Threadsafe Conversation Techniques for CICS Applications

Noel Javier C. Sales

Software Engineer zCICS Development

IBM CICS® User Conference 2009

Terminology

- Quasi-reentrant (QR) TCB
  - The main CICS TCB under which all application code runs prior to OTE
  - CICS dispatcher subdispatches work, so each CICS task has a slice of the action
  - A CICS task gives up control via a CICS dispatcher wait
  - Only one CICS user task is active at any one time
- Quasi-reentrant programs
  - Same program can be invoked by more than one CICS task
    - But only one CICS task is active at any one time
  - Quasi-reentrancy allows programs to share virtual storage e.g. CWA without the need to protect against concurrent update
  - CICS code takes advantage of quasi-reentrancy, e.g. can avoid locking if code always runs on QR.

Terminology

- Open TCBs
  - A new class of CICS TCB available for use by applications
  - Each TCB is for the sole use of the owning CICS task but can be reused by a later task.
    - No subdispatching under Open TCBs, blocking by applications allowed
  - There are several different types or modes of Open TCB.
  - CICS dispatcher domain manages a pool of TCBs for each mode
  - CICS will switch between an Open TCB and the QR TCB as required

Terminology

- Threadsafe programs
  - Are capable of being invoked on multiple TCBs concurrently
  - Cannot rely on quasi-reentrancy to serialise access to resources and storage
  - A threadsafe program is one that does not modify any area of storage that can be modified by any other program at the same time and it does not depend on any area of shared storage remaining consistent between machine instrucutions.
  - Must use serialisation techniques such as compare and swap (CS) or enqueue/dequeue to access shared resources with integrity
  - All programs accessing a shared resource must be made threadsafe e.g. an existing program's reliance on quasi-reentrancy to serialise accessto the CWA is made invalid if just one other program can run concurrently on another TCB and access the same CWA field.

CICS Commands and Threadsafe programs

- Threadsafe CICS commands can run on QR or an open TCB
- Non Threadsafe CICS commands must run on QR TCB
- A threadsafe program can issue both threadsafe and non threadsafe CICS commands
  - A Threadsafe command will not cause a TCB switch
  - A non threadsafe command will cause a switch back to QR if not already on QR
    - There is no data integrity issue, just a performance hit of a TCB switch
- However….a program defined to CICS as threadsafe when its application logic is not compromises the data integrity of its resources eg shared storage (no risk to CICS resources)

Do not neglect Global User Exits
(they will come back to bite you !)

- GLUEs should be made THREADSAFE else excessive TCB switching will occur
- CICS will switch to QR TCB from an open TCB to invoke a non threadsafe GLUE and back again afterwards
- Particularly important exits that need to be threadsafe are XRMIIN and XRMIOUT on the RMI path, and XEIIN/XEIOUT called for all CICS commands.
  - Check with vendors for threadsafe support
  - Use DFH0STAT to find out whats exits in use and if they are defined threadsafe
- Use XPI ENQUEUE and DEQUEUE to serialise access to shared areas like the Global work area (GWA).
- **Warning:** You can no longer rely on field CSACDTA to access the TCA of the current CICS task - only works when running under QR TCB. In CICS TS 3.2 use of CSACDTA produces an ASRD abend.

Steps to take: Ensure the program is read-only

- **DFHSIT Parameter RENTPGM=PROTECT**
  - Specifies you want CICS to allocate the read-only DSAs (RDSA and ERDSA) from read-only key-0 protected storage
    - RDSA for RMODE(24) programs and the ERDSA for RMODE(ANY) programs linked RENT
  - Any attempt to overwrite the program will result in message DFHSR0622 and an ABEND0C4
    - Programs running in key-zero or supervisor state can still overlay RDSA and ERDSA

```
DFHSR0622 An attempt to overwrite the RDSA has caused the abend
          which follows

DFHAP0001 An abend (code 0C4/AKEA) has occurred at offset X'00000ACC'
          in module SQLSPIN.
```

Example: Customer Threadsafe Problem One

- **CICS Region terminated with DFHKE1800**
  - Prior ABEND0C1 in user program SQLSPIN
- **Transaction SQLS in program SQLSPIN suffered the original abend**
- **SQLSPIN is a COBOL program defined as Threadsafe**
- **SQLSPIN Issued DB2 Select command and called Assembler program ASMSETR**

Customer Threadsafe Problem One – COBOL Program SQLSPIN

```
           MOVE 1 TO COUNTER.
           PERFORM WITH TEST BEFORE
               UNTIL COUNTER > 20
                   EXEC SQL
                       SELECT * INTO :VVEMP FROM DSN8710.EMP
                       WHERE EMPNO = '000990'
                   END-EXEC

                   CALL 'ASMSETR' END-CALL
                   ADD 1 TO COUNTER
           END-PERFORM.
           MOVE 'TRANSACTION COMPLETED.'  TO LOGMSG
           EXEC CICS SEND FROM(LOGMSG) LENGTH(22) ERASE END-EXEC.

       END-MAIN.
```

Note: Static and Dynamic Called programs will run on the current TCB
ASMSETR is a Static Called program and will run on the L8 TCB

