CICS Performance and Tuning 101

Ezriel Gross - Circle Software Incorporated

Tuesday, March 8, 2016

Agenda

- Introduction to tuning and reasons to tune
  - Application versus systems
- Tuning methodology
- Anatomy of response time
- Data collection and reporting facilities
- Getting started: monitoring, DFH0STAT, end-of-day (EOD) statistics
- Examples of resources to tune
- Summary

Why tune?

- **Poor response times**
  - Application versus system
  - Network
  - DASD
- **Increased workload**
  - Consolidations
  - Increased volumes
- **Hardware considerations**
  - Postpone upgrades
- **Application costs**
- **Learning experience**

When is tuning usually performed?

- **When problems or issues occur**
- **Why not tune regularly?**
  - Lack of resources due to budget cuts, staff reductions
  - Lack of knowledge
  - Lack of interest – application or file tuning
  - Dependence on outside parties
  - “If it ain’t broke don’t fix it” attitude
  - Third party packages and applications
  - ROI cannot be determined in advance

Rules before starting

- Tuning is a “top-down” activity
- Make changes to address major constraints
- One major change at a time
- Some changes require iteration to find the right values
- Change should be done gradually and monitored
- Tuning will not always be effective
- Do not tune for the sake of tuning
- Have a fallback position

Methodology

Observe

Measure

Analyze

React

Verify

Implement

Methodology

General performance tuning guidelines:

- **Observe**
  - Understand your startup procedure and workload
  - Set realistic objectives
  - Develop a baseline to which you can compare
    - CPU Utilization: overall and CICS
    - Number of tasks per day or hour: peak and average
    - Response times

- **Measure**
  - Identify areas to tune
  - Determine measurement timeline
  - Select tools to be used
    - IBM supplied: DFH0STAT, EOD Statistics, CICS tables/RDO information, LISTCAT etc.
    - Third party monitors and tools

Methodology

- **Analyze**
  - Review outputs
  - Identify tuning opportunities

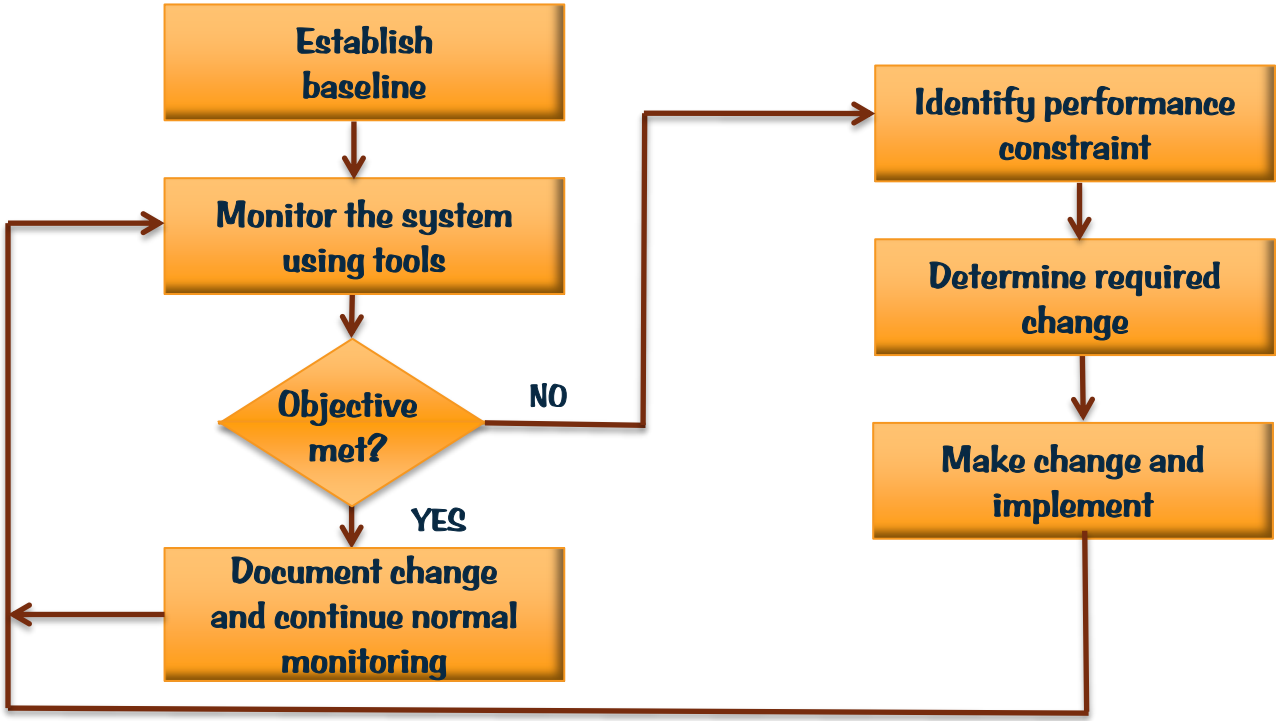
- **React**
  - Make appropriate changes
    - Use test or quality environments first
    - Word to the wise
      - Make major changes one at a time
      - Follow installation standards: change management
    - Ensure backup or fallback plan is ready

Methodology

- **Verify**
  - Review results from changes
  - Make appropriate changes, as required
    - Some tuning may require several iterations: for example LSR pool tuning
  - Go back to the Measure step until changes are meeting your objectives

- **Implement**
  - Move to production and go back to the Observe step

Methodology



Knowing CICS or your applications is not enoughÖ

No two CICS systems are the same !

You must know both CICS and the applications before you can effectively determine performance.

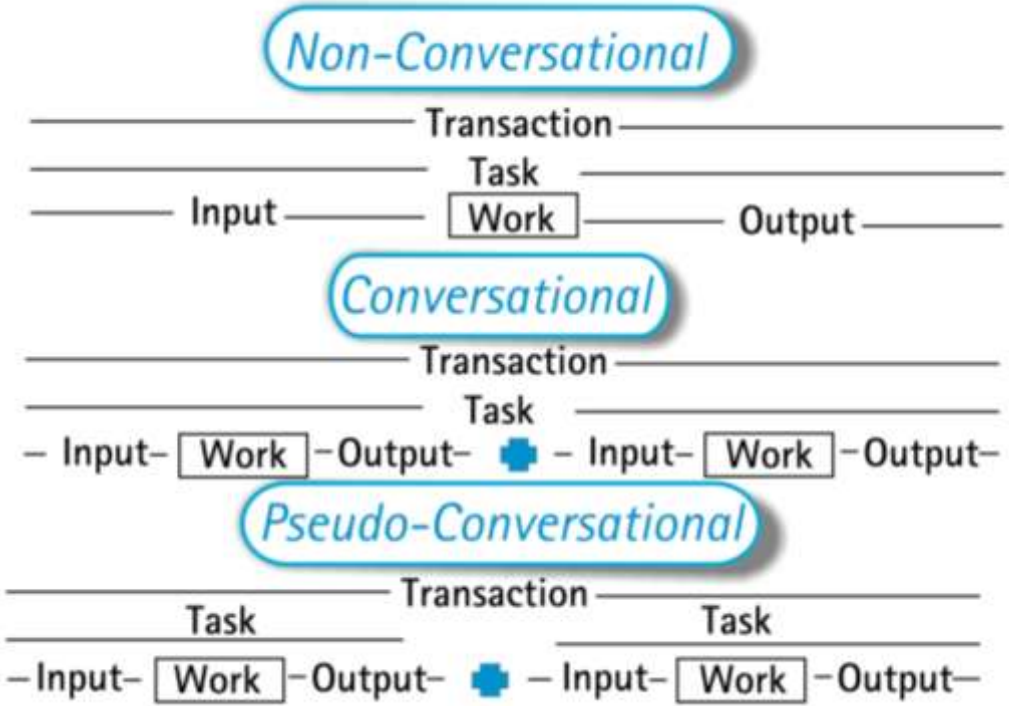
Performance opportunities

- Response time problems
- Processor overloaded
- CPU problems and costs
- Provision for increased workloads
- Availability and reliability problems
- Lack of certain types of CICS resources
- Capacity planning base line
- Realization of new technologies

