Virtual CICS user group: Newsletter 29

Welcome to the Virtual CICS user group newsletter. The Virtual CICS user group at www.fundi.com/virtualcics is an independently-operated vendor-neutral site run by and for the CICS user community.

Virtual CICS user group presentation

The latest webinar from the Virtual CICS user group was entitled, "Using Threadsafe in CICS", and was presented by Colin Pearce.

Colin is a very experienced MVS/CICS Consultant and has 25 years of MVS/CICS systems experience. He has designed and conducted many MVS/CICS courses in many countries around the world. He has worked with different Corporations from Banking and Manufacturing to Retail and Software Developers. He has worked in Australia, Singapore, and now resides in the UK where he is working with Lloyds Bank as a CICS Consultant and Assembler Programmer.

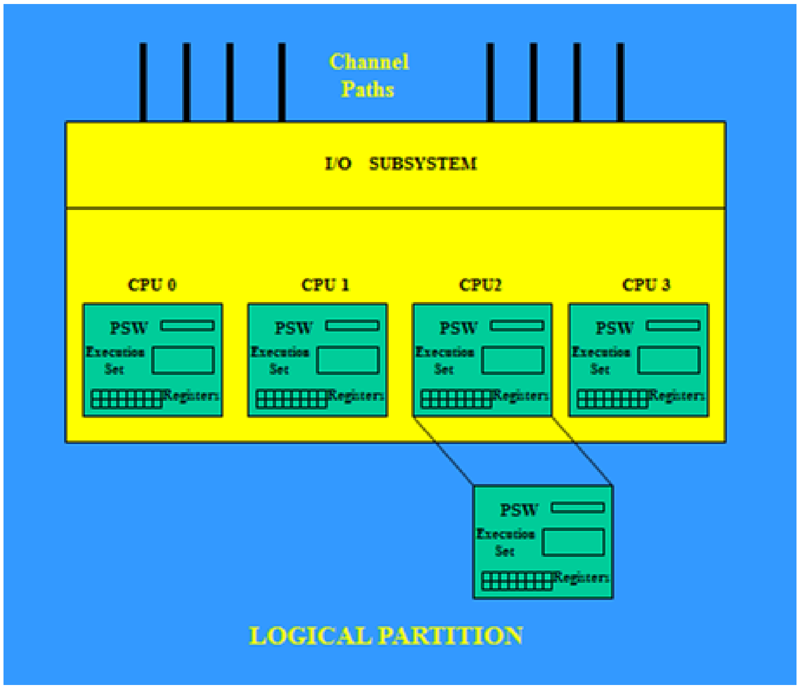
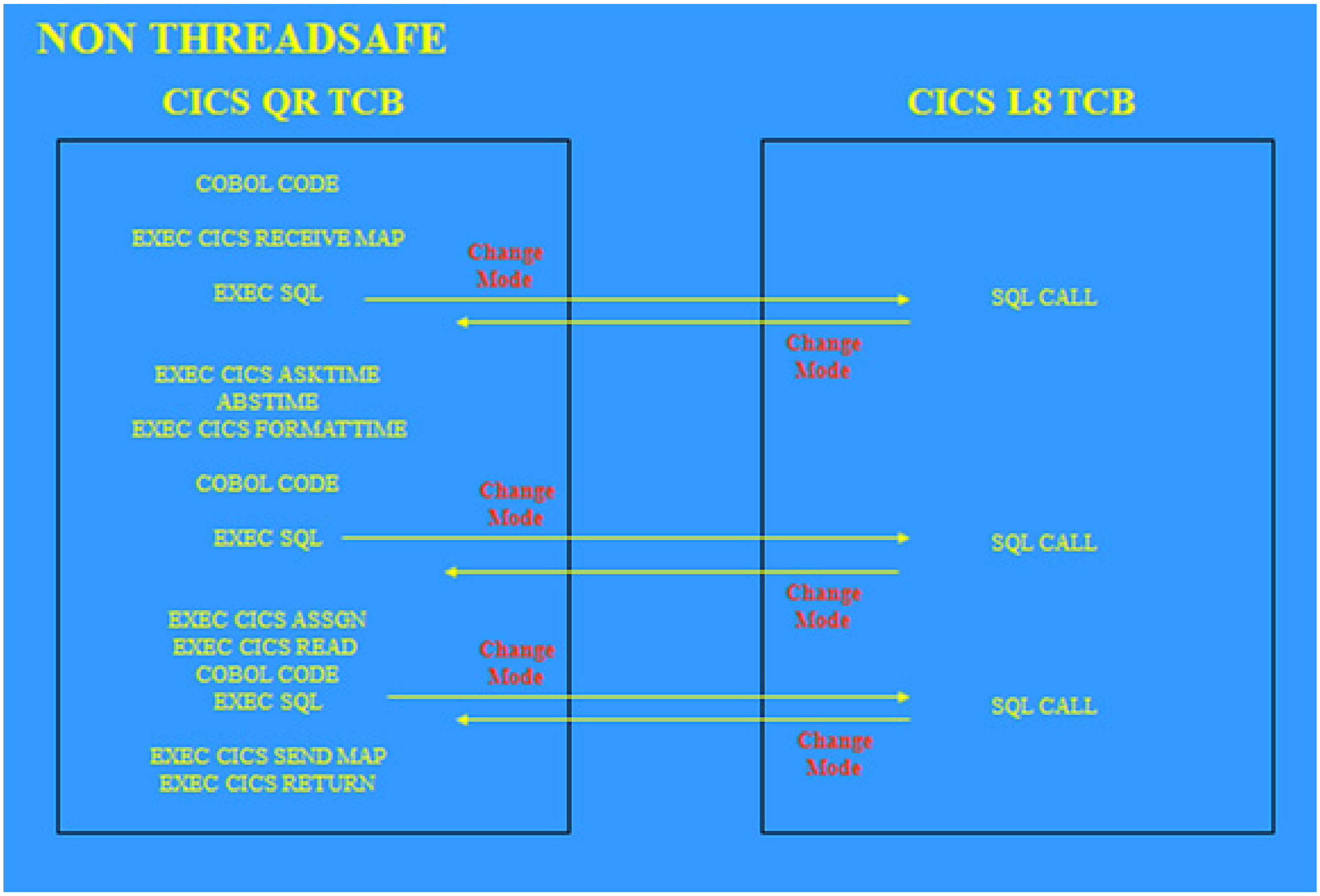


