are accessed only on a locaL[2] system: Non-shared resource (NSR) and Local shared resource (LSR).

- **Non-Shared Resource (NSR)**. NSR is the default VSAM buffering technique. With NSR, VSAM buffers are not shared among VSAM data sets, and the buffers are located in the private area. VSAM data sets with NSR buffering can be accessed sequentially or direct (or both). However, NSR is suited for sequential processing because, if the data set access is sequential, the buffers are managed with a read-ahead algorithm. The read-ahead algorithm provides overlap of I/O and CPU processing and is efficient for sequential accesses. Since NSR is oriented toward sequential access, there is no expectation that a record will be re-used (as might exist with direct processing). Consequently, once a record is processed from the NSR buffers, the buffer is likely to be reclaimed for another record read from DASD.

- **Local Shared Resource (LSR)**. With LSR, VSAM buffers normally are shared among VSAM data sets accessed by tasks in the same address space. Since LSR is oriented toward shared (and direct) access, there is an expectation that a record might be re-used. Consequently, buffer management algorithms retain buffers as long as possible, using a least-recently used (LRU) algorithm, after a record is processed from the LSR buffers. There is no read-ahead algorithm with LSR, and there is no inherent overlap of I/O and CPU processing. LSR is appropriate for

[1] **%LET TYPE42DS = Y;** must be specified in USOURCE(GENGUIDE) must be specified in USOURCE(GENGUIDE) or USOURCE(DASGUIDE) to advise CPExpert that TYPE42DS is available.

[2] Two other techniques can be used: global shared resource (GSR) and record level sharing (RLS). GSR provides serialization of shared resources across multiple systems. VSAM RLS provides multisystem sharing of VSAM data sets in a parallel sysplex.

direct access of VSAM data sets, regardless of whether the data sets are shared.

As described above, NSR is best used for applications that use sequential or skip sequential as their primary access mode. NSR is not suited for direct processing, although NSR often **is** used for direct processing because it is easy to use and is the default buffering technique. Nonetheless, performance can be significantly improved if LSR is used for direct processing of VSAM data sets.

After applying the screening criteria specified for VSAM data sets, and extracting SMF Type 64 information for those VSAM data sets, CPExpert examines SMF Type 42 (Data Set Statistics) information for the selected VSAM data sets. CPExpert uses the TYPE42DS information to compute the percent of direct accesses to the VSAM data set, using the following algorithm:

$$Percent\ direct\ accesses = \frac{S42AMDRB}{S42AMSRB + S42AMDRB}$$

where: S42AMSRB = Blocks read using sequential access
S42AMDRB = Blocks read using direct access

CPExpert produces Rule DAS625 under the following conditions:

- The TYPE42DS S42DSBUF variable showed that NSR was used for KSDS or VRRDS VSAM data sets, and
- The percent of direct accesses for the data component was greater than the value specified for the NSRDIR guidance variable in USOURCE(DASGUIDE.

The default value for the NSRDIR guidance variable is 75%, so CPExpert will produce Rule DAS625 when NSR was specified as the buffering technique, and more than 75% of the accesses were direct for the data component.

The following example illustrates the output from Rule DAS625:


```
RULE DAS625:  NSR WAS USED, BUT LARGE PERCENT OF ACCESS WAS DIRECT

   VOLSER: MVS902.  Non-Shared resources (NSR) was specified as the
   buffering technique for the below VSAM data sets, but more than 75%
   of the I/O activity was direct access.  NSR is not designed for direct
   access, and many of the advantages of NSR are not available for direct
   access. You should consider Local Shared Resources (LSR) for the below
   VSAM data sets (perhaps using System Managed Buffers to facilitate the
   use of LSR).  The I/O RATE is for the time the data set was open. The
   SMF TIME STAMP and JOB NAME are from the last record for the data set.
                                                                ..       I/O     OPEN    -ACCESS TYPE (PCT)-
   SMF TIME STAMP    JOB NAME  VSAM DATA SET                   ..       RATE  DURATION  SEQUENTIAL   DIRECT
   13:19,19SEP2002   NRXX807.  SDPDPA.PK.MVSP.RT.NDMGIX.DATA.....  8.4   0:07:08      0.0       100.0
   13:19,19SEP2002   NRXX807.  SDPDPA.PR.MVSP.RT.NDMGIXD.DATA.....  11.2   0:06:42      0.0       100.0
   13:33,19SEP2002   TSJHM...  SDPDPA.PR.MVSP.RT.NDMRQFDA.DATA.....  0.3   2:21:58      0.0       100.0
   13:33,19SEP2002   TSJHM...  SDPDPA.PR.MVSP.RT.NDMRQF.DATA.....  2.8   3:37:53      0.0       100.0
   13:33,19SEP2002   TSJHM...  SDPDPA.PK.MVSP.RT.NDMTCF.DATA.....  11.1   6:24:10      0.1        99.9
```