Customer Threadsafe problem one – assembler program ASMSETR

```
ASMSETR  CSECT
         USING *,15
RSA      DC    16F'00'            REGISTER SAVE AREA
BEGIN    DS    0H
         SAVE  (14,12)            SAVE R14 Through R12
         ST    13,RSA+4           STANDARD SAVE AREA FOR R13 RECEIVED
         LA    14,RSA             POINT R14 AT SAVE AREA WITHIN PROGRAM
         ST    14,8(,13)          POST FORWARD RSA POINTER
         LR    13,14              POINT R13 TO CURRENT RSA
         LR    3,15               SET BASE REGISTER 3
         USING ASMSETR,3
*
* Business logic was here.
*
         L     13,4(13)
         RETURN (14,12),T,RC=0    RETURN TO SQLSPIN
FILLER   DC    C'XXXXXXXX'
         END   ASMSETR
```

Customer Threadsafe Problem One – Detection and Prevention

- **Detection**
  - This problem would have been detected if the load module containing the COBOL and Assembler program had been Link-Edited with the RENT attribute and RENTPGM=PROTECT used
    - Since ASMSETR stores within itself you would receive an ABEND0C4 when trying to update the Read Only DSA
- **Prevention**
  - Translate ASMSETR, place RSA in dynamic storage DFHEISTG
  - Or change the call to ASMSETR to use EXEC CICS LINK and ensure ASMSETR defined as Quasirent
    - Not recommended, will cause lots of TCB switching

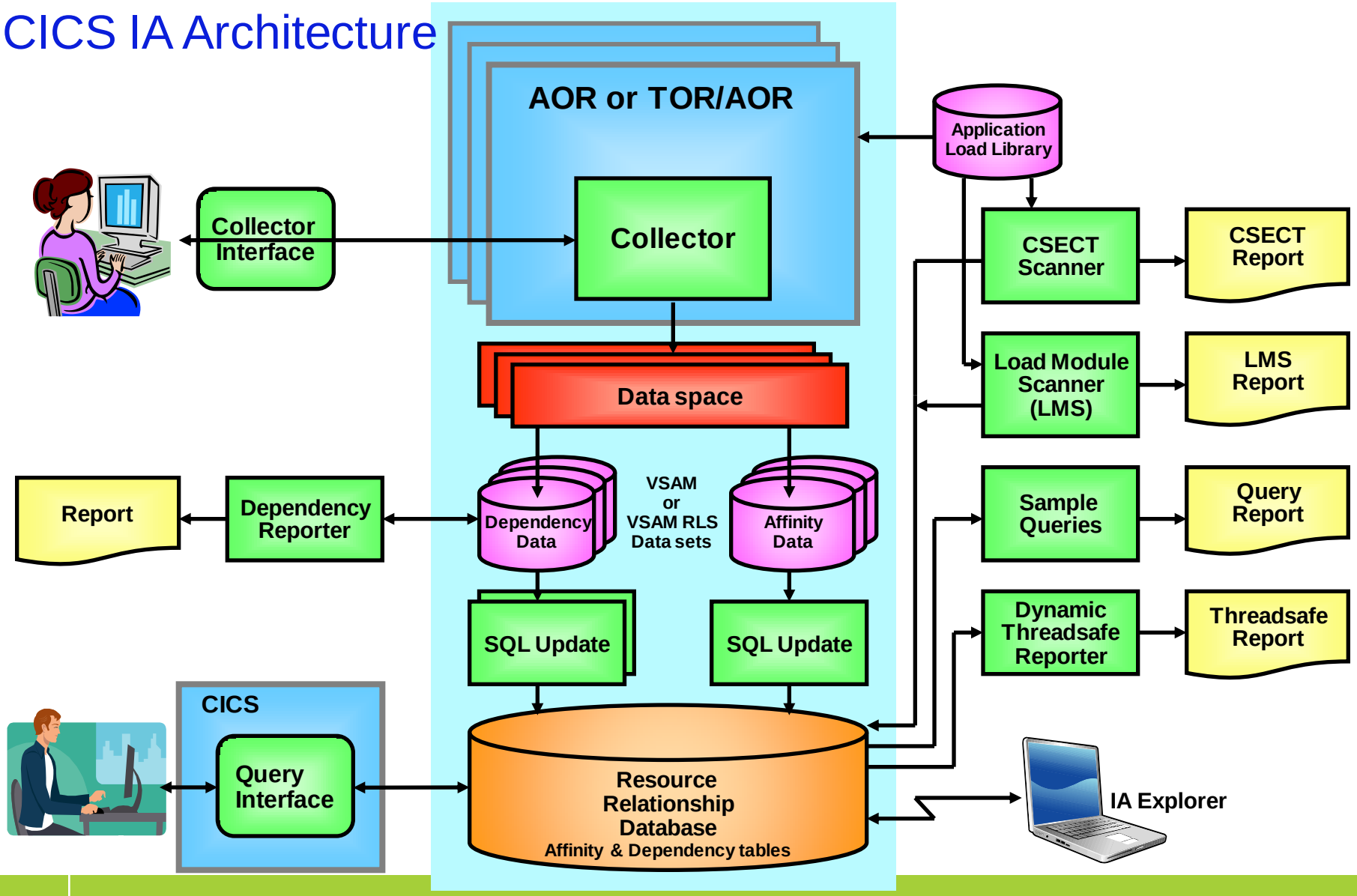
Things to remember about Static and Dynamic Calls