Figure 1: Understanding threadsafe

Colin started his presentation by explaining that each CPU in an LPAR can run a separate piece of work under the main TCB. z/OS only knows about CICS by its main task TCB. However, a main task TCB can attach sub-task TCBs, and so multiple work can be done when the main task is given control/dispatched by z/OS (see Figure 1). CICS subtask TCBs are:

Contents:



NON THREADSAFE

CICS QR TCB

CICS L8 TCB

COBOL CODE

EXEC CICS RECEIVE MAP

Change Mode

EXEC SQL

SQL CALL

Change Mode

EXEC CICS ASKTIME
ABSTIME
EXEC CICS FORMATTIME

COBOL CODE

Change Mode

EXEC SQL

SQL CALL

Change Mode

EXEC CICS ASSGN
EXEC CICS READ
COBOL CODE
EXEC SQL

Change Mode

SQL CALL

Change Mode

EXEC CICS SEND MAP
EXEC CICS RETURN

Figure 2: Non-threadsafe environment

- QR – this is the Quasi Reentrant TCB. There is only one of these and its function is to execute all the CICS API commands and the COBOL code.

- RO – this is the Resource Owning TCB. There is only one of these and its function is to locate and load the application programs into CICS storage.

- L8 – this is the Open TCB for DB2 SQL calls/ MQ calls. This is used whether or not storage protection is enabled. There can be any number of these as defined by MAXOPENTCBS defined in the System Initialization Table.

- L9 – this is an Open TCB for non-DB2 calls. This is used with storage protection enabled and the EXEC KEY of the program is USERKEY. There can be any number of these as defined by the MAXOPENTCBS defined in the System Initialization Table.

