IBM Software Group

Reduce CICS CPU usage with CICS Explorer and CICS Performance Analyzer for z/OS

Steve Blackman

IT Specialist

zSeries Technical Sales for CICS Tools

ON DEMAND BUSINESS

Updated: Friday, 05 December 2008

Abstract

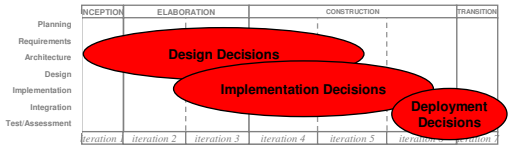
- **This session will show techniques and best practices for CICS performance analysis that can help you to optimize the performance of your CICS applications and systems, and achieve significant CPU savings.**
- **Using a combination of presentation and live demonstration, using the CICS Performance Analyzer plug-in for the new CICS Explorer, will show real-world examples such as threadsafe analysis, CPU time analysis, Response Time and Storage analysis.**
- **If CICS and overall z/OS performance is important to you, then you should attend this session.**

Agenda

- **Approach to performance analysis**
- **Best practices for some key performance focus areas**
- **Live demo**

Application Performance starts at the beginning

- **Performance problems . . .**
  - have moved to center stage
    - ⇨ *no longer simply an issue for internal users*
    - ⇨ *affects customers, revenue, market share, brand equity*
  - but are typically . . .
    - ⇨ ***discovered late***
    - ⇨ ***created early***



- ***Life-cycle Best Practices are essential to . . .***
  - Uncovering latent problems that appear only under load conditions
  - Making better informed configuration decisions
  - Ensuring applications meet end user requirements for response time and availability on day one
  - Ensuring scalability to meet unexpected traffic peaks & future growth
  - Identifying thresholds to be used for production monitoring
  - Avoiding all those business impacts

Analysis using CICS Performance Analyzer and the CICS Explorer

- **Generate SMF 110 records in CICS**
- **Choose representative time period(s) for analysis**
- **Run CICS PA reports to create aggregate csv files**
- **Download csv files to workstation**
- **Import into CICS Explorer**
- **Analyse data offline using CICS PA plug-in to CICS Explorer**
  - Sort sheet views by key metric columns
  - Produce appropriate bar chart views for selected rows
  - Drill down to Transaction Detail view
  - Dynamic link to CICS IA *Uses resources* (if available) to understand application structure

CICS Performance - Focus areas

- **Transaction timing primer**
- **Threadsafe analysis**
- **CPU time analysis**
- **Response time analysis**
- **Storage analysis**
- **File Access analysis**

Why bother with Performance Analysis? Is it really worth it?

- **"The result of the Danske Bank threadsafe conversion was a saving about 300 MIPS during the peak processing period. This can be directly attributed to reduced TCB switching."**
- **Detail results**
  - Based on CICS SMF 110 record analysis. The following CICS SMF 110 records were collected.
    - A point in time before converting to threadsafe for the Danske Bank applications. This was further subdivided into those applications that were issuing SQL calls and all other applications.
    - A point in time after conversion to threadsafe for the Danske Bank applications. This was further subdivided into those applications that were issuing SQL calls and to all other applications.
  - The basis for the measurement was the peak hour between 10-11 AM, during which Danske Bank expects a high transaction rate. The transaction rate per peak hour is shown in Table 11-2.
  - More detail in the CICS Threadsafe RedBook SG24-6351

Table 11-2 Transaction measurements

Date	Description	# Transactions	# TCB switches
10-02-2006 Before	Before converting, including those transactions issuing SQL calls.	3,179,639	250,820,638
11-27-2006 After	After converting, including those transactions issuing SQL calls.	2,678,469	104,224,554
10-02-2006 Before	Before converting, including all transactions.	7,859,741	280,879,869
11-27-2006 After	After converting, including all transactions.	6,462,891	128,826,384