- A statically called program has no CICS program definition of its own
  - Called on whatever TCB the calling program is running on
  - Called programs's logic needs to be threadsafe if calling program is threadsafe
- A Cobol dynamically called program has a program definition
  - But it is only used to allow LE to load the program.
  - Called program is branch and linked to on the callers TCB
  - Called program's logic needs to be threadsafe if calling program is threadsafe
- When a DB2 call is issued from a static or dynamically called routine, the CONCURRENCY attribute of the calling program is used after the DB2 call completes.
  - CICS has no knowledge of the called routine
  - CICS believes current program is the calling program

Steps to take: Look for use of shared storage with CICS IA

- Typical examples of shared storage are …
  - CWA
  - Global user exit global work areas
  - Storage acquired explicitly by the application program with the GETMAIN SHARED option
- You can check whether your application programs use these types of shared storage by looking for occurrences of the following EXEC CICS commands …
  - ADDRESS CWA
  - EXTRACT EXIT GASET
  - GETMAIN SHARED
  - LOAD PROGRAM (eg using dummy assembler program as shared storage)

CICS IA Architecture



Application performance support

- **Results from sample query against the V_CIU_SCAN_TRDSAFE view.**

```
-- q1 - Show all programs with non-threadsafe EXEC CICS commands
**INPUT STATEMENT:
SELECT 'TS31 non-threadsafe calls for ',
      DSNAME, "PROGRAM", OFFSET, COMMAND, ' = ', COUNT(*)
    FROM  V_CIU_SCAN_TRDSAFE
    WHERE (CICS_TS31 = 'N' OR CICS_TS31 IS NULL)
    GROUP BY DSNAME, "PROGRAM", OFFSET, COMMAND;
      +-------------------------------------------------------------------------------------------------------------------+
      |                                |           DSNAME           | PROGRAM  |    OFFSET    | COMMAND  |     |      |
      +-------------------------------------------------------------------------------------------------------------------+
    1_| TS31 non-threadsafe calls for  | CICSIAD.IA2.TEST.LOADLIB   | AFFTESTZ |          685 | INQUIRE  |  =  |    1 |
    2_| TS31 non-threadsafe calls for  | CICSIAD.IA2.TEST.LOADLIB   | AFFTESTZ |          706 | INQUIRE  |  =  |    1 |
    3_| TS31 non-threadsafe calls for  | CICSIAD.IA2.TEST.LOADLIB   | AFFTESTZ |          727 | INQUIRE  |  =  |    1 |
    4_| TS31 non-threadsafe calls for  | CICSIAD.IA2.TEST.LOADLIB   | AFFTESTZ |          748 | DISCARD  |  =  |    1 |
    5_| TS31 non-threadsafe calls for  | CICSIAD.IA2.TEST.LOADLIB   | AFFTESTZ |          769 | SET      |  =  |    1 |
    6_| TS31 non-threadsafe calls for  | CICSIAD.IA2.TEST.LOADLIB   | AFFTESTZ |          790 | INQUIRE  |  =  |    1 |
      +-------------------------------------------------------------------------------------------------------------------+
                                       . . .
```

Threadsafe Dynamic Analysis report

```
1CICS INTERDEPENDENCY ANALYZER VERSION 2.2.0                                  2007/10/03:14.01.34   PAGE     1
 Program Dynamic Analysis - THREADSAFE DETAIL LISTING FOR CICS TS

 Report options:
   PROGRAMNAME=*        REGIONNAME=*        CICSLEVEL=      REPORT=DETAIL   LINESPERPAGE=60


 Definitions of Terms:

   'Threadsafe' calls are EXEC CALLS commands that do not cause a TCB swap.

   'Non-Threadsafe' calls are EXEC CALLS commands that cause a TCB swap.

   'Indeterminate Threadsafe' calls are EXEC CALLS commands where it cannot be determined if the call causes a TCB swap.

   'Dynamic calls' are calls to modules at execution time. Programs that are called dynamically take on the same environment
                  as the calling program.

   'Threadsafe Inhibitor calls' are EXEC CICS commands that need to be investigated further because they may prevent you from
                               defining your program as threadsafe. These commands are: ADDRESS CWA, EXTRACT EXIT, GETMAIN SHARED,
                               and LOAD.
```