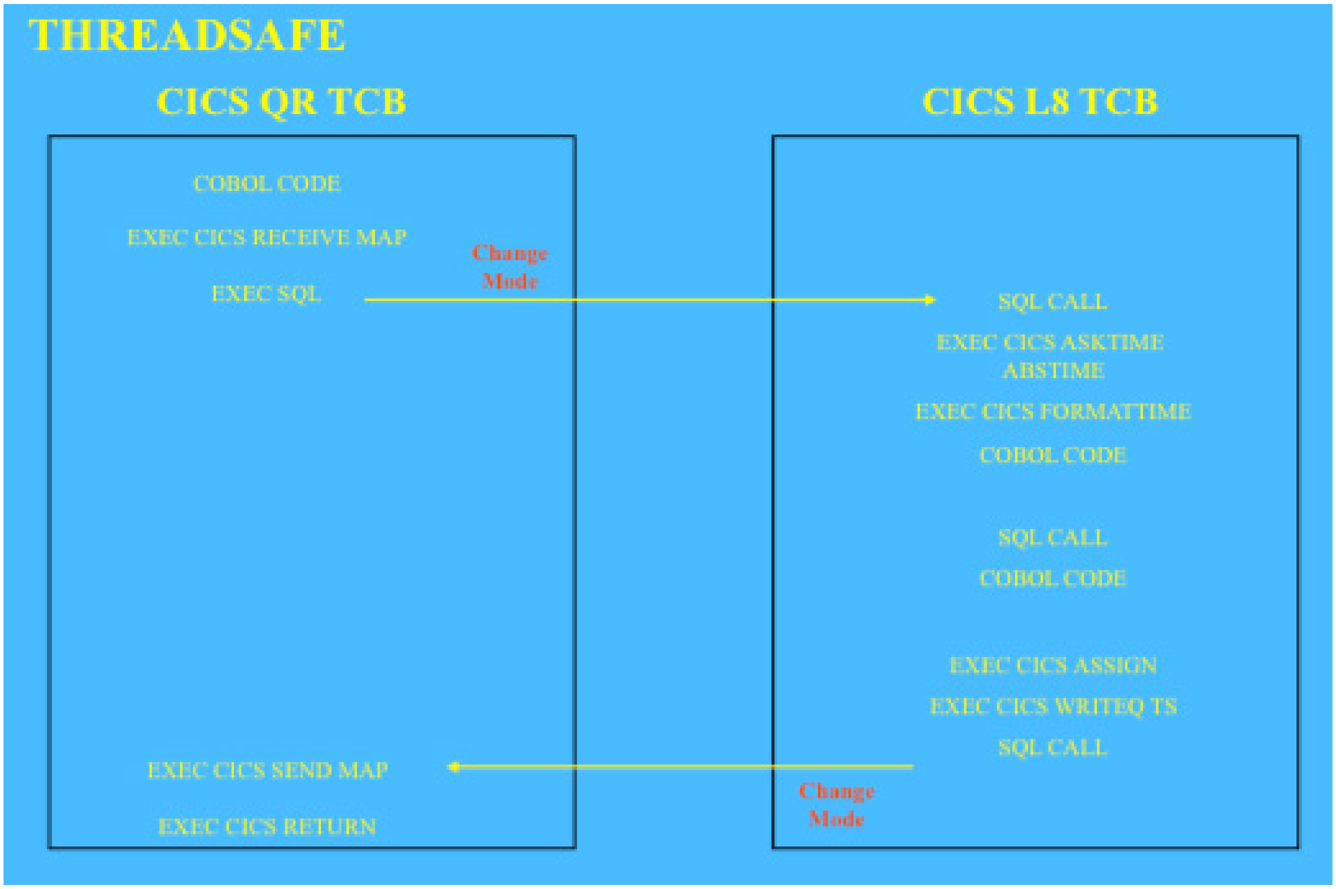
Depending on how an application is designed, defining it as threadsafe can significantly reduce the path length of application tasks. The transactions that will achieve the greatest CPU reduction are likely to be DB2 applications that have the following characteristics:

- A significant number of EXEC SQL calls are invoked per task.

- All programs invoked between the first and last EXEC SQL in each task are defined as threadsafe.

- All exits invoked as part of an EXEC SQL call are defined as threadsafe, and contain threadsafe EXEC CICS commands.