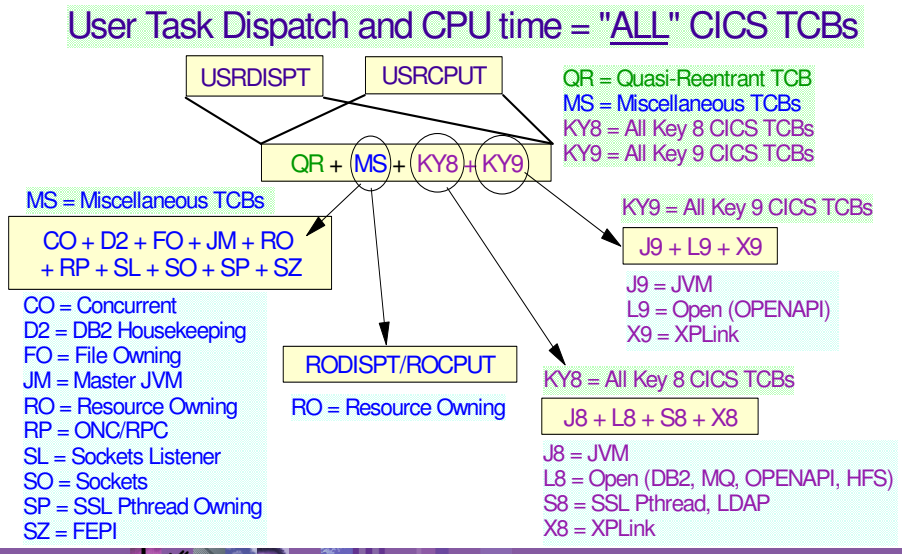
Transaction Timing - What's important?

- **Open Transaction Environment (OTE)**
  - Key metrics provided in SMF 110 records
- **Four major TCB classes**
  - Dispatch and CPU time by CICS TCB Mode (QR and MS)
  - Dispatch and CPU time by CICS TCB Mode (RO)
  - Dispatch and CPU time by CICS TCB Mode (Key 8 TCBs)
  - Dispatch and CPU time by CICS TCB Mode (Key 9 TCBs)
  - CPU time by CICS TCB Mode (J8, J9, L8, L9, S8, X8 and X9)
  - QR TCB Mode Dispatch Delay time (Dispatch Wait time)
  - CICS TCB Change Mode Delay time
- **Response time is typically most important from user PoV**
  - Major components of response time
    - Dispatch time, wait for re-dispatch time, suspend time, CPU time

Transaction Timing …

- **Transaction Start time and Response time**
- **Transaction Dispatch and CPU time**
  - Measured on "ALL" the CICS TCBs (QR, RO, CO, ..)
    - Including SL, SO, SP, D2, J8, J9, L8, L9, S8, X8 and X9 CICS TCBs
  - RLS File request CPU (SRB) time
- **Transaction Suspend time**
  - Includes "ALL" the transaction wait (suspend) time
- **Transaction Dispatch Wait time**
  - QR TCB Mode Dispatch Wait time (QR Mode Delay)
  - CICS TCB Change Mode Delay time

Transaction Timing - TCB taxonomy



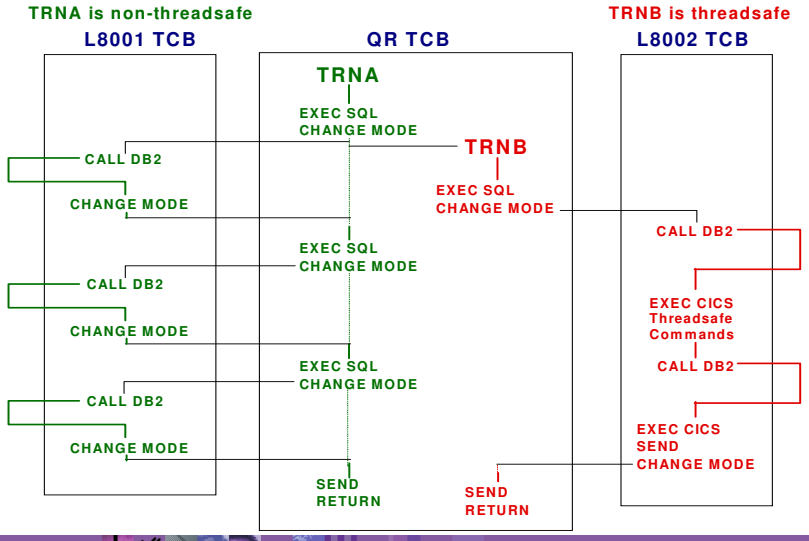
IBM Software Group

Threadsafe Analysis

ON DEMAND BUSINESS

CICS TCB switching

CICS-DB2 Transactions in CICS TS 2.2 and higher

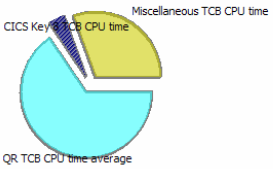


Threadsafe (OTE) - Best practices

- **Maximize the use of Open Transaction Environment (OTE)**
  - Make application programs threadsafe where possible, and
    - QR CPU time will decrease and QR TCB contention will be reduced!
    - CPU will decrease as a result of fewer switches
    - Both result in reduced response time!
  - Useful metrics
    - Targets for Optimization : QR > 30%
    - Mode switches > 25
    - Highly dependent on transaction usage

