

Agenda

- 8:30 A.M. Continental Breakfast
- 9:00 A.M. Introductions – Threadsafe and CICS Tools
- 9:30 A.M. Performance Analyzer overview
- 9:45 A.M. Performance Analyzer hands-on lab/Break 10 min.
- 11:15 A.M. Interdependency Analyzer overview
- 11:30 A.M. Interdependency Analyzer hands-on lab/lunch 30 min.
- 1:30 P.M. Configuration Manager overview
- 1:45 P.M. Configuration Manager hands-on-lab/Break 10 min
- 3:15 P.M. Performance Analyzer Results lab
- 4:00 P.M. Seminar wrap up – Questions?



Preface

The following are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM, CICS, CICS/ESA, CICS TS, CICS Transaction Server, CICSplex, DB2, MQSeries, OS/390, S/390, WebSphere, z/OS, zSeries, Parallel Sysplex.

Java, JavaBeans, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names and logos may be trademarks or service marks of others.



Why make applications Threadsafe?

Improve performance

- ▶ CICS QR TCB is CPU constrained
- ▶ Application tasks are waiting excessively for the QR TCB
- ▶ The CICS region in general is CPU constrained

Reduce cost

Future positioning

- ▶ OTE function introduced
- ▶ TRUEs can exploit OTE
- ▶ Full application use of open TCBs



Why make applications Threadsafe? (Notes)

The CICS QR TCB is CPU constrained The CICS QR TCB is consistently reaching system CP SHARE(QR TCB is running at 100% CPU) and has to wait to be dispatched by the operating system. Every task in the QR TCB is being delayed. Defining transactions as threadsafe, processing as many tasks as possible on the L8 TCBs will remove this constraint on the QR TCB, and reduce the response times of both threadsafe and non-threadsafe transactions.

Application tasks are waiting excessively for the QR TCB The QR TCB is not CPU constrained, but application tasks are contending for their share of QR. Again, defining transactions as threadsafe and moving as many tasks as possible to L8 TCBs will reduce contention for QR, and reduce the response times of both threadsafe and non-threadsafe transactions.

The CICS region in general is CPU constrained The system as a whole is at or approaching 100% busy, and CICS is being constrained along with everything else.

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

- significant number of EXEC SQL calls invoked per task
 - all programs invoked between the first and last EXEC SQL call in each task are defined as threadsafe
 - all exits invoked as part of an EXEC SQL call are defined as threadsafe, and only contain threadsafe EXEC CICS commands
 - all exits invoked between the first and last EXEC 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 above characteristics as threadsafe will all but eliminate TCB switches for the associated CICS tasks.



