

## CICS\TS and TCB Switching

March 10, 2006

By: Eugene S. Hudders

### CICS/TS Performance Tuning Using C\TREK Course

This seminar covers a lot of tuning material and will require 8 hours a day. The total time for the seminar is 40 hours.

The followings topics included in this course are:

- Introduction to CICS Performance Tuning
- Using Operating System Information to Tune CICS/TS
- Tuning CICS/TS Processor Cycles
- Tuning Real Storage in CICS/TS
- Tuning Virtual Storage in CICS/TS
- Tuning CICS/TS Transaction Controls
- Tuning On Line VSAM Files
- Tuning NSR Files in CICS/TS
- Tuning CICS/TS LSR Buffer
- Tuning the Index CISZ
- Reviewing VSAM Free Space
- Reviewing the DB2 Interface

If you wish to attend this course, please send us an email at [ctrek@actpr.com](mailto:ctrek@actpr.com)

CICS/TS V2 brought a series of improvements to web support and Java processing. In addition, this version brought in a new performance feature for DB2 V6 (and higher) users called the Open TCB environment. A new CICS TCB construct was created called an "L8 TCB" to handle the open TCB environment and a new programming concept called "threadsafe" was also introduced. The open TCB environment provides an opportunity for a performance improvement in multi-processor systems that included both better throughput and in some cases lower CPU utilization. However, the improvements were not automatic and if implemented incorrectly, higher CPU utilization could result. This article will address these issues and provide recommendations on how to implement this support. The first part of the article reviews the definition of the environment and an explanation of threadsafe programming while the second part of the article reviews the way the TCB structure works with an application task and the dispatching mechanism.

To implement the Open TCB Environment you have to have DB2 V6 or higher and DB2 support in the CICS/TS system. The total number of L8 TCBs available to the CICS system is defined via the System Initialization Table (SIT) parameter called MAXOPENTCBS that creates a pool of L8 TCBs that can have several uses. The L8 TCBs are used for exit processing, DB2 calls (SQL), OPENAPI defined programs (announced in CICS/TS V3R1 and will be discussed later in the article) and for program processing of a threadsafe application. So, as you can see, the L8 TCBs can be used for other things that are not DB2 related such as exits. The TCBLIMIT in the DB2CONN definition establishes how many of the MAXOPENTCBS are going to be used by DB2 related work. TCBLIMIT can equal MAXOPENTCBS but you may not be able to

dispatch an exit if all the L8 TCBs have been allocated to DB2 requests. So, usually, TCBLIMIT is defined with a lower value than MAXOPENTCBS to account for other L8 TCB uses (Figure 1).

```

D2GL ADVANCED COMPUTER TECHNOL  C\TREK ON-LINE      09D0E280  DATE 03/15/2006
APPLID CICSTS31                  DB2          TIME 13:35:02
VERSION 6.4                      DB2CONN INFORMATION  TERM 0208

CONNECT NAME      RCT1$      EX1 GLOBAL WORK  00225134  MAXOPENTCB (SIT)    20
CICS NAME        CICSTS31  FREE TCB CHAIN  00000000  TCBLIMIT            12
DB2 NAME        DSN1     TCB READYQ CHAIN 00000000  CURR TCB COUNT      0
DB2 RELEASE     0710    ATTACH/DETACH CHN 00000000  HWM TCBS             0
SIGNON ID       CICSTS31  UR INDOUBT LOTS  00000000  FREE TCBS           0
STATISTICS DEST QUEUE CSSL          READYQ TASKS       0
MESSAGE DEST QUEUE 1 CSMT  SERVICE ECB      8001430B  PEAK # TASKS        0
MESSAGE DEST QUEUE 2                                INDOUBT COUNT      0
MESSAGE DEST QUEUE 3                                ATTACH PARM LIST  09D0E3C4  PURGE CYCLE
STATE FLAGS      48     MSB FW SAVE AREA 09D0E37C  MINUTES              0
STANDBY FLAGS    RECONNECT WORKAREA      09D0E40C  SECONDS              30
CONN ERROR FLAGS SQLCODE                    CONNECT TIME        14:43:55.9
TERMINAL RELEASE NO  ABEND PROG NAME      DISCONN TIME        00:00:00.0
THREAD ERRORS    ABEND ABEND PROG ADDR  00000000  SEC REBUILT          00:00:00.0
CONNECT STATUS   CONNECTED SAVE AREA (0-15)  09D0E604
ATTACH STATUS    CEX2  START PSW AT ERROR    09D0E644
SHUTDOWN FLAG    N/A
MSB ERROR FLAG   N/A
IDENTIFY FLAGS   N/A
PF4 - GLOBAL STATS  PF9 - POOL THREADS  PF10 - COMMAND THREADS
4B  :00.1 01/54
    
```