Threadsafe Dynamic Analysis report - Summary

```
CICS INTERDEPENDENCY ANALYZER VERSION 2.2.0                                    2007/10/19:11.50.59    PAGE     3
Program Dynamic Analysis - THREADSAFE SUMMARY LISTING FOR CICS TS 3.2


APPLID   Program   Linkedit   Execution  Concurrency  APIST  Storage  CICS  LIB Dataset Name
                     Date        Key                        Protect  Rel
-------- -------- ---------- ---------- ---------- ------- -------- ----- --------------------------------

IYDZZ328 EMSCONTA ---------- USER       QUASIRENT   CICSAPI ACTIVE   0650  CICSIAD.TEST.LOADLIB
  Total CICS calls:        13  Threadsafe:           1  Non-Threadsafe:                  7  Indeterminate Threadsafe:        0
                               DB2 calls:            0  MQ calls:                        0  IMS calls:                       0
                               Dynamic Calls:        0  Threadsafe Inhibitor calls:      0

IYDZZ328 EMSTESTS ---------- USER       QUASIRENT   CICSAPI ACTIVE   0650  CICSIAD.TEST.LOADLIB
  Total CICS calls:        64  Threadsafe:          34  Non-Threadsafe:                 26  Indeterminate Threadsafe:        0
                               DB2 calls:            0  MQ calls:                        0  IMS calls:                       0
                               Dynamic Calls:        0  Threadsafe Inhibitor calls:      5
```

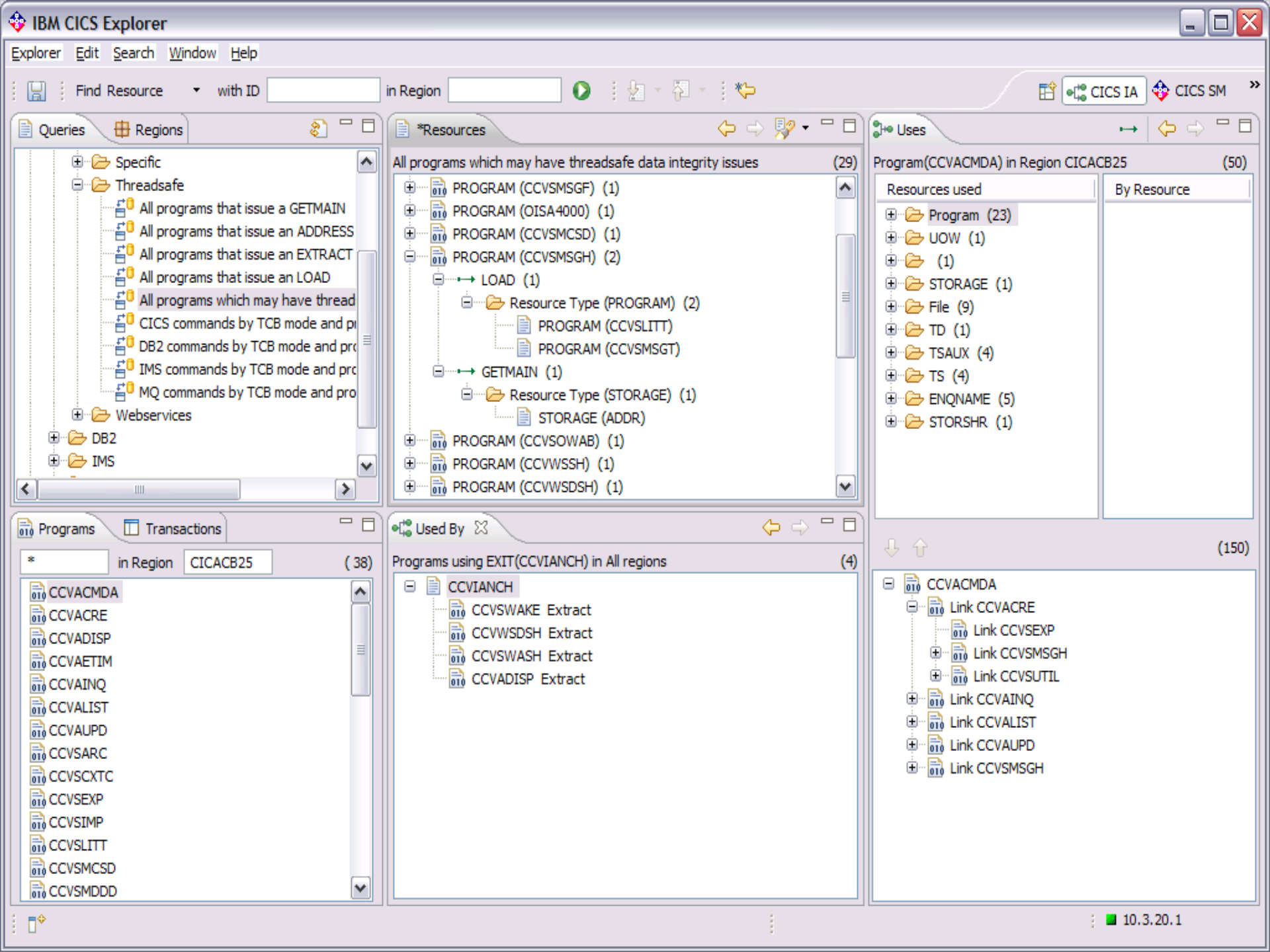
Threadsafe Dynamic Analysis report

```
CICS INTERDEPENDENCY ANALYZER VERSION 2.2.0                                  2007/10/19:12.22.05   PAGE     3
Program Dynamic Analysis - THREADSAFE DETAIL LISTING FOR CICS TS 3.2

APPLID   Program   Linkedit   Execution  Concurrency  APIST  Storage  CICS  LIB Dataset Name
                     Date         Key                       Protect  Rel
-------- -------- ---------- ---------- ---------- ------- -------- ----- --------------------------------

                 CMD  Function              Type       Resource                       Offset   Program  Use      Threadsafe
                 Type                                                                          Length   Count
                 ---- ---------------------- ---------- ------------------------------ -------- -------- --------- ----------

IYDZZ328 EMSTESTS ---------- USER       QUASIRENT   CICSAPI ACTIVE   0650  CICSIAD.TEST.LOADLIB
                 CICS ADDRESS                        CWA                             E76     3380        1     Y *
                 .                                   .
                 CICS ADDRESS                        TCTUA                           E76     3380        1     Y
                 CICS ADDRESS                        TCTUA                          1152     3380        1     Y
                 .                                   .
                 .                                   .
                 CICS DEFINE                COUNTER  TESTDCOUNTER                   1BE0     3380        1     N
                 CICS DELETE                COUNTER  TESTCOUNTER                    1F2C     3380        1     N
                 CICS DELETE                COUNTER  TESTDCOUNTER                   1F78     3380        1     N
 Total CICS calls:        64  Threadsafe:          34  Non-Threadsafe:              26  Indeterminate Threadsafe:
                              DB2 calls:            0  MQ calls:                     0  IMS calls:
                              Dynamic Calls:        0  Threadsafe Inhibitor calls (*):    5
```


