Rule CIC260: Insufficient sessions may have been defined

Finding: CPExpert believes that insufficient sessions may have been defined for the CICS region.

Impact: This finding should normally have a MEDIUM IMPACT on the performance of the CICS region. However, the finding could have a HIGH IMPACT on the performance of individual transactions if these transactions are queued for lengthy intervals.

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

Discussion: A CICS region communicates with other CICS regions on the same system using the Multiple Region Operation (MRO) facility. The support within CICS that provides region-to-region communication is termed Interregion Communication (IRC). IRC is implemented by either (1) a CICS-supplied interregion program running in Supervisor state, or (2) cross-memory services provided by MVS.

A CICS region communicates with other systems using data formats and protocols embodied in IBM Systems Network Architecture (SNA). The support within CICS that provides region-to-system communication is termed Intersystem Communication (ISC). Within the following discussion, the CICS region that initiates communication is termed the "front-end" system and the other region/system is termed the "back-end" system.

A CICS region can communicate with another CICS region or with a remote system only after a "communication link" has been established between the CICS region and the other CICS region or the remote system. The definition of a communication link to a remote system consists of two basic parts:

- **The definition of the remote system itself**. The remote system is defined by the DEFINE CONNECTION command, using the Resource Definition Online (RDO) facility. The remote system is defined by the DFHTCT TYPE=SYSTEM macro, if the macro-level resource definition facility is used.

- **The definition of sessions with the remote system**. The sessions with the remote system are defined by the DEFINE SESSIONS command, using the Resource Definition Online (RDO) facility. The sessions are defined by the DFHTCT TYPE=SYSTEM macro, if the macro-level resource definition facility is used.

Regardless of the method used to define sessions, sessions are defined as "send" sessions and "receive" sessions. The concept of send sessions and receive sessions differs depending upon whether IRC or ISC is used for the communication.

- **IRC communication**. With Interregion communication, each session on the link is characterized as either a SEND or RECEIVE session. SEND sessions are used to carry an **initial** request from the local to the remote region and to carry any other data flows associated with the initial request. RECEIVE sessions are used to receive an **initial** request from the remote system.

  There can be a different number of SEND sessions and RECEIVE sessions. However, session types must be consistent between the local and remote regions. That is, there must be as many RECEIVE sessions defined on the remote region as there are SEND sessions defined on the local region, and there must be as many SEND sessions defined on the local system as there are RECEIVE sessions defined on the remote system.

  After the initial request has been made and received, the regions can communicate without regard to the SEND/RECEIVE characteristics of the session.

- **ISC communication (LU6.1)**. With Intersystem LU6.1 communication, each session on the link is characterized as either a SEND or RECEIVE session. A SEND session is one in which the local CICS is the secondary and is the contention winner. SEND sessions can initiate communication at any time. A RECEIVE session is one in which the local CICS is the primary and is the contention loser. RECEIVE sessions can initiate communication only after "bidding" for permission to begin a bracket. To avoid the overhead of bidding, the number of SEND and RECEIVE sessions should be consistent with the expected flow between the two systems.

- **ISC communication (LU6.2)**. With Intersystem LU6.2 communication, sessions are grouped into sets of sessions, referred to as modegroups. The sessions in each modegroup have identical characteristics, except the sessions are designated as contention winners or contention losers. The SESSIONS definition (for RDO definition) or DFHTCT TYPE=MODESET definition (for macro level definition) specifies the maximum number of contention winners in each modegroup. CICS determines which sessions are to be contention winners or contention losers when the sessions are bound.

A contention loser session can be converted to a contention winner session, if necessary, during operation. This conversion requires additional overhead (the bidding process must be implemented). The overhead should be avoided, if possible, by specifying a correct number of contention winners in the MAXIMUM sessions definition (for RDO definition) or in the MAXSESS definition (for macro level definition).

Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated. Once established, the session normally exists for a long time and can be used by many different transactions. The session normally is terminated by a FREE command.

A session must be available in order to be allocated in response to the ALLOCATE command. If a session is not available, CICS will normally queue the allocate request (and suspend the transaction) until a session is made available. Optionally (using the NOQUEUE specification), control can be returned to the transaction that can take application-dependent action based on the unavailability of a session.

Note that a transaction could remain suspended indefinitely, waiting allocation for a session. The DTIMOUT parameter in DFHPCT TYPE=ENTRY definition can be used to limit the amount of time a transaction is suspended; CICS will terminate a transaction that has been suspended longer than the DTIMOUT value. The DTIMOUT specification can be used to prevent deadlock situations.

With LU6.2, the ALLOCATE command may request allocation from a specific session modegroup (with its own characteristics), or may not request any particular session modegroup. If a specific session modegroup is requested, CICS will restrict its allocation attempt to that modegroup. If no particular session modegroup is requested, CICS will attempt to allocate a session from any modegroup, **but selects modegroups in the order in which the modegroups are defined**.

This last point may be significant from a performance viewpoint. Rule CIC262 describes the implications of this allocation process in more detail.

With LU6.2, if CICS queues the ALLOCATE request, CICS attempts to make a session available by alternate means.

- CICS determines whether any unbound contention winner sessions exist. If an unbound contention winner session exists, the session is bound and allocated.