Response times

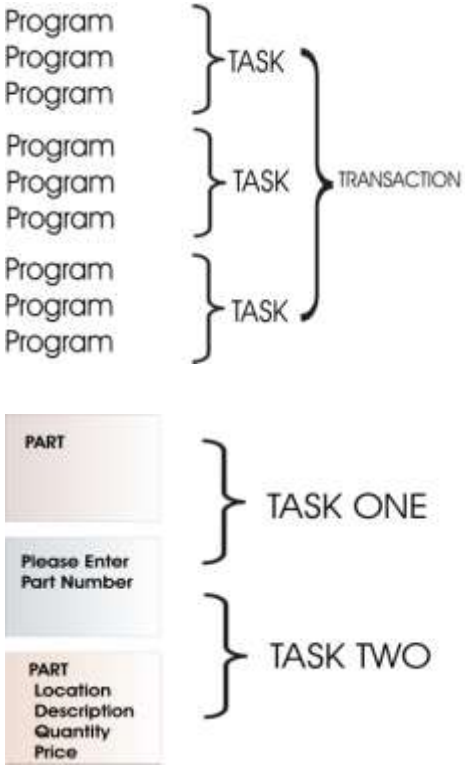
- **System response**
  - Allocation of resources
  - Processor speed
  - Design of application code
- **Network response**
  - Transmission through the network
- **DASD response**
  - Caching and buffering to reduce or eliminate
- **Increased workload**
  - Can cause failures in otherwise stable environment

Response times



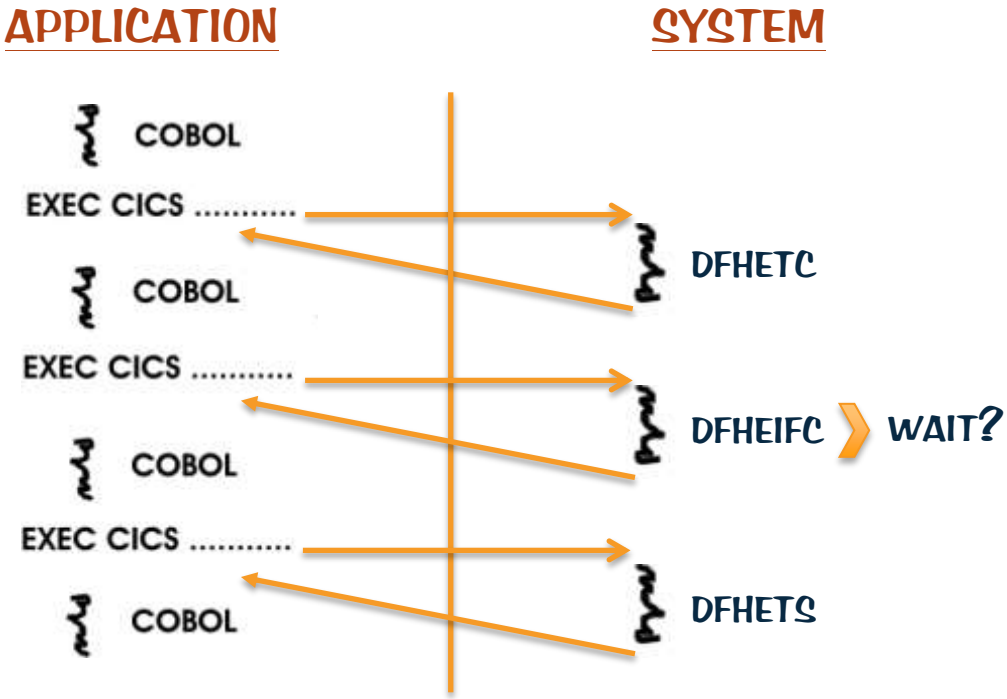
CICS tasks and programs

- A task is an instance of a transaction started by a user.
- When a user types in data and presses Enter or a Function key, CICS begins a Task and loads the necessary programs.
- Tasks run concurrently. Therefore, a user can run multiple instances of the same transaction simultaneously.
- CICS multitasks giving fast response times.
- CICS runs each task, briefly giving CPU to each one.

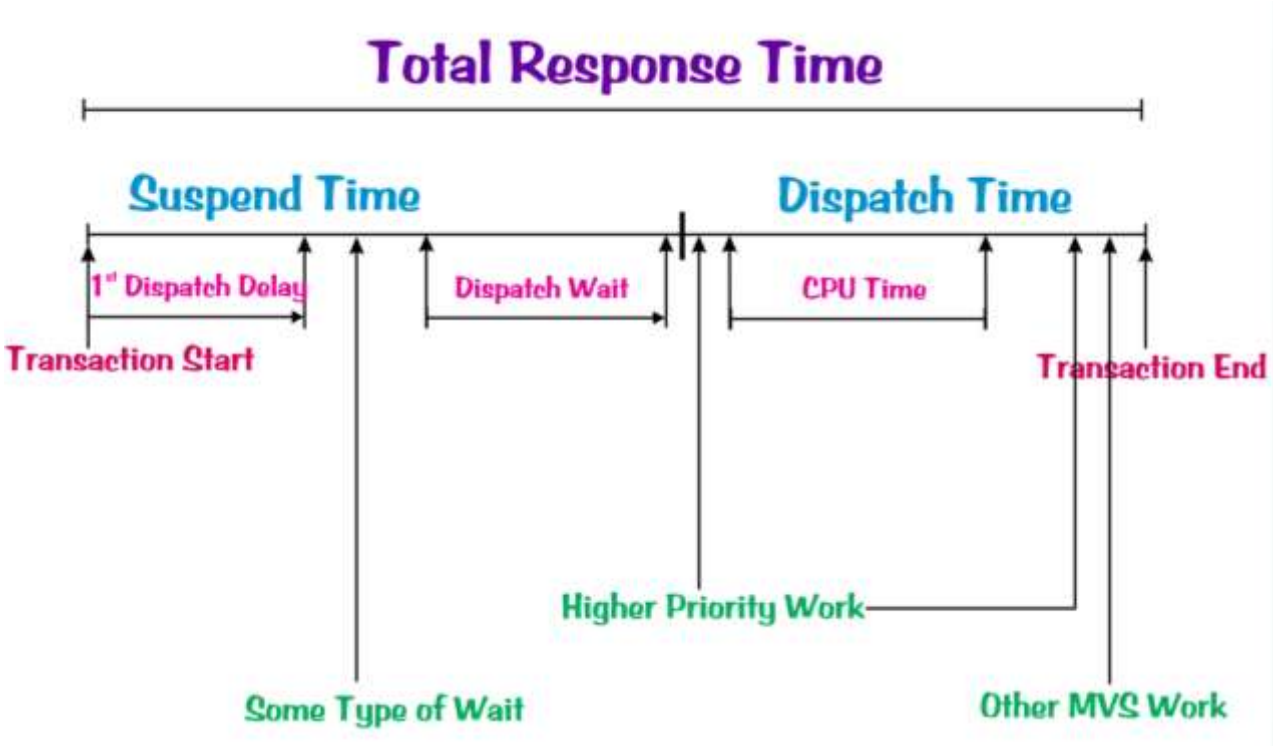


EXEC interface

- CICS programs look like batch with the insertion of Execute CICS commands.
- The CICS commands are used to request Services.
- CICS commands must be translated into COBOL prior or during program compilation.



