
Rule CIC285: Consider specifying protected threads for CICS-DB2 Entry

Finding: The average number of thread terminations for the CICS-DB2 Entry was sufficiently high that you should consider specifying (or increasing) the number of protected threads for the CICS-DB2 Entry.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of CICS tasks in the region that use the CICS-DB2 connection. Additionally, the finding could significantly reduce the overhead involved with the CICS-DB2 connection.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics. This finding applies only with CICS/Transaction Server for OS/390 Release 1.2 and subsequent releases of CICS.

Discussion: The CICS DB2 attachment facility creates an overall connection between CICS and DB2. CICS applications use this connection to issue commands and requests to DB2.

A CICS transaction accesses DB2 via a *thread*, which is an individual connection into DB2. The transaction uses the thread to access resources managed by DB2. Threads are created when they are needed by transactions, at the point when the application issues its first SQL or command request.

When CICS is connected to DB2 Version 5 or earlier, specially created subtask TCBs are used as the thread TCBs. When CICS is connected to DB2 Version 6 or later, open TCBs are used as the thread TCBs. The open TCB associates itself with the thread, and becomes the thread TCB until it dissociates from the thread.

There are three types of threads: Command threads, Pool threads, and Entry threads.

C Command threads are used by the CICS DB2 attachment facility for issuing commands to DB2 via the DSNB transaction.

C Pool threads are used for all transactions and commands that are not using a Command thread (because the transaction is not DSNB), are not using an Entry thread (because an Entry thread had not been defined for the transaction), or have been “overflowed” to the pool because a Command thread or an Entry thread was not available.

-
- C One or more Entry thread categories can be defined (using the DB2ENTRY definition) for specific transactions or groups of transactions. Entry threads are used for transactions that need to be managed separately from the normal transactions, or for transactions that have special accounting needs.
 - C Entry threads can have higher, equal, or lower priority relative to the CICS main task¹.
 - C Thread limits can be specified for Entry threads to ensure that certain transactions always have access to a thread, or to make sure that certain transactions do not monopolize resources (that is, thread limits can be used to “throttle back” types of work).
 - C The THREADWAIT controls the action that is taken if no Entry thread is available (wait for an Entry thread, overflow to pool, or abort).
 - C Different accounting levels and different security authorizations can be specified for Entry threads.
 - C Entry threads can be “protected” so the thread is not terminated immediately when released.

When a thread is no longer needed by the transaction, the thread is released. The thread release typically occurs after syncpoint completion. Command threads will be terminated immediately upon release.

Pool threads will be terminated immediately upon release, unless there is a queued transaction specifying the same plan and using a pool thread. In this case, the pool thread can be reused. This condition (a queued transaction specifying the same plan and using a pool thread) occurs rarely. Consequently, pool threads are rarely reused.

When a transaction releases an Entry thread, the thread can be reused by another transaction specifying the same plan and defined in the same DB2ENTRY. If there is no queued transaction defined in the same DB2ENTRY, Entry threads may be terminated immediately, or may be retained for a period of time, depending on whether “thread protection” has been specified.

Thread protection is specified by the PROTECTNUM attribute of the DB2ENTRY definition. The PROTECTNUM attribute specifies the maximum number of protected threads allowed for the DB2ENTRY.

¹The PRIORITY attribute is ignored with DB2 Version 6 and subsequent versions.

When an Entry thread is released, the thread will be “protected” if the current number of protected threads for the Entry (plus the thread being released) is not greater than the value of the PROTECTNUM attribute of the DB2ENTRY definition. The thread will stay “protected” for an interval controlled by the PURGECYCLE attribute of the DB2CONN² definition.

The default value for the PURGECYCLE attribute is specified as 30 seconds. A protected thread is terminated only after two complete purge cycles if it has not been reused. However, the purge cycle calculation does not begin as each thread is released, but is a timing value (conceptually, a timer interrupt) that runs continuously. The algorithm for an individual thread simply counts the number of purge cycle endings that occur after the thread is released.

The implication of this logic is that, with the default purge cycle value of 30 seconds, a specific protected thread could be protected for a range of 30 to 60 seconds depending on when the first purge cycle ending occurred with respect to the thread release.

C The protected thread could encounter a purge cycle ending immediately upon thread release and encounter another purge cycle ending 30 seconds later. These two purge cycle endings would satisfy the “protected for two purge cycles” logic even though only 30 seconds had lapsed.