IBM CICS Explorer
Explorer Edit Search Window Help
Find Resource with ID in Region
CICS IA
CICS SM
Queries
Regions
Specific
Threadsafe
All programs that issue a GETMAIN
All programs that issue an ADDRESS
All programs that issue an EXTRACT
All programs that issue an LOAD
All programs which may have thread
CICS commands by TCB mode and pr
DB2 commands by TCB mode and pro
IMS commands by TCB mode and pro
MQ commands by TCB mode and pro
Webservices
DB2
IMS
*Resources
All programs which may have threadsafe data integrity issues (29)
PROGRAM (CCVSMSGF) (1)
PROGRAM (OISA4000) (1)
PROGRAM (CCVSMCSD) (1)
PROGRAM (CCVSMSGH) (2)
LOAD (1)
Resource Type (PROGRAM) (2)
PROGRAM (CCVSLITT)
PROGRAM (CCVSMSGT)
GETMAIN (1)
Resource Type (STORAGE) (1)
STORAGE (ADDR)
PROGRAM (CCVSOWAB) (1)
PROGRAM (CCVWSSH) (1)
PROGRAM (CCVWSDSH) (1)
Uses
Program(CCVACMDA) in Region CICACB25 (50)
Resources used
By Resource
Program (23)
UOW (1)
(1)
STORAGE (1)
File (9)
TD (1)
TSAUX (4)
TS (4)
ENQNAME (5)
STORSHR (1)
Programs
Transactions
* in Region CICACB25 (38)
CCVACMDA
CCVACRE
CCVADISP
CCVAETIM
CCVAINQ
CCVALIST
CCVAUPD
CCVSARC
CCVSCXTC
CCVSEXP
CCVSIMP
CCVSLITT
CCVSMCSD
CCVSMDDD
Used By
Programs using EXIT(CCVIANCH) in All regions (4)
CCVIANCH
CCVSWAKE Extract
CCVWSDSH Extract
CCVSWASH Extract
CCVADISP Extract
(150)
CCVACMDA
Link CCVACRE
Link CCVSEXP
Link CCVSMSGH
Link CCVSUTIL
Link CCVAINQ
Link CCVALIST
Link CCVAUPD
Link CCVSMSGH
10.3.20.1

Problem 2 – Non serialized update of the CWA

```
DFHEISTG DSECT
         EXEC SQL INCLUDE SQLCA
EMPNUM    DS   CL6
BONUSLNG  DS   0CL8
BONUSFIL  DS   CL5
BONUSPAK  DS   CL3
CWAREG   EQU  9
SQDWSREG EQU  8
SQDWSTOR DS   (SQLDLEN)C     RESERVE STORAGE TO BE USED FOR SQLDSECT
CWAMAP   DSECT
BONUSTOT DS     XL4
SQLASM1  CSECT
*
         MVC   EMPNUM,=C'000140'
*
* SQL WORKING STORAGE
         LA    SQDWSREG,SQDWSTOR   GET ADDRESS OF SQLDSECT
         USING SQLDSECT,SQDWSREG   AND TELL ASSEMBLER ABOUT IT
*
         EXEC SQL                                                  *
               DECLARE DSN8710.EMP TABLE (                         *
               EMPNO                     CHAR(6),                  *
               SALARY                    DECIMAL,                  *
               BONUS                     DECIMAL,                  *
*
 EXEC SQL SELECT BONUS INTO :BONUS FROM DSN8710.EMP WHERE           *
                 EMPNO= :EMPNUM
*
         XC    BONUSLNG,BONUSLNG
         MVC   BONUSPAK,BONUS
*
         EXEC CICS ADDRESS CWA(CWAREG)
         USING CWAMAP,CWAREG
*
NEWTOTAL EQU   6
*
         CVB   NEWTOTAL,BONUSLNG    Get current bonus in register.
         A     NEWTOTAL,BONUSTOT    Add to old bonus total.
         ST    NEWTOTAL,BONUSTOT    Update CWA with new total.
*
         EXEC CICS RETURN
         END
```