CPU time: JK01

3 transaction(s): 73 TCB mode switches (average).
0.1351 seconds CPU time per transaction



Reference	Value	Overall	Relative
Miscellaneous TCB CPU time average	0.041300	31%	-
RO TCB CPU time average	0.041300	31%	100%
CICS Key 8 TCB CPU time average	0.005400	4%	-
L8 TCB CPU time average	0.005400	4%	100%
QR TCB CPU time average	0.088400	65%	-
QR TCB CPU time average	0.088400	65%	100%

Threadsafe (OTE) - Best practices

- **Minimize CICS TCB Change Modes**
  - Useful metrics
    - If unavoidable, keep Change Mode Delay time to a minimum
    - Keep the number of TCB Change Modes to a minimum
    - Goal is QR TCB mode delay time = 0, or as low as possible

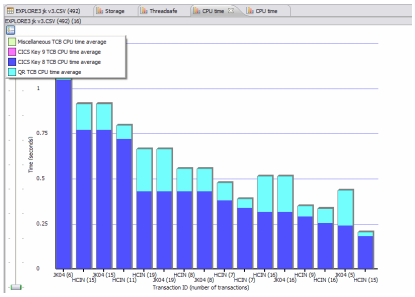
EXPLORE3.jk v3.CSV (492) | Storage | Threadsafe | EXPLORE3.jk v3.CSV (492) (29)

/PA Data/EXPLORE3.jk v3.CSV/22-06-2007+/..

..	Transactio...	Change-TCB ...	Change-TCB mo...	QR TCB dispatch time ...	QR TCB dispatc...	QR TCB CPU time aver...	QR TCB mode delay time ...	QR TCB mode delay count...
35	JK04	0.387400	12717	0.205700	6427	0.159700	0.234200	6426
42	HCIN	0.351700	6043	0.268900	3042	0.198400	0.204400	3041
52	JK04	0.351700	6043	0.268900	3042	0.198400	0.204400	3041
18	HCIN	0.341800	7897	0.289100	3970	0.234200	0.186400	3969
18	JK04	0.341800	7897	0.289100	3970	0.234200	0.186400	3969
28	HCIN	0.334900	9390	0.182900	4734	0.143500	0.163500	4733
28	JK04	0.334900	9390	0.182900	4734	0.143500	0.163500	4733
37	HCIN	0.240200	7360	0.161700	3707	0.124900	0.116200	3706
37	JK04	0.240200	7360	0.161700	3707	0.124900	0.116200	3706
23	JK04	0.184700	6198	0.231000	3125	0.194100	0.085600	3124
32	HCIN	0.147200	3520	0.099700	1773	0.078600	0.078900	1772
40	HCIN	0.167700	5721	0.098100	2880	0.077500	0.076900	2879
17	HCIN	0.141000	3892	0.064500	1962	0.048900	0.072400	1961
25	HCIN	0.122800	4011	0.075400	2021	0.058900	0.068700	2020
35	HCIN	0.104300	4715	0.117400	2378	0.098000	0.046700	2377
06	HCIN	0.053700	1590	0.030800	802	0.022700	0.024100	801
14	DC05	0.036400	1272	0.033100	679	0.028100	0.016200	678
00	HCEQ	0.030400	1600	0.068100	801	0.053900	0.013000	800
18	CSMI	0.025200	1218	0.066300	610	0.057900	0.011000	609
20	HCEQ	0.022300	1720	0.070100	861	0.062400	0.010600	860
00	HCEQ	0.024000	1300	0.033300	651	0.027200	0.009100	650
30	CSMI	0.021000	1030	0.056700	516	0.048800	0.008300	515
11	HCEQ	0.020300	1661	0.062600	831	0.056500	0.007600	830
17	HCEQ	0.018300	1617	0.058900	809	0.051300	0.006900	808
79	CSMI	0.105300	1078	0.111200	541	0.083700	0.006600	540
00	HCEQ	0.014000	1360	0.015200	681	0.011400	0.005200	680
00	HCEQ	0.013500	1480	0.040900	741	0.034500	0.004500	740
14	HCEQ	0.012800	1484	0.032200	743	0.026000	0.003100	742
20	HCEQ	0.009900	1720	0.070300	861	0.063200	0.000800	860