Response times



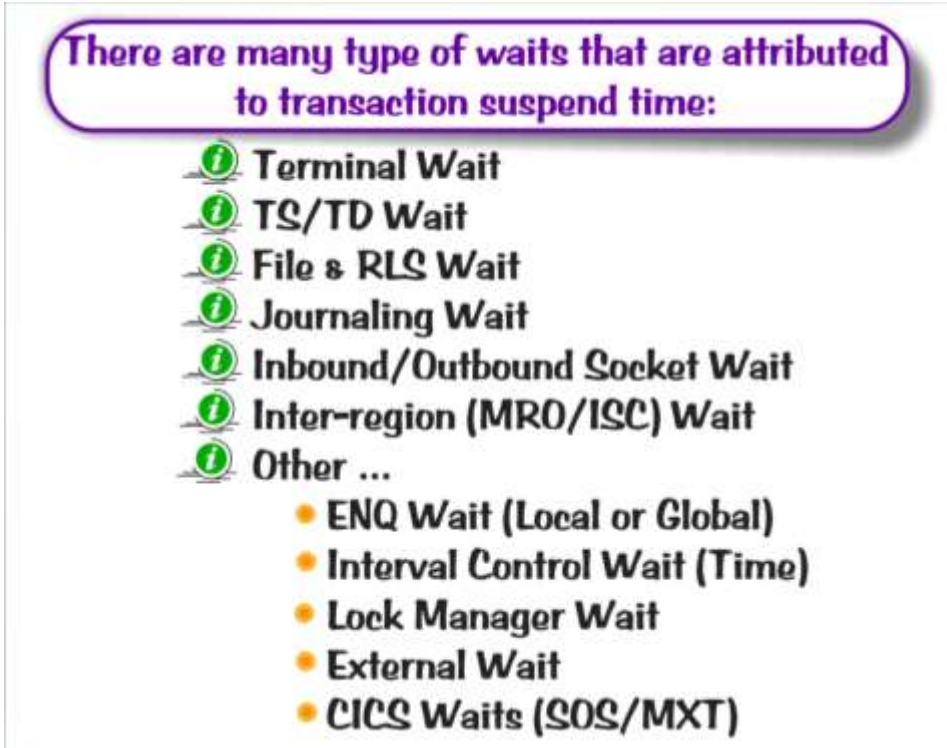
Anatomy of response time

- Response time consists of two elements:

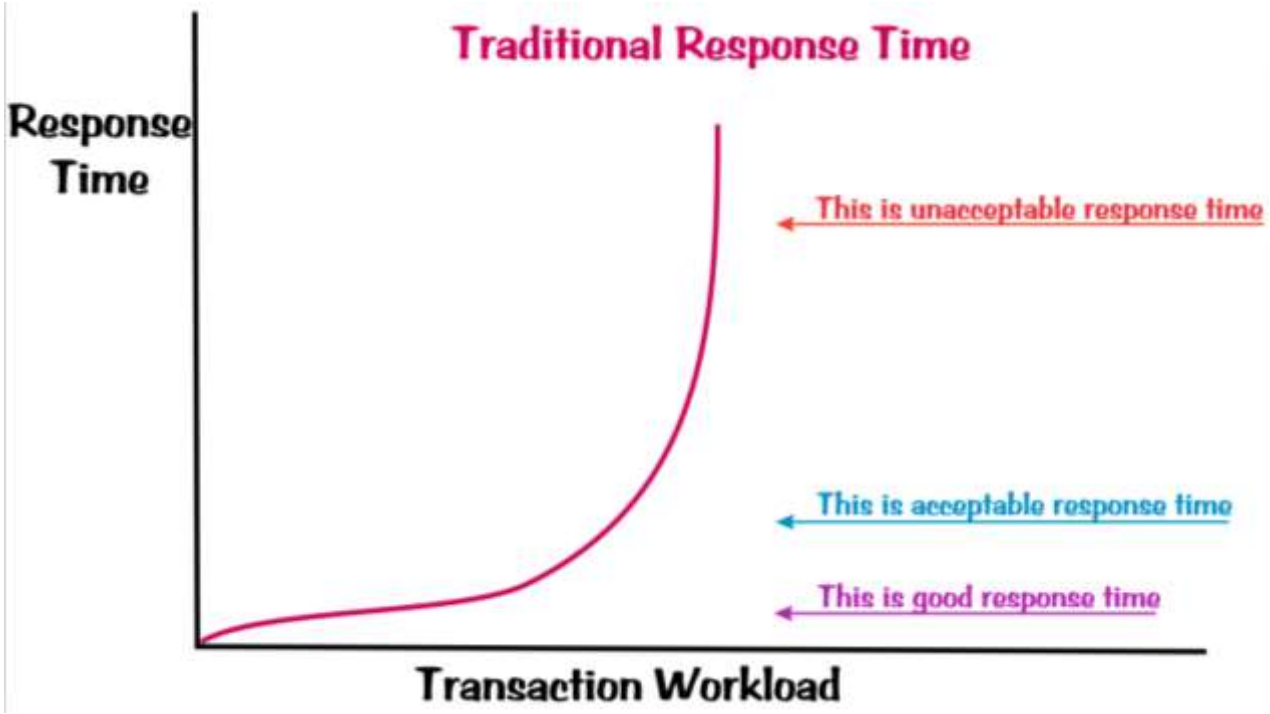
    1. **Suspend time:** the time a task is not executing (waiting).
    2. **Dispatch time:** the time that CICS thinks the task is executing. This time is further divided into:
        A. **CPU time:** the time the task is executing on CPU.
        B. **Wait time:** the time the CPU has been taken away from the task without the knowledge of CICS.

- CPU to dispatch ratio:
  - Ratio = (CPU time/dispatch time) * 100
  - Objective is 80% or higher

Types of waits



Resource load



Instrumentation data and measurement tools

CICS Statistics
C.M.F.
Aux. Trace
IBM CICS Performance Analyzer
Omegamon/CICS
The Monitor for CICS
Service Level Reporter
GTF Trace
RMF Reports
SAS/Meryll's code for CICS
C\TREK
/LISTCAT
DFHOSTAT (STAT)
IPCS (Real Bad Headache!)

CICS monitoring facility (CMF)

- CMF collects data about all transactions in CICS
- Records are written to SMF for later offline processing
- CMF collects four classes of data: exception, identity, performance and transaction resource
- CMF can produce a significant volume of data
- CICS compresses the data by default
- To exclude monitoring data fields, use a monitoring control table (MCT)
- To process output use:
  - Tivoli Decision Support for z/OS
  - CICS Supplied sample program DFH$MOLS
  - CICS Performance Analyzer (CICS PA)

CMF data types

- **Exception Class**
  - Information about resource shortages encountered
    - Queuing for file strings
    - Wait for Temporary Storage buffers
  - Highlights problems in CICS system operation
  - Identifies system constraints that affect performance
  - One exception record written for each condition that occurs
- **Identity Class**
    - Provides enhanced audit information
    - Captures identity propagation data from a client system across a network for eligible transactions

CMF data types

- **Performance Class**
  - Provides detailed transaction information
  - Processor and elapsed time
  - Time spent waiting for I/O, etc…
  - One record per transaction
- **Transaction Resource Class**
  - Additional transaction level information about individual resources accessed by a transaction
  - Items such DPLs, file and temporary storage queues
  - One transaction resource record per transaction monitored
  - Record cut only if transaction accesses at least one resource being monitored

DFH$MOLS

- Sample program supplied with CICS that can process and print SMF records produced by CMF
- Can also produce output data sets from compressed records for use by other analysis products
- Sorts and prints monitoring data based on parameters passed
- Can select or exclude data for printing based on applid, userid, tranid, task number, etc.
- SMF data set must be unloaded prior to using
- MCT is key to determining how much data is collected

CICS Monitoring Control Facility (CEMN)

```
CEMN                    CICS Monitoring Control Facility              CT52 CICSTS52

Type in your choices. When finished, press ENTER.

Item                                  Choice      Possible choices

Monitoring Status              ===>   ON          ON, OFf
Exception Class                ===>   ON          ON, OFf
Performance Class              ===>   ON          ON, OFf
Resource Class                 ===>   ON          ON, OFf
Identity Class                 ===>   OFF         ON, OFf

DPL Resource Limit             ===>   8           0, 1-64 DPLs
File Resource Limit            ===>   8           0, 1-64 Files
Tsqueue Resource Limit         ===>   16          0, 1-64 Tsqueues

Compression Status             ===>   YES         No, Yes

Converse Status                ===>   YES         No, Yes
Syncpoint Status               ===>   YES         No, Yes



PF1=Help     3=End     5=Options                              9=Error List
```

