
Rule CIC310The paging rate was too high for the CICS region

Finding: CPExpert detected that the paging rate was too high for the CICS region.

Impact: This finding should normally have a MEDIUM IMPACT or 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: Page-in operations occur when CICS attempts to reference a virtual page and the page is not in central storage. A page fault occurs and MVS is required to bring the page into central storage. During this page-in operation, the CICS region stops processing until MVS fetches the page.

The time the CICS region spends waiting for a page is not normally serious if the page is retrieved from expanded storage. The time to fetch a page from expanded storage is very small (on the order of 40 - 75 microseconds).

The wait time is much more lengthy (typically on the order of 20-50 milliseconds) if the page must be fetched from auxiliary storage. During this more lengthy time, the entire CICS region stops processing. Note that this situation is unlike a TSO environment, in which only a single TSO user is delayed because of a page fault. With CICS, the entire region is delayed. This is because each TSO user is an "address space" from the MVS perspective, while the entire CICS region is considered an "address space" by MVS.

The time in which CICS stops processing delays the transaction experiencing the page fault. This delay might not be so significant unless the task has many page faults. However, all of the CICS region stops processing, so all other active tasks can be delayed. Since all tasks are delayed, the tasks remain in the region longer and the tasks retain the allocated resources longer. Therefore, in addition to the delay associated with the page fault resolution, serious paging can lead to CICS stress conditions as tasks are delayed.

Paging information is not available in CICS statistics. This information is available only if you have monitor data in a MXG or MICS performance data base.

The amount of paging which can be tolerated by a CICS region depends upon how many active tasks are normally in the region, depends upon the

response objectives for the applications in the region, etc. Consequently, no single "good" or "bad" paging rate can be established. IBM's CICS Performance Guides suggest that less than one page per second is best, less than five pages per second might be acceptable, and more than ten pages per second is a major problem. These values must be evaluated considering the probability that any average page rate reflects periods in which the page rate is many times the average.

CPEXpert fires Rule CIC310 if the average page-in rate (pages per second) was greater than the **CICPAGIN** guidance variable. The default specification for this variable is **%LET CICPAGIN = 2**, indicating that the CICS region should not experience more than an average two pages per second. You can provide different guidance to CPEXpert by changing the CICPAGIN variable if you feel that Rule CIC310 is firing prematurely.

Suggestion: There are many ways to reduce the paging rate for CICS, and the "best" way varies from installation to installation. CPEXpert suggests that you consider the following alternatives to reduce the page-in rate:

- **Implement storage isolation for the CICS region.** This alternative is the easiest, and is the most common. Storage isolation is accomplished by using the PWSS parameter in IEAIPSxx for the performance group to which the CICS region is assigned. Implementing storage isolation would guarantee a minimum target working set to the CICS region.

You normally should set the minimum protected working set (the first value in PWSS parameter) based upon the average working set size. You can obtain the average working set size from inspecting RMF Monitor II display (Address Space State Date Report).

- **Dynamically control the target working set for CICS region.** PPGRTR is a keyword in the IEAIPSxx member of SYS1.PARMLIB that can be used to dynamically control both the amount of paging and the central storage used by the private area.

If PPGRTR is not specified, the target (or protected) working set is always at the minimum value specified for the PWSS keyword. If PPGRTR **is** specified, the target (or protected) working set is dynamically adjusted based upon the page-in rate. The page-in rate is calculated every 10 seconds. The page-in rate is based upon the number of page-in operations from auxiliary storage per second of residency time.

When the page-in rate from auxiliary storage falls below the value specified as the first parameter in the PPGRTR keyword, the target

working set is decreased. The reason for this action is that paging has fallen to a level such that sufficient private pages are in memory to satisfy memory demands. The implication is that memory demand does not require the target working set to be as large, and too much is being protected. The System Resources Manager (SRM) reduces the target working set by 3% when the paging rate falls below the value of the first parameter.

When the page-in rate from auxiliary storage rises above the value specified as the second parameter in the PPGRTR keyword, the target working set is increased. The reason for this action is that paging has risen to a level such that sufficient pages are **not** in memory to satisfy memory demands. The implication is that memory demand requires the target working set to be larger to reduce page replacement. The SRM increases the target working set by 7% when the paging rate rises above the value of the second parameter.