Threadsafe (OTE) - Sub-system Best practices

- **CPU time is now measured in the CICS CPU time fields!**
  - Avoid double accounting
    - User CPU time: Total CICS TCB CPU time
  - Useful metrics
    - L8 CPU time now contains the DB2, MQ, and IP CICS Socket CPU time
- **If the application is threadsafe ...**
  - L8 CPU time will also contain application CPU time and QR CPU time will decrease with reduced QR TCB contention!



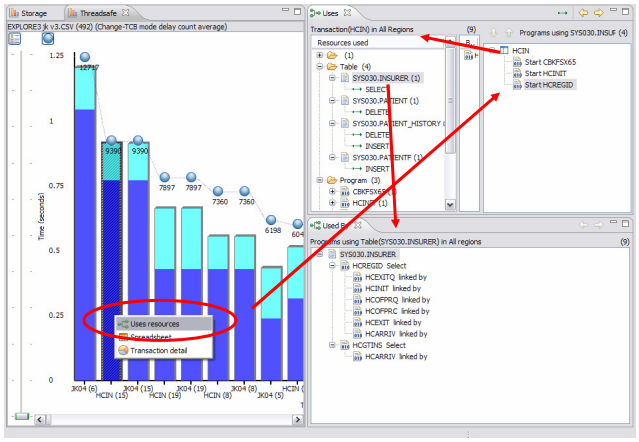
MQ Requires WebSphere MQ V5.3.1 or later, IP CICS Sockets Requires z/OS V1.7 or later

Open TCBs can optionally be used - see z/OS Communications Server IP CICS Sockets Guide

DB2 Requires DB2 7.1 or later

Threadsafe (OTE) - Application analysis

- **SMF records only tell you TCB switches by *transid***
- **You must enable threadsafe by *program*!**
- **CICS Interdependency Analyzer can show you your transaction structure**



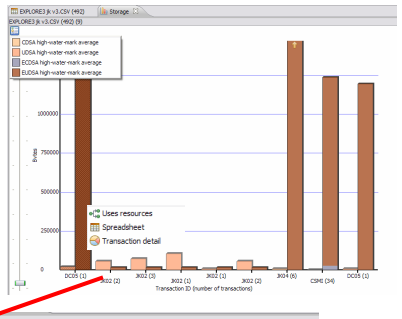
IBM Software Group

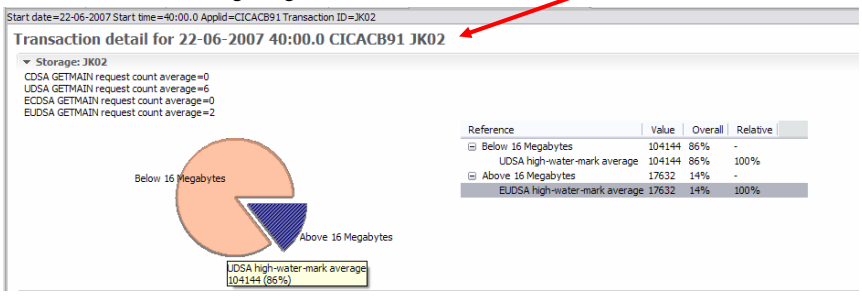
Storage Analysis

ON DEMAND BUSINESS

Storage Usage - Best practices

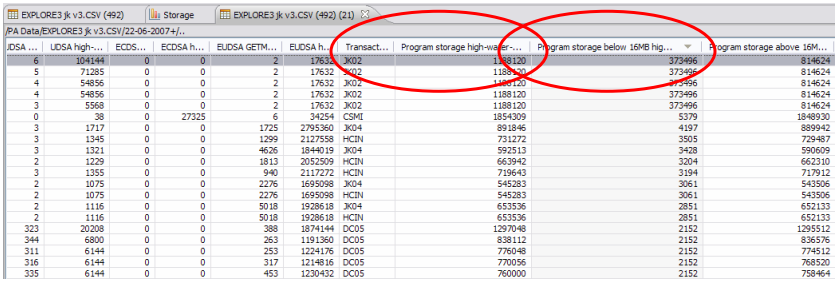
- **User Storage User Dynamic Storage Area (DSA)**
  - Minimize or eliminate Transaction storage usage below 16MB
  - Useful metrics (Below and Above 16 Megabyte line)
    - UDSA Getmain count should approach 0
    - UDSA Storage high-water-mark < 10K