While not shown in the above example, CPExpert also shows the OPEN time for the VSAM data set. This normally is the duration of the current OPEN. If %LET VSAMSMRY=Y; was specified in USOURCE(DASGUIDE), the OPEN time represents the sum of the times the VSAM data was OPEN for all TYPE64 records in the performance data base.

Suggestion: If CPExpert consistently produces Rule DAS625 for the same VSAM data sets, and the related jobs are having performance problems randomly accessing VSAM data, *you can improve performance with no changes in your applications*. You should consider the following alternatives:

- **Consider the use of System Managed Buffering**. If the data sets listed by Rule DAS625 are SMS managed, have extended format, and your installation has DFSMS V1R4 or later, you can use system managed buffering[3]. System managed buffering enables VSAM to determine the optimum number of index and data buffers, as well as the type of buffer management (LSR or NSR). IBM benchmarks[4] show up to 90% reduction in EXCPs, DASD connect time, and CPU time when converting to system managed buffering!

  To indicate that VSAM is to use SMB, specify either of the following options:

  - Specify the ACCBIAS subparameter of the JCL DD statement AMP parameter.

[3] Please review *DFSMS: Using Data Sets* (Section 2.5.4.2.3: Processing Guidelines and Restrictions) before implementing system managed buffering.

[4] *VSAM Demystified*, SG24-6105, Table 7 (Direct access: Benefits of using SMF: Updates and insertions)

- Specify Record Access Bias in the data class and an application processing option in the ACB. The ACB must be NSR and MACRF cannot contain any of the following:

  - ICI (Improved control interval processing)
  - AIX Processing the data set through the alternate index of the path specified in the DDname
  - UBF (Management of I/O buffers left up to the VSAM user)

- **Specify Direct Optimized (DO)**. Regardless of whether the JCL DD statement AMP is used or the data class/ACB is used, you should specify Direct Optimized (DO) for the record access bias if *direct access is near 100% for the VSAM data sets listed*. The DO processing technique optimizes for totally random record access. This technique overrides the user specification for nonshared resources (NSR) buffering with a local shared resources (LSR) implementation of buffering.

- **Specify Direct Weighted (DW)** for the record access bias if the data sets are processed using a mixture of direct and sequential access, but direct access is not near[5] 100%. DW processing provides the minimum read-ahead buffers for sequential retrieval and the maximum index buffers for direct requests.

- **Consider the use of Batch LSR**. For batch applications **and if Rule DAS625 shows that direct processing[6] is near 100%**, you can use Batch LSR. Batch LSR provides advantages in an application using VSAM NSR buffering techniques to switch to LSR without changing the application source code or link-editing the application again. Only a JCL change is required.

- If the above actions are not appropriate, you can change the NSRDIR guidance variable in USOURCE(DASGUIDE). Section 3 describes how to change the NSRDIR guidance variable if you feel that Rule DAS625 is produced prematurely.

- Alternatively, you can exclude the reported VSAM data sets from analysis. Section 3 describes how to exclude VSAM data sets from analysis. However, you should be aware that no analysis of potential

[5] Note that the default value for the NSRDIR guidance variable is 75, indicating that more than 75% of the processing for the VSAM data set was direct access. Unless this guidance is reduced, Rule DAS625 should not be produced unless more than 75% of the accesses were direct.

[6] Using the Batch LSR subsystem with sequential access could degrade performance rather than improve it. This is because Batch LSR does not provide "read-ahead" capability. The "read-ahead" capability is essential for good performance with sequential access.

VSAM problems will be performed on data sets that are excluded from analysis.

Reference: *DFSMS: Using Data Sets* (SC26-7339 for OS/390; SC26-7410 for z/OS)
Section 2.5.4: Determining I/O Buffer Space for Nonshared Resource
Section 2.5.4.2.3: Processing Guidelines and Restrictions