C The protected thread could encounter a purge cycle ending 30 seconds after thread release and encounter another purge cycle ending 30 seconds later. These two purge cycle endings would satisfy the “protected for two purge cycles” logic with 60 seconds lapsed time.

The average thread protected interval, for transactions arriving randomly, would be 45 seconds.

There is a significant reduction in overhead when threads are reused. On the other hand, unused protected threads generate overhead since storage is required and CPU overhead is required for TCB scans. Consequently, there is a performance tradeoff: it is desirable to use protected threads if there is sufficient transaction rate for transactions defined in the same DB2ENTRY that the threads are likely to be reused, while it is undesirable to use protected threads if the protected threads are not likely to be reused.

CICS-DB2 Entry statistics are available in MXG file CICDB2RE. The information in CICDB2RE is presented as one record for each Plan, for

²Note that the PURGECYCLE attribute is specified on the DB2CONN definition. This means that the specified PURGECYCLE value applies to all DB2ENTRY definitions.

each Entry. CPExpert uses data in CICDB2RE to calculate the average thread terminates for each DB2ENTRY, using the following algorithm:

$$\text{Average thread terminates per DB2ENTRY} = \frac{\text{Thread terminates}}{\text{Recording interval duration}}$$

where

Thread terminates = D2RTHRTE

Recording interval duration = DURATM

The average thread terminates reflects the opportunity for threads to be reused if the terminates occur within the time specified by the PURGECYCLE value. As mentioned above, with the default value of 30 seconds for the PURGECYCLE attribute, the average thread protected interval would be 45 seconds. If the average thread terminate frequency is one or more terminates within 45 seconds, there is opportunity for protected threads to be reused.

CPExpert produces Rule CIC285 when the average thread terminates is more than the value specified by the **THRDTERM** guidance variable in USOURCE(CICGUIDE). The default value for the **THRDTERM** is 45 indicating that CPExpert should produce Rule CIC285 whenever the average thread terminate rate was more than one termination per 45 seconds³.

Suggestion: If Rule CIC285 is produced regularly, you should consider the following alternatives:

C Specify a larger value for the PROTECTNUM attribute. If Rule CIC285 is produced, threads are terminated (rather than being reused) at a rate of more than one termination per 45 seconds. This logic is independent of the PROTECTNUM attribute; threads would be reused (rather than terminated) if a protected thread were available. Since the threads were terminated, either no threads were protected or insufficient threads were protected.

The default value for the PROTECTNUM attribute of DB2ENTRY definition is PROTECTNUM=0. You should consider specifying a larger value than is currently specified (you have specified either the default or a value that might be lower than optimal).

³The algorithm properly should consider the value of the PURGECYCLE attribute of the DB2CONN definition. Unfortunately, the PURGECYCLE value is not available in CICS statistics, so CPExpert uses the default PURGECYCLE value of 30 seconds in the algorithm, which yields a 45 second average protect time.

C Verify that protecting threads will not cause accounting problems.

An accounting record normally is produced for each thread *termination* and for each new user signon. This means that only one accounting record would be produced if the thread stays active (because it is protected) and the same user continued using the thread. This accounting record would contain summarized values for all transactions using the same thread under the same user ID, but it would not be possible to assign any value to a specific transaction.

This potential accounting problem can be eliminated by specifying ACCOUNTREC(UOW) to make sure an accounting record is cut per unit of work, or by specifying ACCOUNTREC(TASK) to make sure there is an accounting record cut per CICS task.

C Modify guidance. You can modify the THRDTERM guidance variable in USOURCE(CICGUIDE) if you feel that Rule CIC285 is produced prematurely.

Reference: *CICS/TS Release 1.3 CICS DB2 Guide*: Section 5.4 (Creating, using, and terminating threads)

CICS/TS Release 1.3 Resource Definition Guide: Section 5.1.3 (DB2CONN) and Section 5.1.4 (DB2ENTRY)

CICS/TS for z/OS Release 2.2 CICS DB2 Guide: Section 5.5 (Selecting thread types for optimum performance)

CICS/TS for z/OS Release 2.2 Resource Definition Guide: Section 2.3.4 (DB2 connection definition attributes) and Section 2.4 (DB2 entry definitions)