
Rule CIC141: Transactions experienced a stall-purge

Finding: CPExpert has determined that transactions experienced a stall-purge in a production CICS region.

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

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

Discussion: When CICS is in a short-on-storage condition or the maximum-task limit (specified by the MXT keyword in the SIT) has been reached, CICS will not attach new tasks.

In some situations, a "deadlock" situation can occur in which no task is able to proceed. This situation can arise because tasks are waiting on the completion of other tasks, but the other tasks cannot proceed because of storage limits, or because of AMXT or MXT limits. CICS considers this to be a **system stress** situation, and takes action to resolve the situation.

CICS attempts to solve the deadlock situation by temporarily raising the AMXT (active maximum tasks) value. A task may be causing a deadlock because the number of active tasks in the CICS region has reached the region's AMXT value and the task cannot execute. Potentially, temporarily increasing the AMXT value would allow the task to execute and free the deadlock.

If no tasks can proceed after raising the AMXT value, CICS waits for an interval to see if the deadlock will be resolved. (The deadlock could be resolved by the completion of some event. For example, the task causing the deadlock could be enqueued on some resource or waiting for terminal input.) The interval CICS will wait to see if the deadlock is resolved is a parameter provided by the ICVS keyword in the SIT (the default value for the ICVS keyword is 20 seconds).

After the expiration of the ICVS interval, CICS will attempt to remove the deadlock by purging tasks. CICS will begin purging those tasks which have been defined as SPURGE=YES in DFHPCT or CEDA DEFINE transaction. (The SPURGE keyword indicates whether the transaction can be purged in a CICS stall condition. The default is SPURGE=NO.) CICS will purge one transaction, selecting the lowest priority task. CICS will then wait another ICVS interval to see if the deadlock is removed.

If all transactions with SPURGE=YES are purged and no task is able to proceed, it is necessary to cancel CICS and restart the CICS region.

It is not too uncommon for stall conditions to occur in a test CICS region. This is because a stall condition may be the result of transaction interactions, and the task or system logic may not be completely debugged. **However, a system stall condition should never occur in a production CICS region.** Any purge-stalls of transactions in a production CICS region indicates the need for immediate attention to determine the cause and correct the problem.

An additional situation can arise in which individual tasks may be stalled (or deadlocked), waiting on some action by other tasks. CICS takes no action in this situation if any task in the CICS region is able to proceed. No information is provided about this situation, since CICS does not view the situation as creating a system stress condition.

This situation can be detected based upon user complaints or observant operators (perhaps using a commercially-available CICS monitoring tool). These situations can be removed by issuing a CEMT SET TASK PURGE command. However, these situations should not normally occur in a production CICS region. Remedial programming or system design should be focused on the transactions if these situations occur in a production CICS region.

CPEXpert produces Rule CIC141 if any transactions were stall-purged in a production CICS region. CPEXpert suppresses this rule for CICS Version 3 since data are not available.

Suggestion: CPEXpert suggests that you take immediate action to determine the cause of the stall-purges.

- CPEXpert will produce other rules if the CICS region has encountered a short-on-storage condition (for example, Rule CIC110 will be produced). The advice provided by these rules should be followed.
- If CPEXpert does not detect a short-on-storage condition, the system stress condition is caused by such problems as system design logic, enqueue conflicts, task logic errors, operator responsive, etc. You should determine the cause of the system stress and take remedial action.

Reference: *CICS/OS/VS Version 1.7 Performance Guide*: pages 53 and 156.

CICS/MVS Version 2.1.2 Performance Guide: pages 89 and 381.

CICS/ESA Version 3.1.1 Performance Guide: pages 116 and 174.

CICS/ESA Version 3.2.1 Performance Guide: pages 78 and 330.

CICS/ESA Version 3.3.1 Performance Guide: pages 88 and 349.

CICS/ESA Version 4.1.1 Performance Guide: Section 3.2.3 and Appendix A.1.24.

CICS/TS Release 1.1 Performance Guide: Section 3.2.3.3 and Appendix 1.1.23.

CICS/TS Release 1.2 Performance Guide: Section 3.2.3.3 and Appendix 1.1.24.

CICS/TS Release 1.3 Performance Guide: Section 3.2.3.3 and Appendix 1.1.25.

CICS/TS for z/OS Release 2.1 Performance Guide: Chapter 11 (Purging of Tasks) and Appendix A (Table 111).

CICS/TS for z/OS Release 2.2 Performance Guide: Section 3.2.2.3 Purging of tasks and APPENDIX 1.1.31.1. |