CICS-DB2 Transactions in CICS TS 2.2 and higher

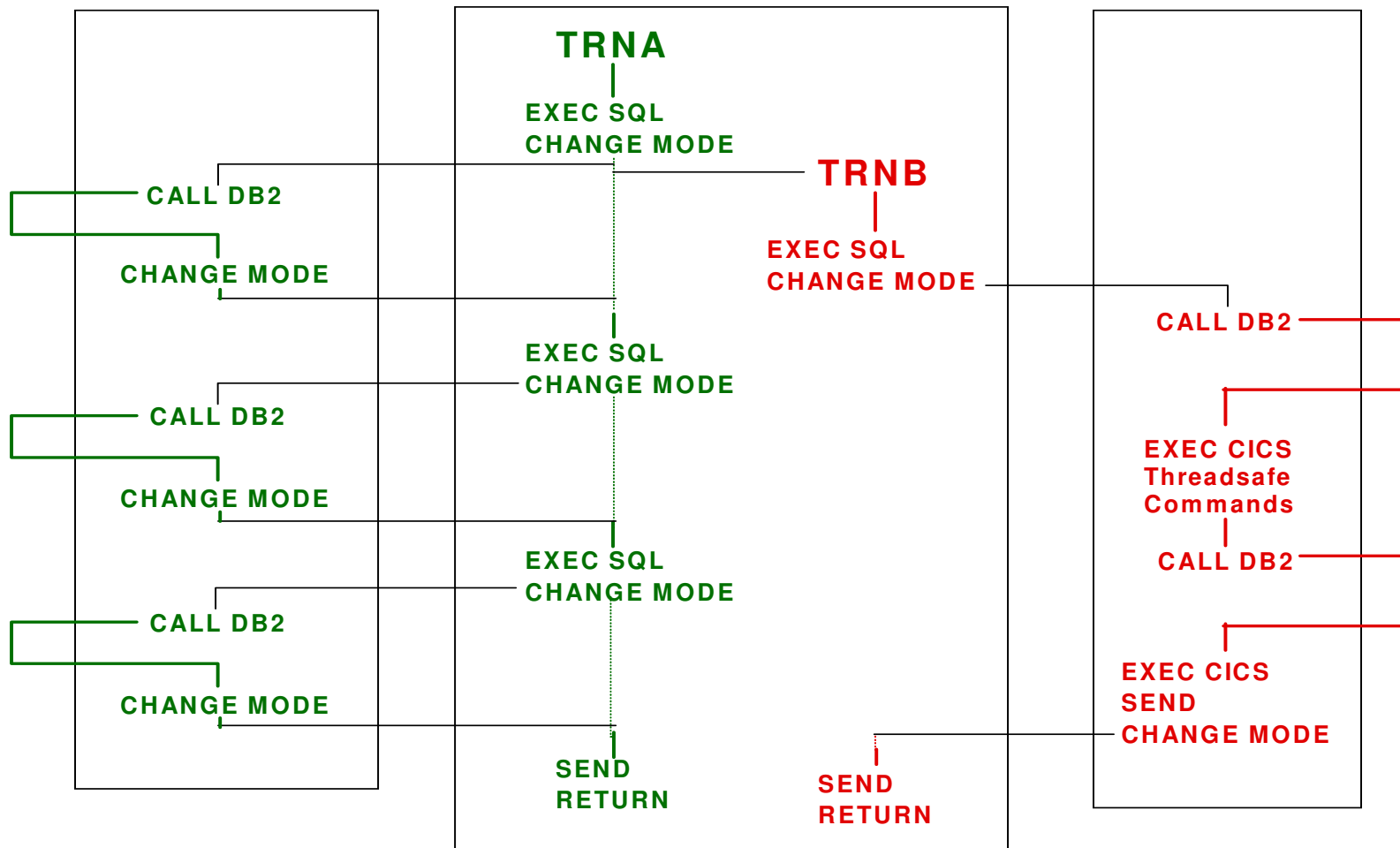
TRNA is non-threadsafe

L8001 TCB

QR TCB

TRNB is threadsafe

L8002 TCB



Notes

- The picture shows how the CICS-DB2 Attachment facility has been upgraded in CICS TS 2.2 to take advantage of OTE and use L8 open TCBs to call DB2 (when using DB2 V6 or higher) rather than privately managed DB2 Attach TCBs.
- TRNA executes a non threadsafe application program. Here the amount of TCB switching is the same as the previous release. Instead of switching to a DB2 Attach TCB, we use an L8 TCB. Switching is achieved via use of a dispatcher domain Change_mode request, instead of use of WAIT/POST logic as in previous releases.
- TRNB executes an application that is threadsafe, and has been defined to CICS with CONCURRENCY(THREADSAFE). When executing the first EXEC SQL request, the TRUE is invoked on an open TCB, and that TCB is used to call DB2. On return from the RMI because the application is defined as threadsafe we stay on the open TCB and return to the application on it. Any threadsafe CICS commands issued by the application will be executed on the L8 TCB, as will any further SQL requests. For the EXEC CICS SEND command (which is non threadsafe) we switch back to the QR TCB. Control will stay on the QR TCB until another DB2 request is issued. In this example there are no more DB2 requests. The EXEC CICS RETURN is executed on the QR TCB. Not shown in the diagram is the end of task syncpoint which will result in four TCB switches for the two phase commit protocol, ie switch to L8 to issue PREPARE to DB2, then back to QR, then switch to L8 to issue COMMIT to DB2, then back to QR.



CICS Tools portfolio

CICS Performance Analyzer

- Comprehensive batch performance reporting and analysis for tuning and capacity planning

CICS Interdependency Analyzer

- Understand your active application inventory for efficient maintenance and upgrades

CICS Configuration Manager

- Manage, replicate, and deploy CICS system definitions

CICS VSAM Transparency

- Enable VSAM to DB2 migration without rewriting applications

CICS VSAM Recovery

- Automate recovery of lost VSAM data

IBM Session Manager

- Provide secure, reliable, and easy access to multiple z/OS and OS/390 applications from a single terminal

CICS Online Transmission Time Optimizer

- Optimize 3270 data streams to increase your system performance

CICS Batch Application Control

- Simplify and automate batch access to CICS resources



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, ...



CICS Tools that can help – CICS IA

- Are my programs using shared resources?
 - ▶ CWA
 - ▶ Global user exit global work areas
- Storage acquired explicitly by the application program with the GETMAIN SHARED option
 - ▶ Data only Load module
- Which non threadsafe CICS commands is my program using?
- Which TCB did my CICS commands run on ?



CICS Tools that can help – CICS CM

- How can I simplify and automate the management of my CICS threadsafe resources ?
 - ▶ Minimizes manual work by operators and system programmers
- How can I migrate the CICS threadsafe resources from different environments under a structured change control process?
 - ▶ Enables migration of CICS resources from different environments under a structured change control process
- Can I see CICS Threadsafe resource history as well as provides back-out to previous change level?
 - ▶ Provides complete audit history of all CICS resource modifications
 - ▶ Tracks resource history as well as provides back-out to previous change level
 - ▶ Provides detailed reports of CICS resources