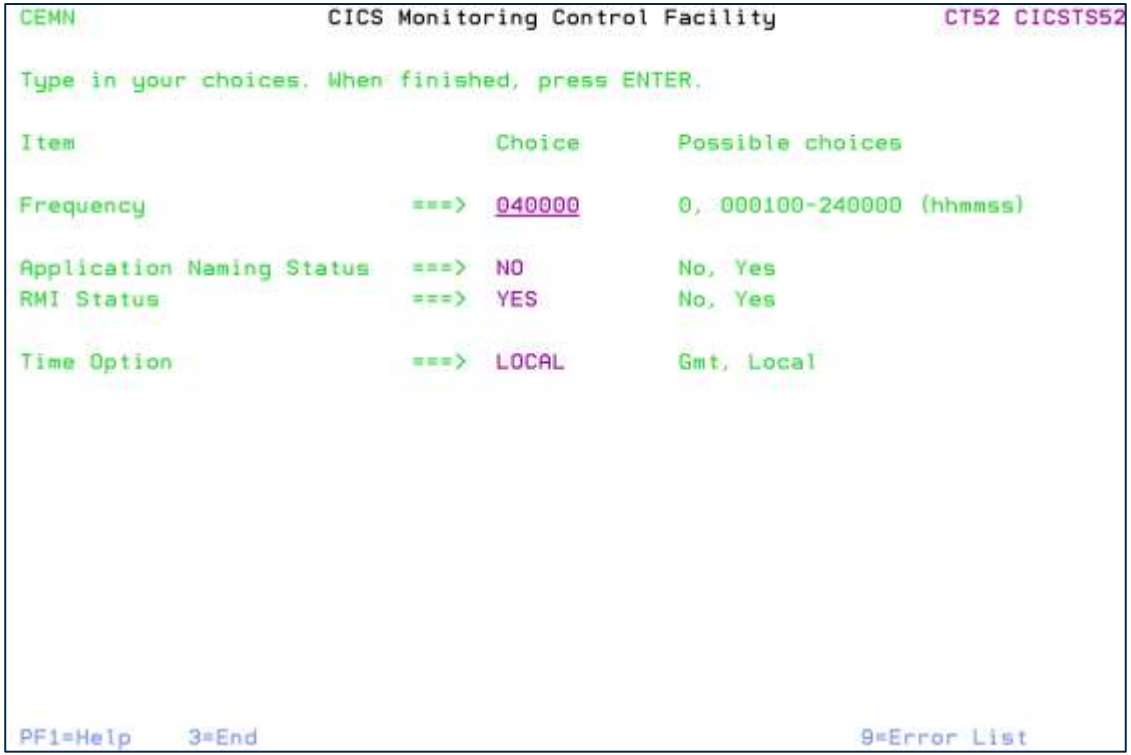
CICS Monitoring Control Facility (CEMN)

```
CEMN                Help: CICS Monitoring Control Facility


 (2) DESCRIPTION OF FIELDS 1.

 The settings shown on the main panel are:-

  Monitoring Status          -  Whether monitoring is required.
  Exception Class            -  Whether exception class data is required.
  Performance Class          -  Whether performance class data is required.
  Resource Class             -  Whether resource class data is required.
  Identity Class             -  Whether identity class data is required.

  DPL Resource Limit         -  The maximum number of DPLs (Distributed
                                Program Links) for which resource class
                                data is collected.
  File Resource Limit        -  The maximum number of files for which
                                resource class data is collected.
  Tsqueue Resource Limit     -  The maximum number of TS queues for which
                                resource class data is collected.



 PF3=Return  7=Back  8=Next  ENTER=Return
```

CICS Monitoring Control Facility (CEMN)

```
CEMN                Help: CICS Monitoring Control Facility

 (3) DESCRIPTION OF FIELDS 2.

 The settings shown on the main panel are:-

  Compression Status        -  Whether monitoring data is to be compressed.

  Converse Status           -  Whether separate performance class records
                               are produced for conversational tasks.

  Syncpoint Status          -  Whether separate performance class records
                               are produced for syncpoint requests.

 PF3=Return  7=Back  8=Next  ENTER=Return
```

CICS Monitoring Control Facility (CEMN)



CICS Monitoring Control Facility (CEMN)

```
CEMN                Help: CICS Monitoring Control Facility

 (2) DESCRIPTION OF FIELDS.

 The settings shown on the option panel are:-

  Frequency                 -  Interval at which CICS produces performance
                               class records for long-running tasks.

  Application Naming Status -  Whether CICS application naming support
                               is enabled.
  RMI Status                -  Whether additional performance monitoring
                               is active for CICS resource managers.

  Time Option               -  Whether the time stamp fields are returned
                               in GMT or local time.




 PF3=Return  7=Back  8=Next  ENTER=Return
```

CICS statistics

- Statistics domain collects a variety of data
- Written to the SMF data set
- Provides information about resources and domains
  - Counts and wait times for resource requests
  - Processor and storage use
- Certain statistics counters can be reset when records cut
- Interval recording can be set on/off using STATRCD (SIT)
- Records can be processed by DFHSTUP or DFH0STAT
- Records can be processed by Tivoli Decision Support for z/OS or CICS Performance Analyzer (CICS PA)

When does CICS collect statistics?

- **Interval statistics**
  - At intervals set: default every hour
  - Requires STARCD=ON in SIT
  - Can be turned on using SET command
- **End-of-day statistics**
  - When CICS shuts down either normal or immediate
  - At midnight (by default) in 24/7 operations
- **Requested statistics**
  - EXEC CICS Perform statistics record
  - EXEC CICS Set statistics RECORDNOW
  - CEMT Perform statistics
  - Can be issued with any combination of resources

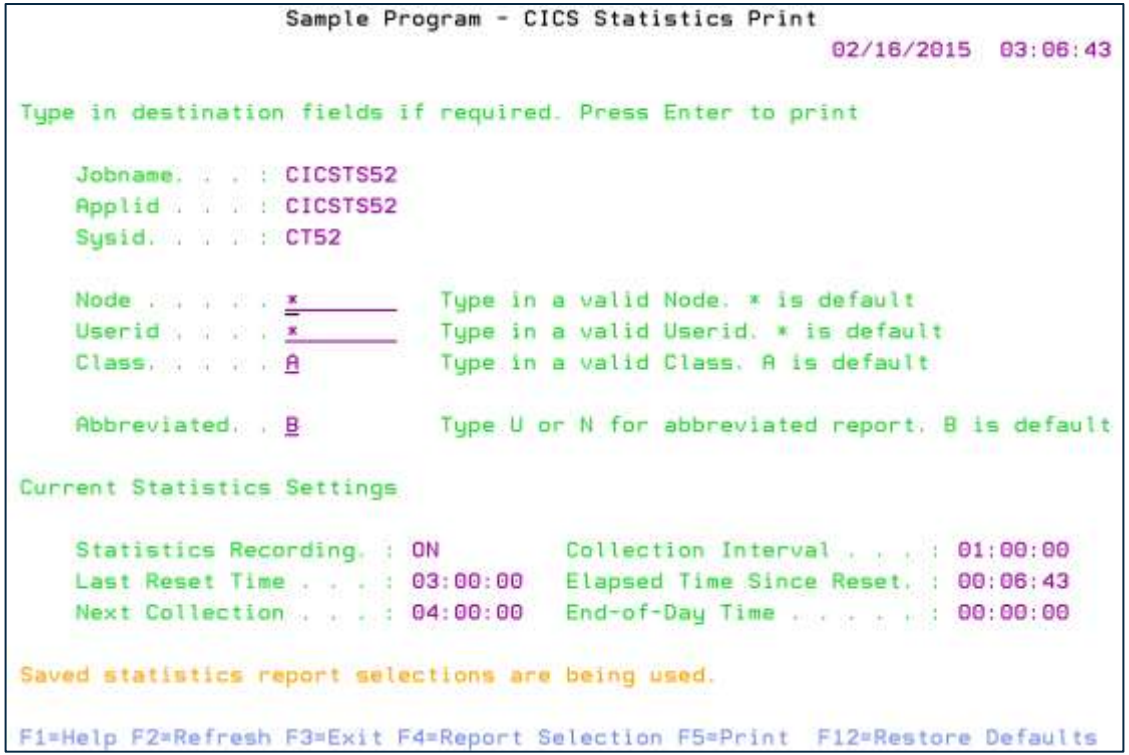
When does CICS collect statistics?