Threadsafe Problem Two – Detection and Prevention

- **Detection**
  - CICS IA would have shown the use of EXEC CICS ADDRESS(CWA)
- **Prevention**
  - EXEC CICS ENQUEUE and DEQUEUE around the update to the CWA
  - Compare and Swap (CS) or Compare Double and Swap (CDS) for Assembler programs
  - Test and Set (TS) instruction for Assembler programs
  - Change RDO Program definition to Quasirent

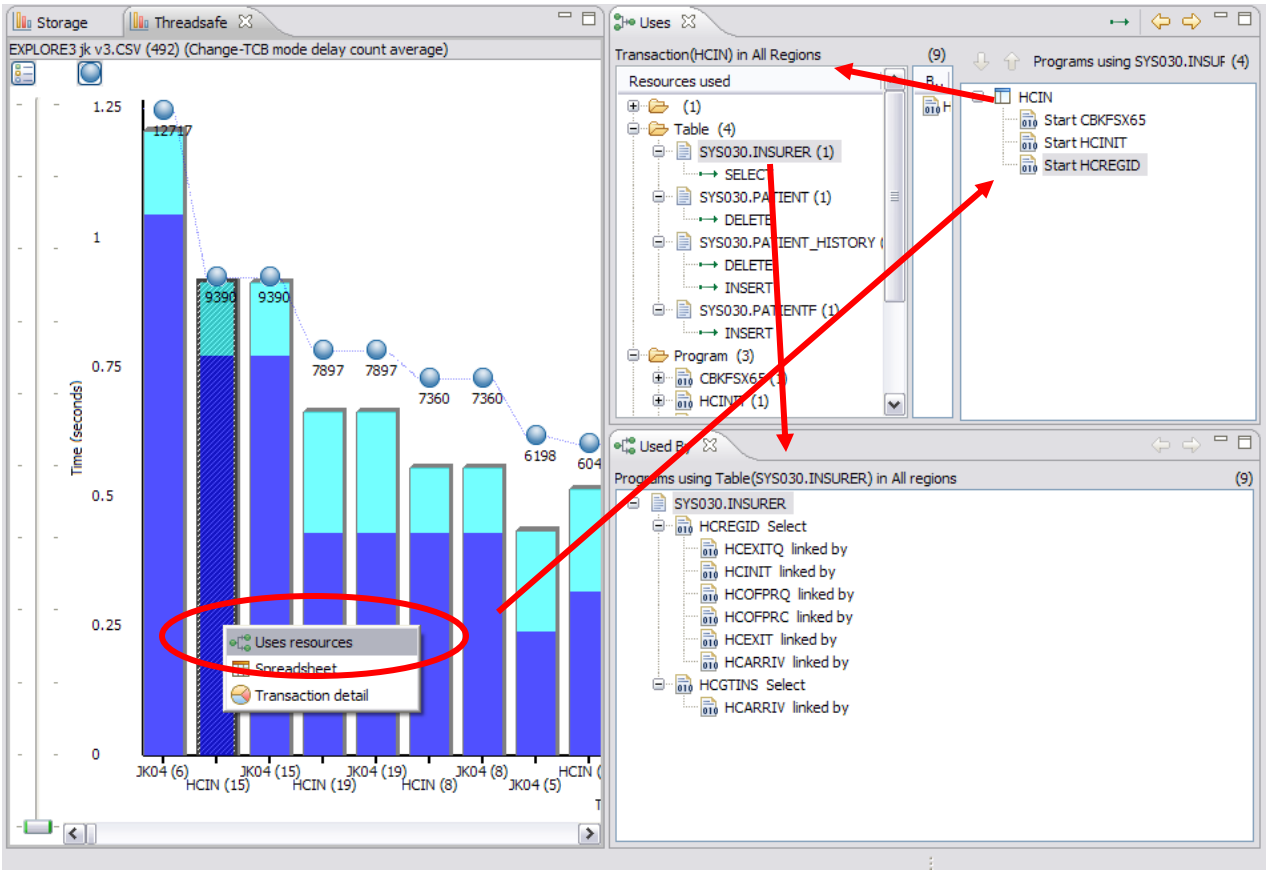
CICS Tools that can help – CICS PA

- **Which TCBs did my transaction use?**
  - How many TCB switches (change modes) occurred?
    - What was the Change Mode delay time?
  - How much Dispatch and CPU time did they use?
  - Performance Summary, List and List Extended Reports, …
  - Sample Report Forms …
    - CPU and TCB Usage, TCB Delays, Change Mode Delays, …

- **Why did my transaction take so long?**
  - Wait Analysis Report, Performance List Reports, …
- **Which Transaction(s) used GETMAIN SHARED?**
- **Where did my transaction go?**
  - Cross-System Report, …
  - Performance List, DB2 and WebSphere MQ Reports, …

Threadsafe (OTE) – Application analysis

- **Use a combination CICS Explorer, CICS IA and CICS PA o see the whole picture**



Recommended serialisation techniques

- **CICS API: ENQ and DEQ**
  - Advantages:
    - Commands are threadsafe and can be used in all CICS supported languages
    - Application failure will not result in a held lock
    - NOSUSPEND option to get control back if contention
    - No knowledge of assembly language required
  - Disadvantages:
    - Costs more cpu than a non-CICS api technique such as compare and swap
    - Always have to perform ENQ/DEQ even when no contention
    - Must consider implications of MAXLIFETIME option

- **CICS XPI for Global user exits: DFHNQEDX ENQUEUE & DEQUEUE**
  - Same as above.