Figure 1. DB2CONN Information (MAXOPENTCBS & TCBLIMIT)

The concept of threadsafe can cause some confusion because there is a tendency to think of threadsafe program as being a reentrant program. However, a threadsafe program has to rise to a higher standard. A threadsafe program has to ensure the integrity of shareable resources through the use of ENQ/DEQ commands or some type of synchronization instructions such as the assembler instructions of Compare and Swap (CS) or Compare Double and Swap (CDS). Examples of shared resources are shared storage obtained via a GETMAIN SHARED command or access to fields in the Common Work Area (CWA) among others. This action was not previously required in CICS because of the way CICS dispatched application code in the system. Application code was “automatically” serialized because it had to be dispatched on a single TCB called the QR or Quasi-Reentrant TCB. Since you could not have multiple application programs running simultaneously within CICS, the QR provided the protection mechanism for shared resources such as the CWA. However, this limited the multi-processing capacity of CICS to specialized functions and in many installations, the QR TCB reflected 80 to 95% of the total CPU used by the CICS address space.

The L8 TCBs can now be used to dispatch application programs that could run in parallel (simultaneously) with the QR TCB and other L8 TCBs as long as there are sufficient processors to support the parallelism. This parallelism now presents the situation where two different tasks running simultaneously on separate TCBs and processors want access to the same resource. If there is no synchronization, you could wind up with invalid or compromised data. If you are “lucky” your program will cancel making it “easy” to identify the condition. However, the actual results are unpredictable. This is where the concept of threadsafe comes into play. A

threadsafe program has the built in mechanism to ensure serialized access to a shared resource to avoid data integrity problems. The responsibility to ensure that a program is threadsafe belongs to the user. Set the CONCURRENCY parameter in the CEDA program definition to THREADSAFE (versus QUASIRENT) to define a program as threadsafe.

It is the user's responsibility to determine whether a program is threadsafe or not. Unfortunately, this is no easy task and a visual review of the programs would probably be in order. IBM provides a utility that can scan the objects looking for possible areas that require attention. This utility allows for the inclusion of other things to verify such as specific user or IBM macros. The utility (DFHEISUP) can be used to scan load modules to identify the use of CICS commands that acquire shared resources. Its use is described in more detail in [Load module scanner: DFHEISUP](#). More information can be found at the following web link <http://www.redbooks.ibm.com/redbooks/SG246351/12-05.htm>. Once this analysis has been done, then the application programs can be marked as threadsafe. It should be noted that not all CICS commands used by application programs are threadsafe. CICS recognizes non-threadsafes commands (EXEC CICS) and takes the appropriate action to avoid any problems. CICS uses a TCB to dispatch work on a processor. There are many different types of TCBs within the CICS region however, many of them are specialized and used infrequently. For example, the File Owning TCB (FO) is only used for opening and closing user files. Once the bulk of the files are opened, this TCB is not used unless another file is to be opened or closed. Tasks are initially dispatched on the QR TCB. Prior to CICS/TS V2, DB2 requests were handled by the DB2 exit (also on the QR TCB) that assigned a DB2 thread (TCB) for execution. Once the request to DB2 completed, the request would have to re-queue to the QR TCB. If there were many outstanding DB2 requests to reconnect to the QR TCB, then the reconnection could result in a wait for dispatch and a delay in the task's response time.