- **Requested Reset statistics**
  - EXEC CICS Perform statistics record RESTNOW
  - EXEC CICS Set statistics RECORDNOW RESETNOW
  - CEMT Perform statistics all RESTNOW
  - Differs from Request Statistics as counters are reset
  - Causes loss of data since the last statistics interval
- **Unsolicited statistics**
  - Collected for resources allocated or de-allocated
  - Written to SMF before resource is deleted
  - Produced for resources such as, Atom Feeds, Autoinstalled Terminals, Files, DB2, FEPI, IPCONN, etc.

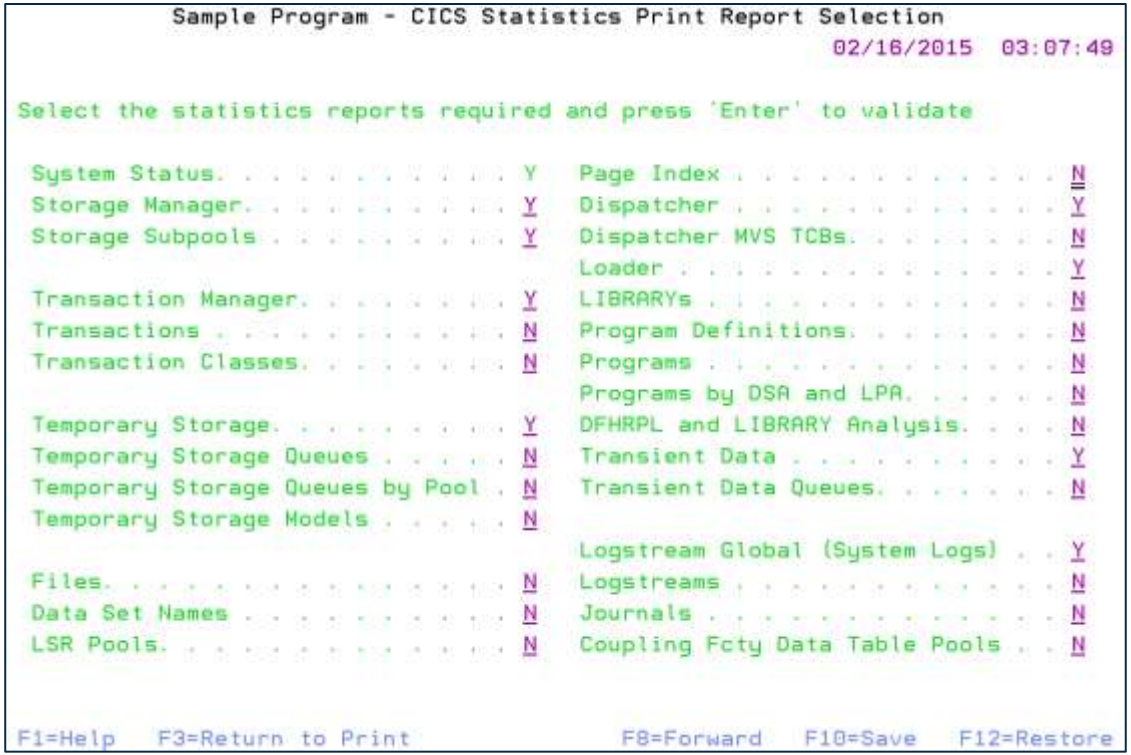
DFHOSTAT

- RDO Group DFH$STAT contains required definitions
- Transaction identifier is STAT
- COBOL sample provided in source and load format
- Good example of the use of EXEC CICS Collect Statistics
- Output information includes:
  - Dispatcher, Storage, Loader, etc.
- Spool = YES must be specified in the SIT

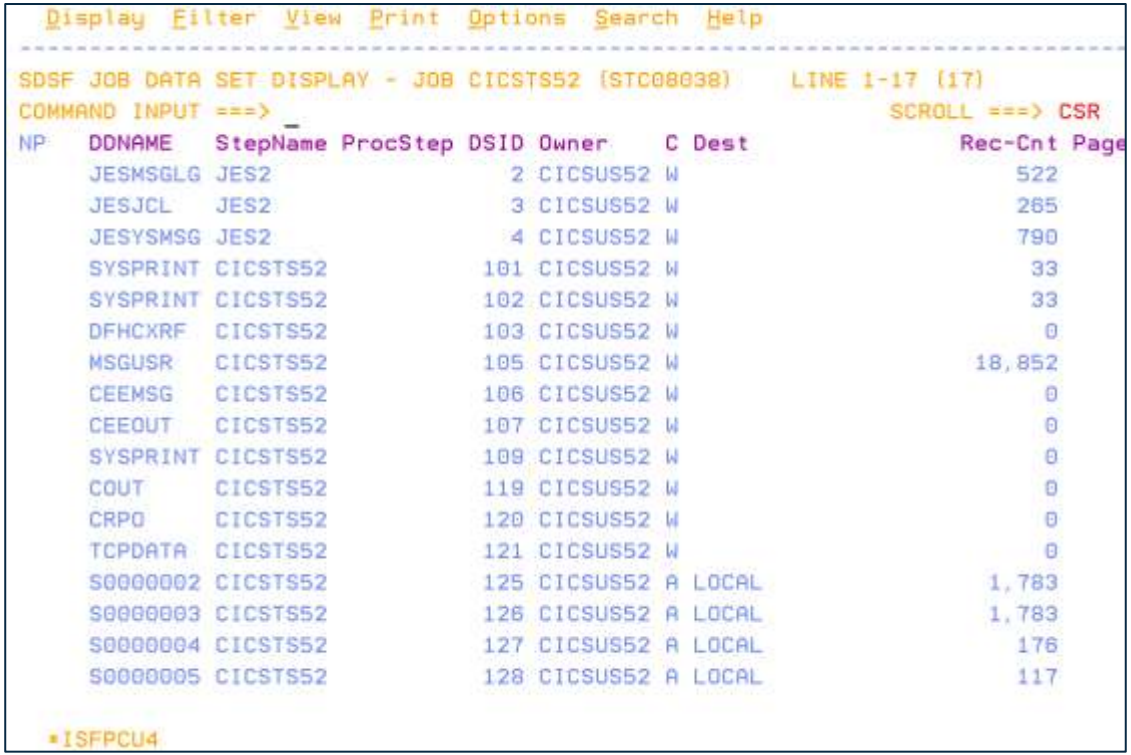
DFHOSTAT


Sample Program - CICS Statistics Print
02/16/2015 03:06:43
Type in destination fields if required. Press Enter to print
Jobname. . . : CICSTS52
Applid . . . : CICSTS52
Sysid. . . . : CT52
Node . . . . . * Type in a valid Node. * is default
Userid . . . . * Type in a valid Userid. * is default
Class. . . . . A Type in a valid Class. A is default
Abbreviated. . B Type U or N for abbreviated report. B is default
Current Statistics Settings
Statistics Recording. : ON Collection Interval . . . : 01:00:00
Last Reset Time . . . : 03:00:00 Elapsed Time Since Reset. : 00:06:43
Next Collection . . . : 04:00:00 End-of-Day Time . . . . . : 00:00:00
Saved statistics report selections are being used.
F1=Help F2=Refresh F3=Exit F4=Report Selection F5=Print F12=Restore Defaults