Recommended serialisation techniques

- **Compare and Swap on shared data element**
  - Advantages:
    - Potentially the best performance
  - Disadvantages:
    - Cannot be used for fields greater than 4 bytes (8 bytes for CDS instruction)
    - For fields less than 4 bytes activity on adjacent bytes could cause additional attempts
      - Could move shared data element to new fullword
    - Requires assembly language program or subroutine

Recommended serialisation techniques

- **Compare and Swap or Test and Set on separate "lock" byte"**
  - Advantages:
    - "Lock" may be defined for non-contiguous areas
    - If using CS, "locked" status could be task number, terminal id
      - Identifies who owns the lock
  - Disadvantages:
    - Application failure while holding a lock will cause other TCBs to spin until the lock is manually cleared
      - Can avoid the spin by using a lock retry counter but access still denied until lock released
    - Requires assembly language program or subroutine

Example of COMPARE and SWAP on a shared element

```
* Increment a 4-byte field. ROLD and RNEW are two general purpose regs
INC      DS   0H
         L    ROLD,SHARED          Get  SHARED data element
RETRY    DS   0H
         LR   RNEW,ROLD            Get old value of SHARED
         LA   RNEW,1(,RNEW)        increment value
         CS   ROLD,RNEW,SHARED     Update SHARED element
         BNZ  RETRY                Update failed, start again
         B    RETURN               Update worked
```

Example of COMPARE and SWAP on a shared element with retry count

```
* Increment a 4-byte field. ROLD, RNEW and RCOUNT are three general purpose regs
* Abort if retries exceed maxtries
INC      DS   0H
         XR   RCOUNT,RCOUNT       Clear retry count
         L    ROLD,SHARED         Get SHARED data element
RETRY    DS   0H
         LA   RCOUNT,1(,RCOUNT)   Increment retry counter
         CL   RCOUNT,MAXTRIES     too many attempts?
         BNL  ERROR               yes quit trying
         LR   RNEW,ROLD           Get old value of SHARED
         LA   RNEW,1(,RNEW)       increment value
         CS   ROLD,RNEW,SHARED    Update SHARED element
         BNZ  RETRY               Update failed, start again
         B    RETURN              Update worked
ERROR    DS   0H
         < too many retry attempts >
```

Non-recommended serialisation techniques

- **LINK to a QUASIRENT program**
  - Adds TCB switches, degrades performance
- **CICS transaction class**
  - Maxactive(1) very crude, severe impact on response times
- **MVS enqueue/dequeue**
  - Issuing MVS services disallowed (until openapi in CICS TS 3.1)
  - Threadsafe CICSAPI programs cannot control when a switch to QR may occur

Application Design Considerations

- **An ideal threadsafe CICS-DB2 application:**
  - Uses only threadsafe CICS commands
  - All GLUEs on its execution path are threadsafe
  - Switches to L8 once, and stays there for its duration
    - Avoids TCB switching giving cpu decrease
    - Avoids running on QR giving throughput increase

- **Many applications won't fit this ideal**
  - It is still possible to design applications to minimise TCB switches
  - Do not interleave EXEC SQL requests with non threadsafe CICS commands
    - Place non threadsafe CICS commands before first SQL call or after last SQL

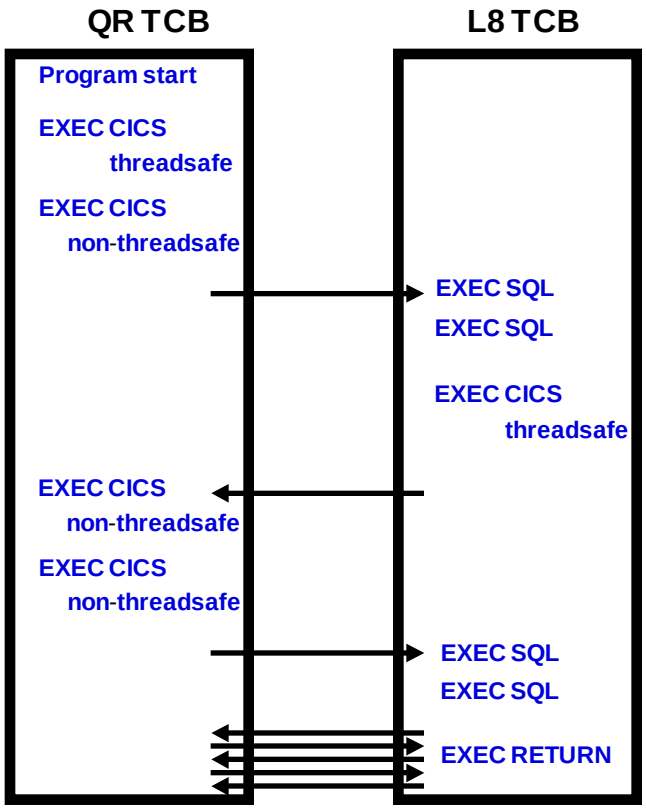
Application Design with CICS TS 3.1 : CICSAPI versus OPENAPI