The dispatching mechanism changed with CICS/TS V2 and DB2 V6. The dispatching process is similar to prior releases except for two items:

- Global and TRUE exits marked as threadsafe and OPENAPI are initially dispatched on an L8 TCB and their main function is to assign an L8 TCB to handle the SQL request. The DB2 TRUE exit is dispatched on an L8 TCB so it must be made threadsafe.
- If the application program is defined as threadsafe, then the application program can continue to run on the L8 TCB once the DB2 request completes and does not have to re-queue to the QR TCB.

If the exit is not defined as threadsafe, then a TCB switch from the L8 to the QR must occur in order for the exit to function (Figure 2). This process is called a TCB switch and there is a CPU price to pay for the switch. A TCB switch costs around 2,000 instructions. So, defining the DB2 exit as non-threadsafes is going to cost additional CPU cycles. So, the task was executing on the QR when a DB2 request was made, a switch to an L8 TCB is made at a cost of 2,000

instructions. Since the exit is not coded as threadsafe, then another switch back to the QR TCB is made at a cost of 2,000 instructions. It is interesting that a simple check to see if the exit was threadsafe was not made prior to switching to the L8 TCB. The DB2 exit will assign an L8 TCB to the user task to handle the SQL request and communications with DB2. While the DB2 request is pending, the QR TCB can dispatch other work while the L8 TCB waits for a response from DB2.

```

TRUD ADVANCED COMPUTER TECHNOL  C\TREK ON-LINE      09F8A180  DATE 03/15/2006
APPLID CICSTS31                  APPLICATION DOMAIN  TIME 13:39:54
VERSION 6.4                      USER EXIT DETAIL  TERM 0208

TRUE ADDRESS      09F8A180  DATA LOCATION    BELOW  TIE COUNT      0
EXIT NAME         CTREKSVR  CICS KEY          USER   TIE ANCHOR ADDR 00000000
ENTRY POINT       89B76000  THREADSAFE/QUASI THRDSAFE SECURITY COUNTER 0
LOAD POINT        N/A      DEFINED LANGUAGE  ASSMBLR INTEREST PROFILE 00000004
INSTANCE COUNT    1      DEDUCED LANGUAGE ASSMBLR QUALIF TO TRUE
STARTED?          YES     CPE ADDRESS       09DDB760
ACTIVATION COUNT  1      PROGRAM USE CNT   0
TSK USER AREA LN 0      INSTALL TYPE     GRUPLIST
EXIT FLAG         00     LOAD STATUS       LOADABLE
LOAD MODULE NAME CTREKSVR  PROGRAM AMODE     31
PPT ADDRESS       09DD9C38  PROGRAM RMODE     ANY
DATE              03/10/2006  PROG TOTAL USE    1
TIME              14:46:04.6  CURRENT USERS     1
TRUE PGM MODE     1      TOT # PROG LOADS  1
GLUE PGM MODE     1      TOT # PRG COPIES 1
PURG STATUS       NOTPURGE  GLOBAL WORK AREA  00225154
CEDF ALLOWED?    NO      GLOBAL WAREA LEN  32760
PROG ENABLED?    NO      GLOBAL WAREA USE  1

ENTER REFRESH PF1 HELP PF3 PREV PAGE PF5 MEMORY CLR MAIN MENU
4B :00.1 01/54
    
```

Figure 2. Threadsafes Exit

A round trip from the QR TCB to the L8 TCB and back would represent around 4,000 instructions. The same applies to all exits. Therefore, it is imperative that all exits be converted to threadsafe or the CPU overhead between DB2 requests in CICS/TS V1.3 versus CICS/TS V2 or V3 could be 10% or higher in a heavy DB2 environment. The use of L8 TCBs can have two major advantages for a CICS/DB2 program:

- CPU utilization can be reduced in those cases where more than one consecutive SQL command can be issued without any intervening non-threadsafes command. In other words, if you have a series of consecutive SQL commands, the L8 TCB can be used without having to switch back to the QR TCB as in prior releases. This can represent a savings in CPU cycles
- If the application is coded as threadsafe, then threadsafes application programs can continue to execute on the L8 TCB without having to return to the QR TCB. This represents more parallel processing of application programs on a multi-processing system. The execution of application programs on the L8 TCB

without having to switch back to the QR TCB can result in improved response times because of a reduction of waiting for dispatch timing events

Threadsafe application program execution on the L8 TCB continues until the task terminates or issues a non-threadsafe CICS command. CICS recognizes which commands are not threadsafe and forces a TCB switch back to the QR to process the command. The number of non-threadsafe CICS commands continues to be reduced with each new release but the major function of File Control continues to be non-threadsafe. Once switched, the task will continue to execute on the QR TCB until another SQL command is issued.