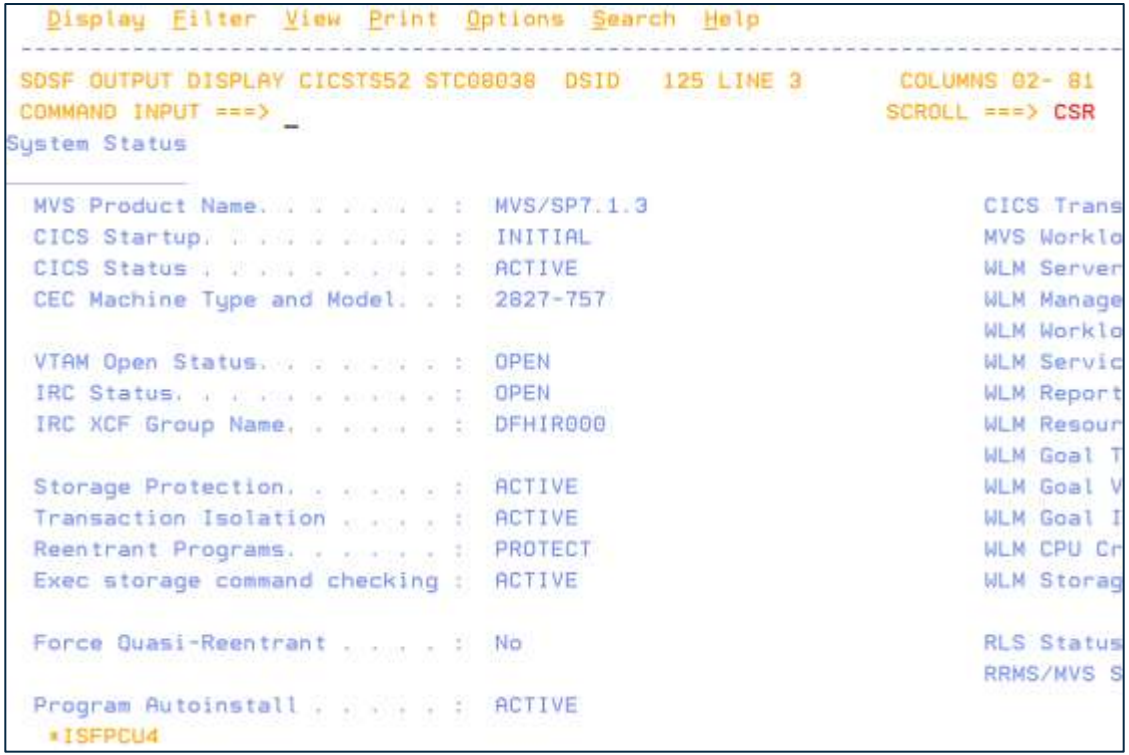
DFHOSTAT


Sample Program - CICS Statistics Print Report Selection
02/16/2015 03:07:49
Select the statistics reports required and press 'Enter' to validate
System Status. . . . . Y
Storage Manager. . . . . Y
Storage Subpools . . . . . Y
Transaction Manager. . . . . Y
Transactions . . . . . N
Transaction Classes. . . . . N
Temporary Storage. . . . . Y
Temporary Storage Queues . . . . . N
Temporary Storage Queues by Pool . N
Temporary Storage Models . . . . . N
Files. . . . . N
Data Set Names . . . . . N
LSR Pools. . . . . N
Page Index . . . . . N
Dispatcher . . . . . Y
Dispatcher MVS TCBs. . . . . N
Loader . . . . . Y
LIBRARYs . . . . . N
Program Definitions. . . . . N
Programs . . . . . N
Programs by DSA and LPA. . . . . N
DFHRPL and LIBRARY Analysis. . . . N
Transient Data . . . . . Y
Transient Data Queues. . . . . N
Logstream Global (System Logs) . . Y
Logstreams . . . . . N
Journals . . . . . N
Coupling Fcty Data Table Pools . . N
F1=Help F3=Return to Print F8=Forward F10=Save F12=Restore

DFHOSTAT


Display Filter View Print Options Search Help
SDSF JOB DATA SET DISPLAY - JOB CICSTS52 (STC08038) LINE 1-17 (17)
COMMAND INPUT ===> SCROLL ===> CSR
NP DDNAME StepName ProcStep DSID Owner C Dest Rec-Cnt Page
JESMSGLG JES2 2 CICSUS52 W 522
JESJCL JES2 3 CICSUS52 W 265
JESYSMSG JES2 4 CICSUS52 W 790
SYSPRINT CICSTS52 101 CICSUS52 W 33
SYSPRINT CICSTS52 102 CICSUS52 W 33
DFHCXRF CICSTS52 103 CICSUS52 W 0
MSGUSR CICSTS52 105 CICSUS52 W 18,852
CEEMSG CICSTS52 106 CICSUS52 W 0
CEEOUT CICSTS52 107 CICSUS52 W 0
SYSPRINT CICSTS52 109 CICSUS52 W 0
COUT CICSTS52 119 CICSUS52 W 0
CRPO CICSTS52 120 CICSUS52 W 0
TCPDATA CICSTS52 121 CICSUS52 W 0
S0000002 CICSTS52 125 CICSUS52 A LOCAL 1,783
S0000003 CICSTS52 126 CICSUS52 A LOCAL 1,783
S0000004 CICSTS52 127 CICSUS52 A LOCAL 176
S0000005 CICSTS52 128 CICSUS52 A LOCAL 117
*ISFPCU4

