
Rule CIC106: Too many tasks were queued for MAXACTIVE reason

Finding: CPExpert has detected that too many tasks in a transaction class were queued because the MAXACTIVE limit had been reached, the PURGETHRESH value had not limited the number of queued tasks, and the CICS region encountered a Short-on-Storage condition.

Impact: This finding has a HIGH IMPACT on the performance of the CICS region.

Logic flow: The following rule caused this rule to be invoked:
CIC110: CICS encountered a Short-on-Storage condition

Discussion: Please refer to Rule CIC105 for a discussion of the processing of transaction tasks: how the transactions are accepted, purged, or queued for processing.

When the number of active transactions reaches the MAXACTIVE value, attach requests are queued.

Each queued transaction requires only 256 bytes plus the size of the terminal input/output area (TIOA) holding any terminal input. However, if a large number of transactions are queued, the queue could grow to occupy all available storage in the CICS Dynamic Storage Area (DSA). CICS then could become short-on-storage, with serious performance consequences.

The PURGETHRESH value controls the number of newly-created tasks that are accepted but must be queued because of MAXACTIVE reasons. A limit on the number of queued transactions is necessary in some situations because a very large number of transactions might be queued in a heavily-loaded system.

CPExpert produces Rule CIC106 if the CICS interval statistics reported that the CICS region encountered a short-on-storage situation, and if transactions were queued for MAXACTIVE reason.

Rule CIC106 is the counterpart of Rule CIC107: Rule CIC107 suggests that the PURGETHRESH value be increased because transactions were purged but no short-on-storage situation occurred; Rule CIC106 notes that a short-on-storage situation was encountered, but transactions were queued.

CPEXpert estimates and reports the storage used by the queued transactions by multiplying the peak number of queued transactions by 256 bytes. This calculation results in a low value, as the TIOA is not included.

Suggestion: CPEXpert suggests that you review the PURGETHRESH attribute for the transaction class. Under most circumstances, the PURGETHRESH ceiling should be high enough that transactions are queued but are not purged.

However, when the CICS region encounters a short-on-storage situation, the PURGETHRESH is one way in which storage may be regained. You can reduce the number of queued transactions by specifying a lower value for the PURGETHRESH attribute for the transaction class.

CPEXpert suggests that you first evaluate other alternatives to lowering the PURGETHRESH value. The PURGETHRESH attribute is intended to prevent a significant amount of storage being used by queued transactions. This situation might occur when the CICS region slows down or when an unexpectedly large number of transactions attempt to attach.

Before specifying a smaller value for the PURGETHRESH attribute, you should explore other alternatives. Some alternatives are:

- Improve the overall performance of the CICS region so transactions are processed faster. This might involve adjusting other CICS parameters, giving CICS a higher dispatching priority, rescheduling work so that CICS does not compete for resources, acquiring additional resources, improving overall system performance, etc.
- Impose MAXACTIVE limits on transaction classes which are heavy resource users. This option would be particularly applicable if you can identify high resource using transactions, place these transactions into their own transaction class, and impose a MAXACTIVE limit on the high resource using transaction class. This alternative might allow the CICS region to process the transactions which use few resources.
- Implement task prioritization, and give a low priority to tasks which are use a large amount of system resources. Note that CICS does not implement a preemptive dispatching algorithm. This alternative might not have any positive effect to reduce the queue of transactions if the tasks that use a large amount of CPU resources are dispatched and they fail to release the processor.

There are, of course, other alternatives you should consider before adjusting the PURGETHRESH value.

Reference: *CICS/ESA Version 4.1 Performance Guide*: Section 4.7.5 and Appendix A.1.30.

CICS/TS Release 1.1 Performance Guide: Section 4.7.5 and Appendix 1.1.27.

CICS/TS Release 1.2 Performance Guide: Section 4.7.5 and Appendix 1.1.28.

CICS/TS Release 1.3 Performance Guide: Section 4.11.4 and Appendix 1.1.31.

CICS/TS for z/OS Release 2.1 Performance Guide: Chapter 23 (MAXACTIVE) and Appendix A (Table 129).

CICS/TS for z/OS Release 2.2 Performance Guide: Section 4.10.4 Using transaction classes (MAXACTIVE) to control transactions |