
Rule CIC166: VSAM LSR files have same CI size for data and index records

Finding: CPExpert has detected that VSAM files assigned to a Local Shared Resources (LSR) pool have been defined with the same control interval (CI) size for data and index records.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: VSAM files assigned to a LSR pool share common buffers (and also share strings) assigned to the LSR pool. Since the buffers are shared, significantly fewer buffers normally are required to support I/O access operations. This is because not all files will be accessed at any particular time. Please refer to Rule CIC165 for more discussion of this issue.

The size of the CIs for data sets is not an operand specified to CICS. Rather, the size of the CIs is specified using VSAM (using the IDCAMS utility). It is common to specify the CI for data records, and to allow VSAM to select an appropriate CI for index records. (In fact, VSAM will override any specified CI size for index records if VSAM decides that the specified size is insufficient.)

LSR pools are generated using the Resource Definition Online (RDO) LSRPOOL definition, or using the DFHFCT TYPE=SHRCTL macro. The BUFFERS operand of this macro is used to define the number and size of the buffers assigned to the pool. Buffers are normally specified with different sizes, with the different sizes corresponding to the size of the CI for data and index records of files assigned to the LSR pool. The pools with different buffer sizes are referred to as LSR subpools.

When executing an I/O operation, VSAM selects a buffer sufficiently large to hold the CI, regardless of whether the CI is for data or index records. Buffers for index CIs and data CIs will be drawn from the same buffer subpool if any file has index records with the same CI as the data records for any file in the LSR pool. CICS performance can be severely degraded if any LSR subpool services both data records and index records. This is because the VSAM "look-aside" logic will tend to find CIs for data records in the buffers, rather than finding CIs for index records.

A major advantage of using LSR pools for VSAM files is that VSAM will use its "look-aside" logic to determine whether a required CI is already in a

buffer, before executing any physical I/O operations. If the required record is already in a buffer, VSAM will use the record in storage, rather than issuing a read to DASD. This has the effect of implementing an in-storage caching of the file, and can **significantly** reduce the number of physical I/O operations required.

With sufficient buffer allocation, VSAM often can find 80%-95% of the I/O requests for index records in buffers, and over 50% of the I/O requests for data records in buffers. Whether records are in a buffer is, of course, a function of the file size, the file accessing characteristics, etc. However, allocating sufficient buffers to LSR pools (and assigning VSAM files to the LSR pools) often can produce a **significant** performance improvement for CICS.

VSAM uses a least-recently-used (LRU) algorithm in deciding when to reuse a buffer in a subpool. There normally are many more CIs for data records than there are CIs for index records. Consequently, the CIs for data records will dominate a subpool if the CIs are the same for data and index records. This effect is exactly the opposite of what is desired for performance, and significantly degrades the VSAM "look-aside" logic.

The CI size of the data and index records is provided for each file in the CICS statistics. CPExpert analyzes this information to determine whether there are any conflicts between CI size for data and index records for any files assigned to LSR subpools. CPExpert produces Rule CIC166 if any LSR subpool services VSAM files where any data CI is the same size as any index CI.

CPExpert makes sure that the files have a reasonably large number of I/O operations before firing Rule CIC166 (there is little point in worrying about this situation if the file has few accesses). CPExpert arbitrarily selected 1,000 as the threshold of file accesses; CPExpert will not be produced Rule CIC166 if the number of buffer reads (this means physical I/O operations) is less than 1,000 for any LSR subpool having a conflict with the CI size of data or index records.

Additionally, CPExpert examines the "look-aside" read hit ratio (described in Rule CIC165). CPExpert does not produce Rule CIC166 if this ratio is greater than 90% for the VSAM file(s) involved. (CPExpert concludes that there is little harm being done by the same size specification, and does not produce the rule spuriously.)

Note that CICS/ESA provides separate LSR buffer pools for data and index records. However, if only data buffers are specified for the LSR pool, only one set of buffers are built. This set of buffers will be used for both data and index records. Consequently, this rule applies to CICS/ESA if only data buffers were specified for the LSR pool.

The initial version of the CICS Component does not examine the DFHFCT macros to determine specifications. Consequently, Rule CIC166 may be produced in CICS/ESA environments when there is no real problem (that is, you may have specified separate LSR buffer pools for data and index records in CICS/ESA). Depending upon user feedback, future versions of the CICS Component may analyze the DFHFCT macros.

Suggestion: Allocating sufficient buffers to LSR pools (and assigning VSAM files to the LSR pools) often can produce a **significant** performance improvement for CICS. This improvement occurs mostly because of the VSAM "look-aside" logic. However, the "look-aside" logic will normally function poorly if data and index files assigned to the LSR pool have the same CIs.

CPEXpert suggests that you consider the following alternatives:

- Analyze the data and index CI sizes for the files identified by Rule CIC166. Determine whether the data or index CI sizes can be changed so that VSAM will use buffers from different LSR buffer subpools.
- If the VSAM file characteristics cannot be changed, consider assigning the conflicting files to different LSR pools. This option eliminates the advantage of minimizing buffer requirements among the files. However, this advantage is not as significant with CICS Version 1.7, since the buffer allocation is above the 16 megabyte line. This option may allow the VSAM "look-aside" logic to significantly improve CICS performance by eliminating many physical I/O operations associated with the file.
- If neither of the above options are feasible, consider assigning more buffers to the LSR buffer subpool. This option may not improve performance, however, depending upon the file accessing characteristics.

Reference: *CICS/OS/VS Version 1.7 Performance Guide*: pages 65-68, pages 232-238, and page 244.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 170, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73, pages 93-106, and page 239.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152, page 155, and pages 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162, pages 165-166, and pages 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 Performance Guide: Chapter 18 (VSAM resource usage (LSRPOOL)), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 Performance Guide: Section 4.5.2 Defining VSAM resource usage, Section 4.5.4 Defining VSAM buffer allocations for LSR, and Appendix 1.1.17.6. |
|
|