
Rule CIC221The CICS journal buffer size may be too small (user log)

Finding: CPExpert believes that the buffer size allocated for a user journal may too small.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region or of individual tasks referencing the user journal. The level of impact depend on the number of times the "buffer full" condition was experienced by the user journal.

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

Discussion: A user journal may be used to record file control or message activity during CICS execution. CICS can be directed to perform automatic journaling by specifying entries in either the file control table or the program control, or users can provide explicit user journaling routines.

For systems prior to CICS Version 3.2.1, a single buffer is acquired for a user journal (CICS Version 3.2.1 acquires two buffers for each journal). The size of each journal buffer is specified in the BUFSIZE operand of the DFHJCT macro.

The BUFSIZE operand determines the maximum size of a physical journal block. A physical journal normally contains more than one logical journal record, however. Journal records are placed sequentially in the journal buffer until a physical journal block is written. A physical journal block is written for a user journal under the following conditions:

- The SYSWAIT=ASIS operand was entered in the DFHJCT macro. CICS maintains a dynamic threshold called the "sliding buffer shiftup value" which is adjusted according to the rate of journal requests and the amount of data written. The intent is to achieve an output rate of approximately two blocks per second. IBM discourages use of the SYSWAIT=ASIS option for CICS versions prior to Version 3.2.1, and the option is not available for CICS Version 3.2.1.
- The logical records placed into the buffer used the amount of buffer space specified in the BUFSUV operand in the DFHJCT macro (the default value of the BUFSUV operand is the BUFSIZE value). When the logical records placed into the buffer use the amount of buffer space specified in the BUFSUV operand, a physical block output is started.

For CICS prior to Version 3.2.1, the BUFSUV operand can be used to overlap physical output with placing of additional logical records into the buffer. For CICS Version 3.2.1, two buffers are acquired and overlap is managed by switching buffers.

Prior to CICS Version 3.2.1, a physical output is initiated when the buffer space used by logical records reached the BUFSUV value. CICS continues to add logical records to the buffer until the physical output is complete (or until the buffer space is used up). After the physical output is completed, the logical records added to the buffer are "shifted up" to the beginning of the buffer.

If the buffer is full, tasks using the user journal must wait until the physical output completes. This wait time delays the tasks and lengthens response. These response delays should be prevented, unless there are system design reasons for regularly writing out the journal buffers.

Additionally, there is potentially a more serious effect for the overall CICS region when tasks wait for user journal, if the journal is used by several tasks. When tasks wait for a journal output, they will become dispatchable whenever the physical output completes and their records are placed in the buffer. CICS can enter a stress condition if a number of tasks are suddenly dispatchable and require resources.

CPEXpert produces Rule CIC221 if the number of times a buffer full" condition exceeded the CPEXpert **JCBUFUL** guidance variable in CPEXPERT.USOURCE(CICGUIDE). This guidance variable is provided so that the rule will not be produced spuriously in those situations when system design decisions require user journals to be written frequently and the BUFSUV operand is insufficient to meet the requirement. The default value for the JCBUFUL guidance variable is zero, indicating that the buffer full condition should not be encountered for user journal logs. This default is set so low to ensure that you read this narrative and appreciate the potential performance implications of the buffer full conditions.

Suggestion: Unless you have unusual circumstances, CPEXpert suggests that you increase the buffer size for the user journal. This is accomplished by increasing the value of the BUFSIZE operand in the DFHJCT macro for the user journal. If your system design considerations require the user journal be written frequently, change the CPEXpert **JCBUFUL** guidance variable so that this rule is not produced spuriously.

If you do not wish to increase the buffer size, examine the BUFSUV value (pre-CICS Version 3.2.1).

-
- The default value for BUFSUV is the value specified for the BUFSIZE operand. This default means that the buffer will not be written until the buffer is full. However, if the BUFSUV value is less than the BUFSIZE operand value, the buffer will be written when the BUFSUV value is reached. Additional records will be placed into the buffer (as described above) while the physical block is written. Consequently, the output of the physical block will be overlapped with insertion of new logical records into the buffer.
 - If the value for the BUFSUV operand is not specified or if the value for the BUFSUV operand is approximately the same as the BUFSIZE value, little or no overlap will occur between writing the physical block and insertion of new logical records. Tasks wishing to write to the user log journal must wait in this case, with the potential performance degradation described above.
 - If the value for the BUFSUV operand is not specified for this user journal or if the value for the BUFSUV operand is approximately the same as the BUFSIZE value, you should specify a value for the BUFSUV operand. This value should be **less than** the BUFSIZE operand value. It is not possible to determine how much less the value should be on a general basis. However, the value should be sufficiently less that the "buffer full" condition does not arise.

If necessary, you can experiment with successively lower values until the "buffer full" condition ceases. For example, you might set the BUFSUV value to be lower than the BUFSIZE value by an amount sufficient to accommodate 2 logical records. If that specification does not eliminate the "buffer full" condition, then you should continue lowering the BUFSUV value until the "buffer full" condition is eliminated.

Note that lowering the BUFSUV value will cause the physical block size to be smaller, and more physical blocks will be written for a given number of journal records. This, of course, generates more I/O operations in writing the journal records and more I/O operations for the software processing the journal records. It probably is not wise to lower the BUFSUV value too much below the existing BUFSIZE value. Rather, you might wish to increase the BUFSIZE value and minimize the number of I/O operations.

Reference: *CICS/OS/VS Version 1.7 Performance Guide*: page 81 and pages 274-276.

CICS/MVS Version 2.1.2 Performance Guide: pages 203-205 and page 452.

CICS/ESA Version 3.1.1 Performance Guide: pages 85-87 and pages 268-271.

CICS/ESA Version 3.2.1 Performance Guide: pages 182-184 and pages 304-305.

CICS/ESA Version 3.3.1 Performance Guide: pages 192-194 and pages 323-324.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.6.2 and Appendix A.1.17.

CICS/TS: not applicable

CICS/TS for z/OS: not applicable..