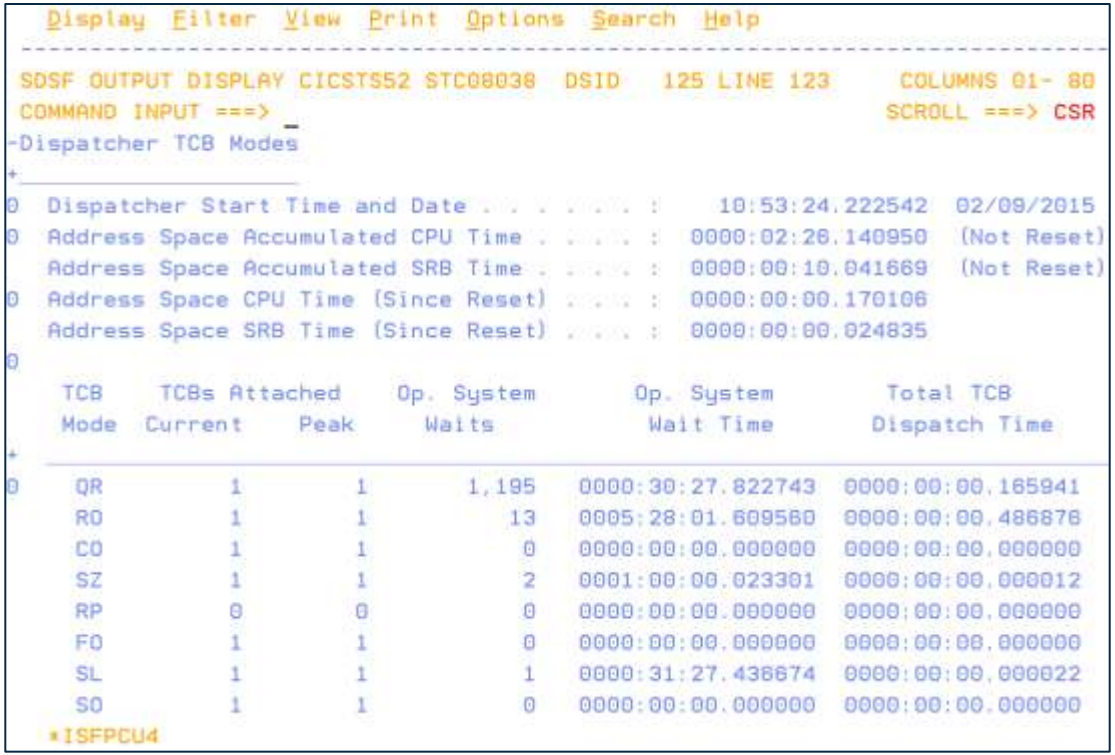
DFHOSTAT


Display Filter View Print Options Search Help
SDSF OUTPUT DISPLAY CICSTS52 STC08038 DSID 125 LINE 3 COLUMNS 02- 81
COMMAND INPUT ===> SCROLL ===> CSR
System Status
MVS Product Name. . . . . : MVS/SP7.1.3
CICS Startup. . . . . : INITIAL
CICS Status . . . . . : ACTIVE
CEC Machine Type and Model. . : 2827-757
VTAM Open Status. . . . . : OPEN
IRC Status. . . . . : OPEN
IRC XCF Group Name. . . . . : DFHIR000
Storage Protection. . . . . : ACTIVE
Transaction Isolation . . . . : ACTIVE
Reentrant Programs. . . . . : PROTECT
Exec storage command checking : ACTIVE
Force Quasi-Reentrant . . . . : No
Program Autoinstall . . . . . : ACTIVE
*ISFPCU4
CICS Trans
MVS Worklo
WLM Server
WLM Manage
WLM Worklo
WLM Servic
WLM Report
WLM Resour
WLM Goal T
WLM Goal V
WLM Goal I
WLM CPU Cr
WLM Storag
RLS Status
RRMS/MVS S

DFHOSTAT

```
  Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
SDSF OUTPUT DISPLAY CICSTS52 STC08038  DSID   125 LINE 101      COLUMNS 01- 80
COMMAND INPUT ===> _                                          SCROLL ===> CSR
-Dispatcher
+_________
0  Current ICV time . . . . . . . . . . . . . :          10,000ms
   Current ICVR time. . . . . . . . . . . . . :             500ms
   Current ICVTSD time. . . . . . . . . . . . :               0ms
   Current PRTYAGING time . . . . . . . . . . :          32,768ms
0  MRO (QR) Batching (MROBTCH) value. . . . . :                 1
0  Concurrent Subtasking value. . . . . . . . :                 1
-  Current number of CICS Dispatcher tasks. . :                31
   Peak number of CICS Dispatcher tasks . . . :                33
0  Current number of TCBs attached. . . . . . :                13
   Current number of TCBs in use. . . . . . . :                10
0  Last Excess TCB Scan . . . . . . . . . . . : 02:27:36.146154    02/16/2015
   Number of Excess TCB Scans . . . . . . . . :                 6
   Last Excess TCB Scan - No TCB Detached . . : 02:27:36.146154    02/16/2015
   Excess TCB Scans - No TCB Detached . . . . :                 6
   Number of Excess TCBs Detached . . . . . . :                 0
   Average Excess TCBs Detached per Scan. . . :                 0
-  Number of CICS TCB MODEs . . . . . . . . . :                18
   *ISFPCU4
```

DFHOSTAT

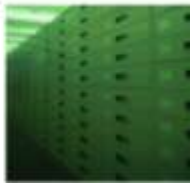


```
  Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY CICSTS52 STC08038  DSID   125 LINE 123     COLUMNS 01- 80
 COMMAND INPUT ===> _                                         SCROLL ===> CSR
-Dispatcher TCB Modes
+
0  Dispatcher Start Time and Date . . . . . :    10:53:24.222542  02/09/2015
0  Address Space Accumulated CPU Time . . . . :  0000:02:26.140950  (Not Reset)
   Address Space Accumulated SRB Time . . . . :  0000:00:10.041669  (Not Reset)
0  Address Space CPU Time (Since Reset) . . . :  0000:00:00.170106
   Address Space SRB Time (Since Reset) . . . :  0000:00:00.024835
0
    TCB     TCBs Attached     Op. System       Op. System        Total TCB
    Mode  Current     Peak      Waits          Wait Time       Dispatch Time
+
0     QR          1         1       1,195   0000:30:27.822743  0000:00:00.165941
      RO          1         1          13   0005:28:01.609560  0000:00:00.486878
      CO          1         1           0   0000:00:00.000000  0000:00:00.000000
      SZ          1         1           2   0001:00:00.023301  0000:00:00.000012
      RP          0         0           0   0000:00:00.000000  0000:00:00.000000
      FO          1         1           0   0000:00:00.000000  0000:00:00.000000
      SL          1         1           1   0000:31:27.436674  0000:00:00.000022
      SO          1         1           0   0000:00:00.000000  0000:00:00.000000
   *ISFPCU4
```

What resources to tune

Ceaseless In Consuming Storage

Virtual Storage
Below/Above the Line
Above the Bar
Above/Below Common Storage
Private Area
LSQA
Dynamic Storage Areas
Real Storage
Paging
CICS Modules In LPA



What resources to tune

What resources to tune

Buffering Techniques

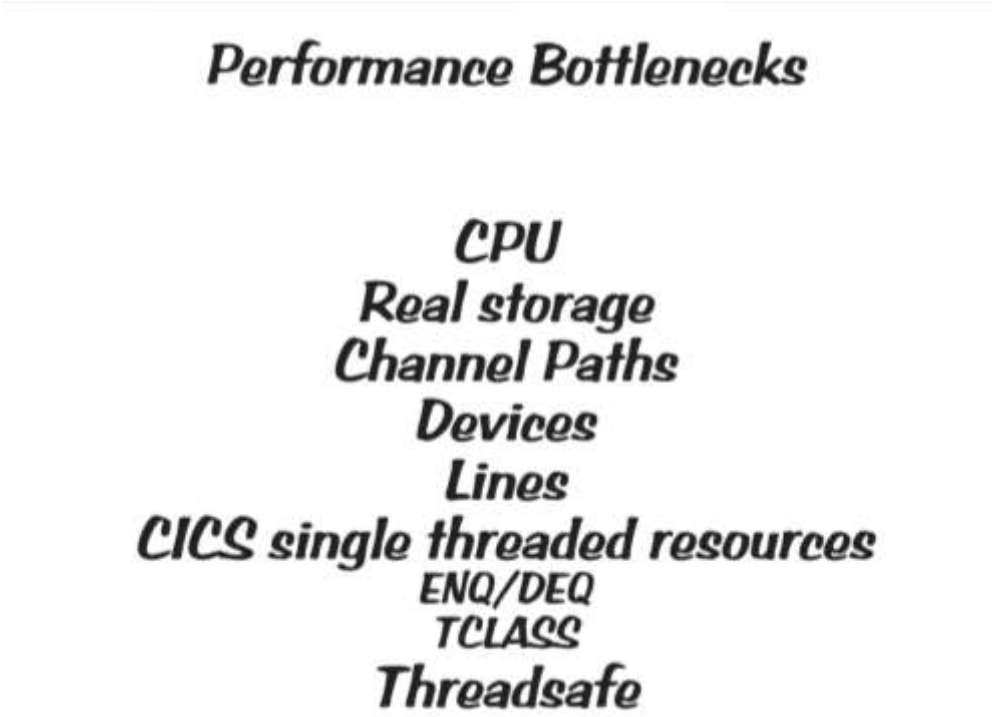
File Control
Temporary Storage
Transient Data
LOGGER
VSAM LSR
Program Loader
DB2 and DL1 Interfaces
VSAM RLS

What resources to tune

Network Considerations

Data Stream
Compression
Application Design
Correct Use Of MDTs
Web Considerations

Understanding serialization



Making a change

Do not tune for the sake of it
Tuning can be counter productive

Major constraints first
Make one change at a time

Do large changes gradually
Monitor at regular Intervals

Look at systems end - Check CICS thresholds
Look at applications end - During design phase

Develop Goals For Tuning Your System

Making a change

Statistics

XM Statistics
TCLASS Statistics
SM Statistics
VTAM Statistics
FCP Statistics
TS Statistics
TD Statistics
DS Statistics
TCP/IP Statistics
DB2 Statistics