Thus, the PPGRTR keyword allows the target working set size to be dynamically adjusted, based upon varying memory demands. Central storage is protected when needed to minimize paging, but made available when not needed. Please refer to Rule SRM113 in the SRM Component User Manual for further discussion of the PPGRTR keyword.

For example, specify **PPGRTR=(2,5)** to indicate that the target working set should be increased when the paging rate rises above 5 pages per second and that the target working set should be decreased when the paging rate falls below 2 pages per second.

- **Specify "negative" storage isolation for selected low priority workloads.** When storage isolation is specified, the page stealing algorithm does a "pre-scan" of address spaces, stealing pages from any address spaces whose working set exceeds the maximum protected working set (the maximum value specified in PWSS). (Actually, it also includes the maximum value specified in CWSS, but specifying negative storage isolation for the common area is strongly discouraged.)

Thus, page stealing can be directed to specific workloads by specifying a maximum value in PWSS that is less than the amount of storage commonly used by these workloads. "Negative" storage isolation might, for example, be used with low priority batch jobs to allow more page frames for use by CICS.

- **Limit the number of competing workloads requiring processor storage.** This can be done by (1) controlling the system multiprogramming level based upon paging rate (adjust the

RCCPTRT parameter in the IEAOPTxx member), (2) controlling the system multiprogramming level based upon page delay time (adjust the RCCPDLT parameter in the IEAOPTxx member), (3) controlling the system multiprogramming level based upon real memory use (adjusting the RCCUICT parameter in the IEAOPTxx member), or (4) limiting the multiprogramming level of selected workloads (reduce the maximum MPL of the CNSTR parameter of selected domains in IEAIPSxx). (Please refer to RULE SRM205 in the SRM Component's User Manual for a discussion of MPL controls.)

- **Workload scheduling.** Schedule lower priority workloads to a time when they do not compete with CICS.
- **Decrease the amount of central storage used for logical swapping** (adjust the LSCTxxx parameters in IEAOPTxx). If less central storage is used for logical swapping, more central storage will be available for active working sets and less paging will occur. Note, however, that decreasing logical swapping will increase the swap-in delay. This delay could have a detrimental effect on TSO performance. (Please refer to RULE SRM202 in the SRM Component's User Manual for a discussion of logical swapping controls.)

If the above alternatives can not be used to reduce the paging for the CICS region, you should consider alternatives to improve the page fault resolution.

- **Place paging devices on dedicated paths.** Significant performance improvement can be realized by placing paging devices on dedicated paths. Consider reconfiguring to place the paging devices on dedicated paths, if they are not already on dedicated paths.
- **Ensure that local page data sets reside on dedicated (or low utilized) devices.** Considerable queuing delays may result if local page data sets reside on devices which have other active data sets. RULE MVS140 will fire if this is a likely problem. (RULE MVS140 analyzes the number of I/O operations to local page data sets, compares that number with the number of page I/O operations to the data set, and fires if the difference is more than a token amount.)
- **Increase the number of local page data sets.** Significant performance improvements might be realized by increasing the number of local page data sets. This will allow more paths to be used, and less queuing to result. Perhaps more importantly, the page slots will be placed on more physical devices and there is less chance of seek delay.

-
- **Acquire faster paging devices.** If the above options have been exhausted and paging delays are still unacceptable, you should consider acquiring faster paging devices.

Additionally, the CICS Performance Guides for Version 1.7 and Version 2.1.2 list many application-related changes which can be implemented to reduce paging effects.

Reference: *CICS/OS/VS Version 1.7 Performance Guide*: pages 158-159 and pages 374-376.

CICS/MVS Version 2.1.2 Performance Guide: pages 89-91 and pages 335-336.

CICS/ESA Version 3.1.1 Performance Guide: pages 175-176.

CICS/ESA Version 3.2.1 Performance Guide: pages 79-80.

CICS/ESA Version 3.3.1 Performance Guide: pages 89-90.

CICS/ESA Version 4.1.1 Performance Guide: Section 3.2.5.

CICS/TS Release 1.1 Performance Guide: Section 3.2.5.

CICS/TS Release 1.2 Performance Guide: Section 3.2.5.

CICS/TS Release 1.3 Performance Guide: Section 3.2.5. |