- CICSAPI - services available today under QR TCB
  - CICS command level application programming interface
  - CICS system programming interface
  - CICS Resource Manager Interface (RMI)
  - CICS Exit Programming Interface (XPI) - for Global User exits
  - Systems Application Architecture (SAA) Common Programming Interfaces
    - CPI-C and CPI-RR
  - LE callable services
- OPENAPI - additional APIs possible under Open TCBs
  - Use of MVS services
  - Use of a specified set of POSIX services via MVS Unix System Services

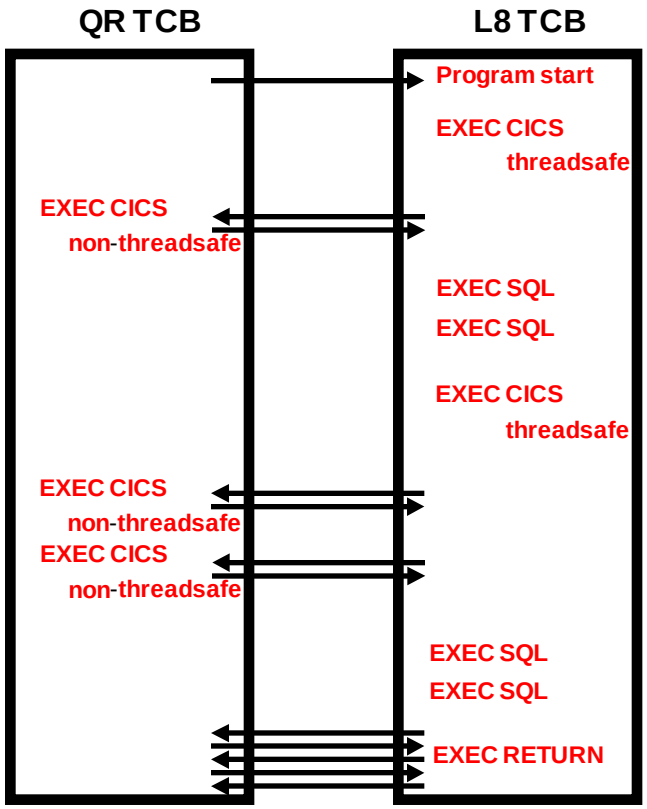
OTE Changes in CICS TS 3.1 : OPENAPI Programs

- New API Keyword on program definition
  - –Attribute of program. Applies to applications, TRUEs, URMs, PLT (ignored for GLUEs)
  - –CICSAPI (the default) means the program only uses CICS permitted interfaces
  - –OPENAPI means the program requires an Open TCB to use other APIs (OPENAPI requires CONCURRENCY(THREADSAFE))
- New L9 TCB for running userkey OPENAPI programs
  - –For OPENAPI programs TCB key must match PSW key for non-CICS apis to work (whereas CICS APIs run in either key irrespective of TCB key)
  - –SIT parm MAXOPENTCBS now covers L8 and L9 TCBs
- New CICS uses of L8 TCBs
  - –When accessing DOCTEMPLATEs or static HTTP responses held on HFS
  - –WebServices and XML support implemented using cicskey OPENAPI programs

CICS TS 3.1 - TCB switching



The program for transaction BLUE is defined THREADSAFE , API= CICSAPI



The program for transaction RED is defined THREADSAFE, API= OPENAPI ,EXECKEY=CICS

CICSAPI THREADSAFE versus OPENAPI THREADSAFE

- **NB: OPENAPI TRUEs must run CICS key on an L8 TCB**
  - A user key threadsafe OPENAPI program calling DB2 will switch from L9 to L8 then back to L9 for each DB2 request. Same applies to WMQ and sockets
  - A user key threadsafe CICSAPI program running on an L8 TCB will remain on the L8 TCB to call DB2 or WMQ or sockets.

- **Candidates for CICSAPI with THREADSAFE**
  - SQL programs with some non-threadsafe API
  - SQL programs with USER key
  - Mixed FC and DB2 and/or WMQ and/or socket applications
    - FC relies on the DB2 or WMQ or sockets call to push the task onto the L8 TCB

- **Candidates for OPENAPI with THREADSAFE**
  - programs with threadsafe APIs only including FC RLS and FC Local LSR
  - SQL programs with CICS key
  - CPU intensive programs

Further information

- CICS TS 2.3/3.1/3.2/4.1 Information Centers
  - Refreshed regularly, can be downloaded from:
  - http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi
- Redbook: Threadsafe considerations for CICS SG24-6351-00
  - http://www.ibm.com/redbooks
  - Third edition (November 2007) incorporates CICS TS 3.2 enhancements
- CICS Tools to help threadsafe migration
  - http://www.ibm.com/cics/tools