Storage Usage - Best practices

- **Program Storage**
  - Applications should not use large amounts of Program Storage
    - If required, ensure it is above the line
  - Useful metrics
    - Storage high-water-mark Total
    - Storage high-water-mark by Dynamic Storage Area (DSA)
    - Storage high-water-mark Below and Above 16MB



EXPLORE3 jk v3.CSV (492) | Storage | EXPLORE3 jk v3.CSV (492) (21)

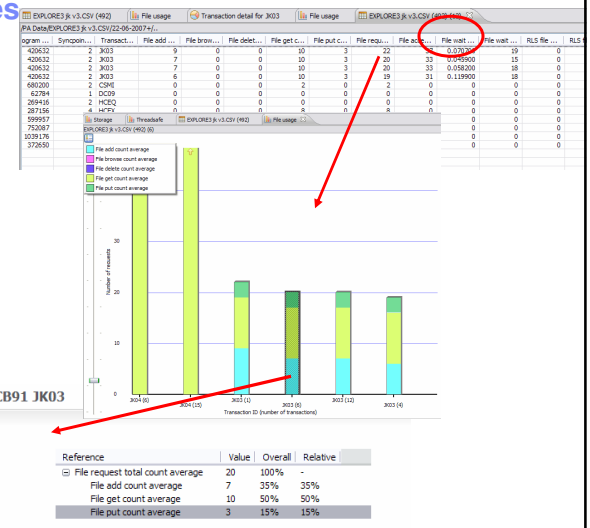
/PA Data/EXPLORE3 jk v3.CSV/22-06-2007+/..

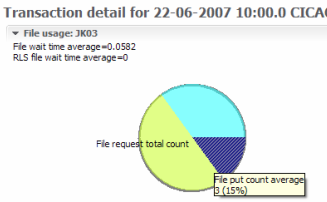
/DSA ...	UDSA high-...	ECDS...	ECDSA h...	EUDSA GETM...	EUDSA h...	Transact...	Program storage high-water-...	Program storage below 16MB hig...	Program storage above 16M...
6	104144	0	0	2	17632	JK02	1188120	373496	814624
5	71285	0	0	2	17632	JK02	1188120	373496	814624
4	54856	0	0	2	17632	JK02	1188120	373496	814624
4	54856	0	0	2	17632	JK02	1188120	373496	814624
3	5568	0	0	2	17632	JK02	1188120	373496	814624
0	38	0	27325	6	34254	CSMI	1854309	5379	1848930
3	1717	0	0	1725	2795360	JK04	891846	4197	889942
3	1345	0	0	1299	2127558	HCIN	731272	3505	729487
3	1321	0	0	4626	1844019	JK04	592513	3428	590609
2	1229	0	0	1813	2052509	HCIN	663942	3204	662310
3	1355	0	0	940	2117272	HCIN	719643	3194	717912
2	1075	0	0	2276	1695098	JK04	545283	3061	543506
2	1075	0	0	2276	1695098	HCIN	545283	3061	543506
2	1116	0	0	5018	1928618	JK04	653536	2851	652133
2	1116	0	0	5018	1928618	HCIN	653536	2851	652133
323	20208	0	0	388	1874144	DC05	1297048	2152	1295512
344	6800	0	0	263	1191360	DC05	838112	2152	836576
311	6144	0	0	253	1224176	DC05	776048	2152	774512
316	6144	0	0	317	1214816	DC05	770056	2152	768520
335	6144	0	0	453	1230432	DC05	760000	2152	758464

File Analysis

File Usage - Best practices

- **Application should run without any resource constraints - avoid deadlocks, waits, or abends**
  - Useful metrics
    - File wait count, File wait time, RLS File wait count, RLS File wait count: as close to zero as possible





Reference	Value	Overall	Relative
File request total count average	20	100%	-
File add count average	7	35%	35%
File get count average	10	50%	50%
File put count average	3	15%	15%