- All exits invoked between the first and last EXEC



THREADSAFE

CICS QR TCB

COBOL CODE

EXEC CICS RECEIVE MAP

EXEC SQL

Change Mode

CICS L8 TCB

SQL CALL

EXEC CICS ASKTIME ABSTIME

EXEC CICS FORMATTIME

COBOL CODE

SQL CALL

COBOL CODE

EXEC CICS ASSIGN

EXEC CICS WRITEQ TS

SQL CALL

Change Mode

EXEC CICS SEND MAP

EXEC CICS RETURN

Figure 3: Threadsafe environment

SQL call in each task are defined as threadsafe.

- All EXEC CICS statements invoked between the first and last EXEC SQL call in each task are threadsafe.

Defining transactions with the preceding characteristics as threadsafe will all but eliminate TCB switches for the associated CICS tasks. This switching of TCBs can be easily identified in a CICS Trace as a ‘Change Mode’, providing the Dispatcher Trace option is enabled.

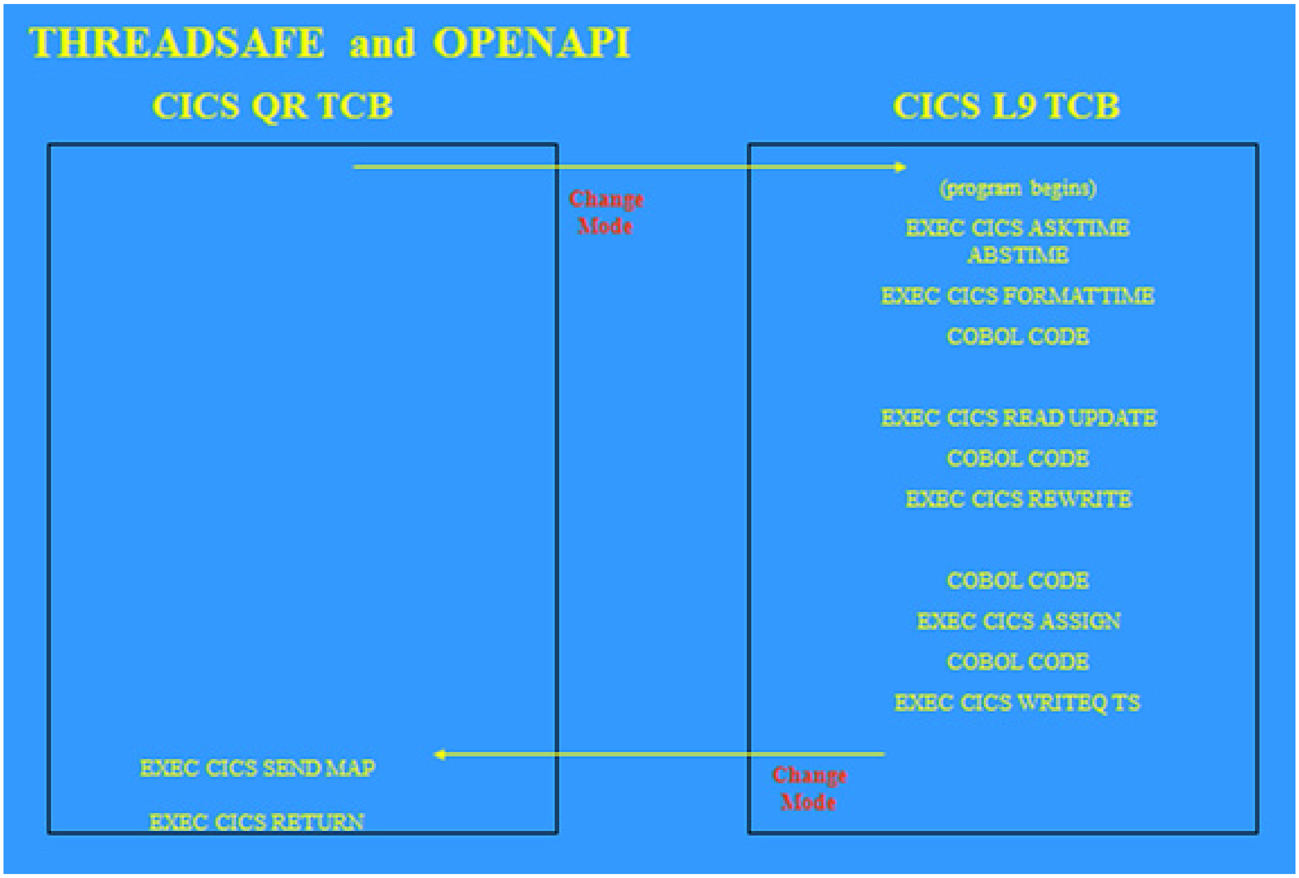
Each TCB is added to a TCB queue, so that when CICS is dispatched by z/OS, a search down the TCB queue will determine which TCB goes first. This TCB queue can be manipulated as to how these TCBs are prioritized in that the QR TCB can be higher, equal, or lower in priority order than the L8 TCBs.