It is important to note that that a non-threadsafe application program will switch back to the QR TCB immediately after the SQL request completes even if the next command was another SQL request. This TCB switching can be very expensive CPU wise especially if the DB2 exit is non-threadsafe. It is not unusual to see tasks waiting for dispatch in a heavy non-threadsafe DB2 environment. Therefore, the priority for DB2 shops when converting to CICS/TS V2 or V3 is to first ensure that the exits, in particular the DB2 exit, is threadsafe and that as a second step, the application programs are converted to threadsafe to get the full benefit of the L8 TCB structure. As a side note, it is also important for application programmers to learn the concept of threadsafe programming even if your installation is not planning to convert DB2 programs at this point to threadsafe programming or you do not have DB2. CICS/TS is moving toward providing performance improvement in the response time area by providing more parallel application program execution within the CICS region on multi-processors. This can be seen with the latest announcement of OPENAPI definition for programs in CICS/TS V3R1.

Prior to this announcement, parallel application program execution was limited to DB2 applications. There was no authorized method for a non-DB2 program to use the L\* TCB for execution to gain parallelism. However, programmers found a way to “fool” CICS into providing an L8 TCB for a non-DB2 program. This method was not supported nor official but it worked. In order to facilitate this option for non-DB2 programs, CICS/TS V3R1 provided a way for non-DB2 application programs to be dispatched on an L8 TCB for execution instead of using the QR TCB. This is accomplished by specifying OPENAPI in the API parameter in the CEDA program definition. The program should also be defined as threadsafe. It is important to note that this option should be used with great care because it could represent an increase in CPU utilization by the program.

Through the use of this option, the OPENAPI program is initially dispatched directly on an L8 TCB. The moment that a non-threadsafe command is issued, the request is switched to the QR TCB for execution. However, different from a DB2 switch under this condition where the task remains under the QR TCB until another SQL command is issued, the OPENAPI program is switched back to the L8 TCB upon completion of the request. If there are sufficient non-threadsafe commands in the program, a significant increase in CPU time could result by using this specification. Therefore, you should use OPENAPI for threadsafe programs that have a

minimum or no non-threadsafe CICS commands. It is going to require planning to use OPENAPI. The important point is that as more CICS commands are made threadsafe, more parallel processing can be accomplished by using this option. Thus, understanding threadsafe programming is an important issue of which the application programmers must be aware if additional parallel processing and response time improvements are desired.

This article has provided an overview into the process called TCB switching that is specifically related to CICS/TS V2R2 and above. TCB switching costs around 2,000 instructions per switch. If not handled appropriately by the user through the use of threadsafe exits and application programs, the CPU utilization may increase. More important, the benefits of using an L8 TCBs to simultaneously execute application programs on several processors resulting in improved response times and possibly reduced CPU utilization through the consecutive execution of multiple SQL commands without having to switch TCBs, will not be achieved. Future performance improvements to CICS response time and throughput without processor upgrades will only be achieved through the use of threadsafe programming.

If you have any questions, comments or request for more information, please contact us:

Address: C\TREK Corporation PO BOX 560069 MONTVERDE FL 34756	Phone: (407) 469-3600 (787) 756-5620	C\TREK Corporation Advanced Computer Technology
E-mail: <a href="mailto:ctrek@actpr.com">ctrek@actpr.com</a>	Fax: (787) 756-5150	
Website: <a href="http://www.ctrekcorp.com">www.ctrekcorp.com</a>	Support:: (787) 397-4150 (321) 297-5838 (787) 462-0406	