- CICS determines whether any unbound indeterminate sessions exist. Indeterminate sessions may exist when the total number of contention winner sessions defined on both the front-end and back-end is less than the total number of sessions defined. If an unbound indeterminate session is found, CICS declares it to be a contention winner session. The session is then bound and allocated.

- CICS determines whether any bound contention loser sessions exist but are unallocated. If a bound contention loser session exists, a bid is issued to the back-end, and the session is allocated after permission is obtained.

Some queuing for allocation requests may be unavoidable because an installation may have deliberately restricted the number of sessions to minimize resource use by CICS. The CICS/ESA Version 3.3.1 Performance Guide lists a number of effects potentially caused by increasing the number of sessions:

- The amount of real and virtual storage required to support CICS may increase.

- The use of storage on GATEWAY NCPs in the network may increase.

- The use of storage by VTAM may increase.

- The line loading in the network may increase.

- The back-end CICS system may not be able to cope with an increased workload from the front-end system.

- The increased control block scanning by CICS may degrade performance.

In fact, some installations deliberately restrict the number of sessions as a means of controlling the number of CICS tasks (rather than using the TCLASS control method).

The decision of whether to increase the number of sessions should be viewed as a tradeoff between the potential disadvantages, versus the advantages of less task suspension, faster response, shorter transaction life, earlier release of resources, etc.

From a performance view, the ISC/IRC statistics provide information that can be used to assess whether enough sessions have been defined, whether the balance between contention winner sessions and contention loser sessions is appropriate, and whether there conflicting usage between APPC modegroups.

CPExpert detects the potential problem caused by queuing session ALLOCATE requests for **generic** sessions in Rule CIC260. CPExpert detects the potential problem caused by queuing session ALLOCATE requests for **specific** modegroups in Rule CIC261.

For Rule CIC260, CPExpert performs the following analysis:

- CPExpert evaluates the Peak Outstanding Allocates (A14ESTAM). A14ESTAM contains a count of the maximum number of ALLOCATE requests that were queued by CICS when a session could not be allocated.

  Note that if the NOQUEUE option were exercised by a user, unsatisfied ALLOCATE requests would not be included in this count. Unsatisfied ALLOCATE requests that were returned to the user with a SYSBUSY indicator would be included in Failed Allocates Due to Sessions in Use (A14ESTAO for generic requests). (Please refer to Rule CIC167.)

- CPExpert detects a **potential** problem when the maximum number of queued ALLOCATES for generic session allocation requests is greater than the **ALLOCQ** guidance variable in USOURCE(CICGUIDE). The default specification is **ALLOCQ=1**, indicating that Rule CIC260 would be produced if more than one ALLOCATE request could not be satisfied because no sessions were available. **This low default value is intended only to alert you to a potential problem with the number of sessions defined, and is intended to make you aware of this analysis mechanism**.

  For many installations, the default should be changed after executing the CICS Component a few times. The **ALLOCQ** variable should normally be used to cause CPExpert to signal a problem only when you wish to be informed of abnormal situations. For example, some installations always have a few ALLOCATE requests queued. Occasionally, however, several hundred or even several thousand requests are queued. Analysts at these installations are not concerned about the few queued requests, but are concerned about the situations when hundreds or thousands of requests are queued.

Suggestion: CPExpert suggests that you consider increasing the number of sessions defined. Please refer to the above discussion to assess whether the number of sessions should be increased.

Alternatively, you may need to change the distribution of contention winners versus contention losers in LU6.1 or LU6.2 environments. Rule CIC163 provides more discussion of this alternative.

Alternatively, change the **ALLOCQ** guidance variable to cause CPExpert to signal a potential problem only when you view the problem as serious.

Reference: CICS/TS for z/OS Release 2.3 *Resource Definition Guide*
Chapter 2.1 (CONNECTION resource definition)

CICS/TS Release 3.1 Performance Guide:
Section 2.2.26 (ISC/IRC system and mode entry statistics)
Appendix 1.1.13 (ISC/IRC system entry: Resource statistics)

CICS/TS for z/OS Release 3.1 *Resource Definition Guide*
Chapter 2.2 (CONNECTION resource definitions)

CICS/TS Release 3.2 Performance Guide
Chapter 39. (Interpreting ISC/IRC system and mode entry statistics)
Table 81 (ISC/IRC system entry: Resource statistics)

CICS/TS for z/OS Release 3.2 *Resource Definition Guide*
Chapter 6 (CONNECTION resource definitions)

CICS/TS Release 4.1 Performance Guide
Chapter 41. (Interpreting ISC/IRC system and mode entry statistics)
Table 96 (ISC/IRC system entry: Resource statistics)

CICS/TS for z/OS Release 4.1 *Resource Definition Guide*
Chapter 7 (CONNECTION resource definitions)

CICS/TS Release 4.2 Performance Guide
Chapter 31. DFHSTUP reports. (Interpreting ISC/IRC system and mode entry statistics) Table 108 (ISC/IRC system entry: Resource statistics)

CICS/TS for z/OS Release 4.2 *Resource Definition Guide*
Chapter 7 (CONNECTION resource definitions)

CICS/TS Release 5.1 Performance Guide
Chapter 31. DFHSTUP reports. (Interpreting ISC/IRC system and mode entry statistics) Table 108 (ISC/IRC system entry: Resource statistics)

CICS/TS for z/OS Release 5.1 *Resource Definition Guide*
Chapter 7 (CONNECTION resource definitions)