By default each L8 TCB will wait for the QR TCB to become available, when a CICS API call is encountered.

MAXOPENTCBs is the total number of Open TCBs L8/L9 that can be created/utilized. The more Open TCBs, the longer it can take z/OS to run down the TCB queue looking for a TCB ready to go.

Within CICS, all transactions execute on the Quasi-Reentrant TCB. There is only one of them, which means that while a transaction has control, no other transactions can run, until this transaction gives up control, and causes a return to the CICS Dispatcher. Any I/O will cause this. Not all CICS API commands cause the transaction to be interrupted.

CICS can run multiple TCBs to maximize throughput. The Open TCB process was created to reduce the demand for the QR TCB.



THREADSAFE and OPENAPI

CICS QR TCB

CICS L9 TCB

Change Mode

(program begins)

EXEC CICS ASKTIME ABSTIME

EXEC CICS FORMATTIME

COBOL CODE

EXEC CICS READ UPDATE

COBOL CODE

EXEC CICS REWRITE

COBOL CODE

EXEC CICS ASSIGN

COBOL CODE

EXEC CICS WRITEQ TS

Change Mode

EXEC CICS SEND MAP

EXEC CICS RETURN

Figure 4: Threadsafe and OPENAPI environment

For all DB2 calls, the L8 Open TCB was created in order to run SQL calls on it, and so free up the QR TCB. Because the CICS/DB2 exit is defined as CICSKey, it will cause all DB2 calls to run on the L8 TCB, including any CICS API commands. However, if the application has no DB2 calls, and has only VSAM calls and is defined as UserKey (the default with Storage protection turned on), this program can run on the L9 TCB with threadsafe enabled.

This means that in a multi-CPU environment, the more TCBs that can be run under the address space (the main TCB), the more work can be done and so throughput is increased.

Threadsafe was introduced to allow most of the CICS API commands and COBOL code to execute on the Open TCB. There can be many L8/L9 TCBs concurrently and the MAXOPENTCBs value controls this number.

It makes no sense to run on the L8 TCB issuing SQL calls and to switch back to the QR TCB just to get the date and time. Figure 2 shows the normal default setup in a non-threadsafe CICS/DB2 environment. The EXEC SQL call will cause the switch from the QR TCB to the L8 TCB. A switch back to the QR TCB will occur at the completion of the SQL call.

This command would execute on the QR TCB. However, it must queue for this TCB, because as there is only one, it is likely that it is already occupied by a current transaction's activity. Once the QR TCB has been obtained, control will stay there (and of course stop other transactions from using it), until the next EXEC SQL command, which will cause a switch back to the L8 TCB.

Too much switching between the QR and L8 TCB adds considerable overhead to the overall Performance. It has been estimated as 2000 instructions per one way switch. Therefore the more SQL calls, inter-mixed with CICS API calls, in a non-threadsafe environment, the more switching and the more instructions are used to support this.

Figure 3 shows that more switches between L8 TCBs and the QR TCB are done because the application has been defined as non-threadsafe (the default).

Remember there may be many L8 TCBs in progress, each one assigned a thread to access DB2, each one will queue for the QR TCB when the SQL call has completed.

The switch back to the L8 TCB will only occur on the next EXEC SQL call. It is estimated that one switch back and forth, has some 4000 instructions.