File Usage - Best practices

- **Correct specification of number of VSAM strings and buffers is crucial for good performance of an LSR pool.**
  - Useful metrics
    - String Wait count > 0
    - Total String Wait count > 0

Regions | Tasks | Connections | Terminals | Files | Transactions

CNX0211I Scope: CICSPLX1. Resource: LOCFILE. 117 records collected at 01-Dec-2008 18:02:33

Region	Name	String Count	String Wait Count	Total String Wait Count	LSR Pool ID	Status	Open Status
CICSDM25	ACCTFIL	1	0	0	01	ENABLED	CLOSED
CICSDM25	ACCTNAM	1	0	0		ENABLED	CLOSED
CICSDM25	ACINUSE	1	0	0	01	ENABLED	CLOSED
CICSDM25	ADNREPF0	2	0	0	01	ENABLED	OPEN
CICSDM25	CBKCNTL	1	0	0	00	ENABLED	CLOSED
CICSDM25	CCVCTL	8	0	0	00	ENABLED	CLOSED
CICSDM25	CCVDDD	8	0	0	08	ENABLED	OPEN
CICSDM25	CCVJNL	8	0	0	00	ENABLED	CLOSED
CICSDM25	CCVMIGC	10	0	0	08	ENABLED	OPEN
CICSDM25	CCVPT1	8	0	0	00	ENABLED	CLOSED
CICSDM25	CCVPT2	8	0	0	00	ENABLED	CLOSED
CICSDM25	CCVR0001	5	0	0	00	UNENABLED	CLOSED

Total requests queued because all strings in use

IBM CICS Explorer - The New Face of CICS

- **Key features**
  - Common, intuitive, Eclipse-based environment for architects, developers, administrators, system programmers, and operators
  - Task-oriented views provide integrated access to broad range of data and control capabilities
  - Powerful, context-sensitive resource editors
    - Integrated in Rational Developer for System z V7.5
  - Integration point for CICS TS, CICS Tools, CICS TG, PD Tools, and Rational Tools
  - Extensible by ISVs, SIs, and customers
- **CICS support**
  - CICS Transaction Server for z/OS V3.1, V3.2
    - CICSPlex SM WUI server required for CICS resource views
- **Availability**
  - SupportPac available November 5 2008
  - Even more capability in 2009
  - More information at http//ibm.com/cics/explorer
- **Statement of Direction dated 5 Aug 2008**
  - http://www.ibm.com/common/ssi/rep_ca/8/897/ENUS208-248

CICS EXPLORER

Available now in CICS Explorer

- **CICS TS - Real-time resource status**
- **CICS IA - Resource dependency views**
- **CICS CM - Query and manage resource definitions**
- **CICS PA - Performance data drill-down, Threadsafe analysis**
- **RDz - CICS resource definition**
- **Coming soon - Plug-ins for CICS TG and OMEGAMON XE for CICS**

CICS Explorer SupportPacs

- **CS1J: IBM CICS Explorer for Windows SupportPac**
  - CICS Transaction Server for z/OS, V3.1, or later
  - Category 2 - un-supported
- **CS1O: IBM CICS Explorer for Linux SupportPac**
  - CICS Transaction Server for z/OS, V3.1, or later
  - Category 2 - un-supported
- **CS1N: CICS Tools Plugins for SupportPac**
  - CICS Configuration Manager (CM) for z/OS V1.2, and/or
  - CICS Interdependency Analyzer (IA) for z/OS V2.2, and/or
  - CICS Performance Analyzer (PA) for z/OS V2.1
  - Category 2 - un-supported
- **CA1R: CICS Explorer SDK SupportPac**
  - Classes and javadoc to integrate in-house, ISV, and SI tools with the CICS Explorer

All CICS Explorer SupportPacs can be downloaded from the CS1J page (http://tinyurl.com/6o6n9v) or from the CICS Explorer home-page http://ibm.com/cics/explorer

IBM Software Group

Demonstration

ON DEMAND BUSINESS

IBM Software Group

Backup material

ON DEMAND BUSINESS

Terminology

- **Quasi-reentrant (QR) TCB**
  - The main CICS TCB under which all application code runs prior to OTE
  - CICS dispatcher sub-dispatches 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-re-entrancy allows programs to share virtual storage e.g. CWA without the need to protect against concurrent update
  - CICS code takes advantage of quasi-re-entrancy, 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 sub-dispatching 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