With threadsafe, the program can stay on the L8 TCB, and continue to execute both threadsafe CICS API commands and COBOL code. While executing on the L8 TCB, if CICS recognizes any API command that is not threadsafe, such as a terminal control access or Transient Data request (pre-V5), then it will switch back to the QR TCB automatically.

MVS dispatches TCBs within the address space.

In Figure 4, the program has been defined as threadsafe and its API option specifies OPENAPI.

In this instance the program begins on the Open TCB. In this example, the program follows its EXEC Key definition. If it is specified as user key, then the program will commence on the L9 TCB.

It will stay there for the duration of the program, thus relieving the demand and load on the QR TCB.

Of course, if there is a DB2 call, this will cause migration to the L8 TCB, and if there is an non-threadsafe command encountered, CICS will automatically migrate back to the QR TCB to process it.

Because the program is specified as OPENAPI, CICS will migrate back to the L9 TCB before handing control back to the program.

Each successive release of CICS Transaction Server has increased the number of CICS API commands that have been modified to run on the Open TCB.

CICS/TS 3.2 extended this support to all File Control commands. CICS/TS V4 extended this VSAM support to the mirror task for remote files as long as the connection is IP. CICS/TS V5 added Transient Data API commands to threadsafe

RLS is now included in this support.

This means that with Storage Protection enabled, for all programs that have only DB2 calls, the L8 TCB will be used. For all programs that have DB2 calls, the L8 TCB will be used. For the 2 previous examples, the L8 TCB is used because the CICS/DB2 attach exit is defined as CICSKey. For all VSAM calls and no DB2 calls, then the L9 TCB will be used, because the program will be defined as UserKey.

The new CONCURRENCY (REQUIRED) option means that that program will be moved to an OPEN TCB once is has finished the current call on the QR TCB, before continuing. So it is the KEY that defines which Open TCB will be used.

In order to implement threadsafe, the application must be reviewed, in order to identify any API commands or code that would not make the program threadsafe and compromise the integrity of the application. Commands such as: EXEC CICS

GETMAIN SHARED; EXEC CICS ADDRESS(CWA); and COBOL Calls (Assembler) (ensure all save areas are in their own getmain storage).

Colin talked at some length about the steps that need to be taken to implement threadsafe in CICS.

A copy of Colin Pearce's presentation is available for download from the Virtual CICS user group Web site at www.fundi.com/virtualcics/presentations/ThreadsafeNov15.pdf.

You can see and hear the whole user group meeting by downloading the WMV file from www.fundi.com/virtualcics/presentations/2015-11-10meeting.wmv.

Meeting dates

The following meeting dates have been arranged for the Virtual CICS user group:

- On 12 January 2016, we have IBM's David Lawrence talking about "Creating an Agile development environment while reducing costs".
- The following meeting is on 8 March 2016.

We will be using Citrix GoToMeeting for the user group meetings.

Recent CICS articles

Opening the Lid on CICS Channels and Containers by Pete Dally, Brenda Hawkins, and Andy Wright in the August/September 2015 edition of *Enterprise Tech Journal.* You can find the article at http://ourdigitalmags.com/publication/?i=272568&p=16.

CICS news

IBM has announced CICS Transaction Server for z/OS V5.3. New and enhanced capabilities are delivered in three main areas:

- Service agility – focuses on enhanced support for Java and WebSphere Liberty profile;
- Operational efficiency – includes performance optimizations, enhanced metrics, and additional security; and
- Cloud with DevOps – introduces new cloud and DevOps support to automate CICS deployments.

Full details can be found at http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=an&subtype=ca&supplier=897&letternum=ENUS215-363.

About the Virtual CICS user group

The Virtual CICS user group was established as a way for individuals using IBM's CICS TS systems to exchange information, learn new techniques, and advance their skills with the product.

The Web site at www.fundi.com/virtualcics provides a central point for coordinating periodic meetings (which contain technically-oriented topics presented in a webinar format), and provides articles, discussions, links, and other resources of interest to IBM CICS practitioners. Anyone with an interest in CICS is welcome to join the Virtual CICS user group and share in the knowledge exchange.

To share ideas, and for further information, contact trevor@itech-ed.com.

The Virtual CICS user group is free to its members.