Maximum Tasks (MXT)

Possible Starting Point:

MXT = ((Transactions/Second * 1.50) +
(# of Long Running Transactions * 1.25) +
(#of Conversational Transactions) + 25)

If Transaction/Second <1, use 6.00
If Conversational Transactions <1, use 4
If MXT <40, Use 40

MXT is used to determine the number of
Performance Blocks (PB) generated:

Total PB = MXT + estimated number of System Tasks + 1

Maximum Tasks (MXT)

Possible Solutions

Increase REGION size on JCL (REGION=0M preferred)

Increase current EDSALIM by a minimum of ((MXT * 1 MB) * 2)

Use TCLASS to control high volume transactions below the line (Next topic)

Maximum Tasks (MXT)

```
Applid CICSTS51  Sysid CT51  Jobname CICSTS51     Date 02/19/2014  Time 10:17:07     CICS 6.8.0     PAGE 3

Transaction Manager

   Total Accumulated transactions so far . :          2,290

   Accumulated transactions (since reset). :              8   Transaction Rate per second. :      0.00

   Maximum transactions allowed (MXT). . . :            100
   Times at MXT. . . . . . . . . . . . . . :              0
   Current Active User transactions. . . . :              3
   Peak Active User transactions . . . . . :              3
   Total Active User transactions. . . . . :              7

   Current Running transactions. . . . . . :              1
   Current Dispatchable transactions . . . :              0
   Current Suspended transactions. . . . . :              2
   Current System transactions . . . . . . :              0

   Transactions Delayed by MXT . . . . . . :              0
   Total MXT queueing time . . . . . . . . : 00:00:00.00000
   Average MXT queueing time . . . . . . . : 00:00:00.00000

   Current Queued User transactions. . . . :              0
   Peak Queued User transactions . . . . . :              0
   Total Queueing time for current queued. : 00:00:00.00000
   Average Queueing time for current queued: 00:00:00.00000
```

Transaction Class (TCLASS)

Five reasons for using TCLASS

Controlling resource "hogs"
Single threading to protect resources
Control number of transactions below the line
Avoid MRO "sympathy sickness"
Uncontrollable Input

Transaction Class (TCLASS)

How do you measure the effectiveness of a TCLASS?

Why did you use TCLASSes?

- Resource Hog
- Single Thread
- MRO Sympathy Sickness
- Control of SOS below the line

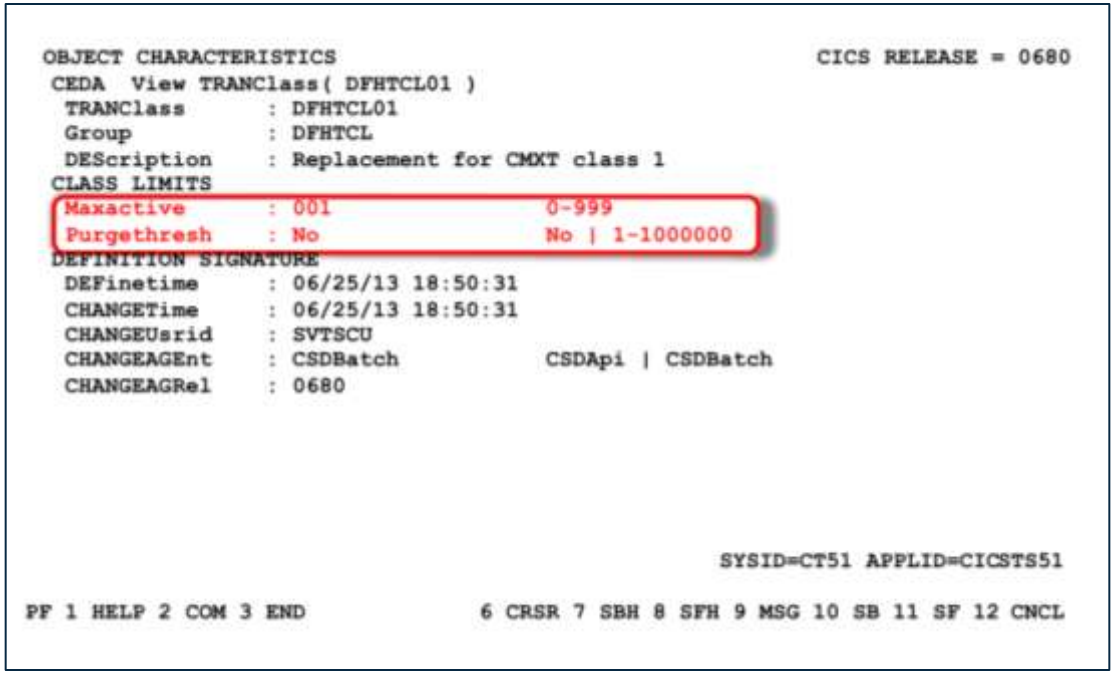
Have any tasks waited?

If so, were resources available?

- CPU
- Real/Virtual Storage
- I/O devices

How does the peak number of tasks compare to the maximum number allow in the class?

Transaction Class (TCLASS)



Task Priority & PRTYAGE (SIT)

Sample PRIORITY SETTINGS

255 RESERVED FOR SYSTEM TASKS
254 HIGHEST EVER USER PRIORITY
120 MENUS
110 DATA ENTRY TASKS / TIME CRITICAL
100 EVERYTHING ELSE
 90 BROWSE TYPES
 80 CPU CRUNCHERS WITH EXEC CICS SUSPEND

Task Priority & PRTYAGE (SIT)

Use priority settings sparingly and should be set aside by classes, e.g.,

- System Tasks -> high 250 to 255
- Normal Tasks -> 1
- Certain Important Tasks -> 5

If you are going to use PRTYAGE, do not use wide ranges for tasks.

- PRTYAGE only increments priority by one at the end of the PRTYAGE period

Task Priority & PRTYAGE (SIT)

Recommendation

In general, only use transaction priority for system transactions.

If used, base priority on transaction and person, not terminal.

If used, do not use wide ranges of priorities when using PRTYAGE.

Task Priority & PRTYAGE (SIT)

Question that has to be answered is:

Why am I using transaction priorities?

If the answer is to favor important transactions, then why should I want to dispatch a lower priority transaction over important transactions?

The system has to be running at a very high CPU utilization for a task not to get dispatched.

So, the question is:

Why should I want to use precious CPU cycles to worry about the dispatching priority of low priority tasks?

Why not use these cycles to dispatch tasks instead of

Region Exit Interval (ICV)

Determine the maximum amount of time CICS/TS will "sleep" When there is no work to run

Not material in very busy systems

CPU cycles can be wasted if set incorrectly for low activity systems and/or periods

Interval Runaway (ICVR)

```
 OBJECT CHARACTERISTICS                                        CICS RELEASE = 0680
  CEDA  View TRANSaction( TREK )
   TRANSaction     : TREK
   Group           : CTREK
   DEScription     :
   PROGram         : KVPKVEW
   TWasize         : 00000              0-32767
   PROFile         : DFHCICST
   PArtitionset    :
   STAtus          : Enabled            Enabled | Disabled
   PRIMedsize      : 00000              0-65520
   TASKDATALoc     : Any                Below | Any
   TASKDATAKey     : Cics               User | Cics
   STOrageclear    : No                 No | Yes
   RUnaway         : 0010000            System | 0 | 500-2700000
   SHutdown        : Disabled           Disabled | Enabled
   ISolate         : Yes                Yes | No
   Brexit          :
+ REMOTE ATTRIBUTES

                                                    SYSID=CT51 APPLID=CICSTS51

PF 1 HELP 2 COM 3 END            6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

Interval Runaway (ICVR)

RECOMMENDATIONS

Set the ICVR to less than one second in today's modern processors

If a transaction cancels with an AICA abend, consider assigning a separate value for this transaction on the transaction RDO definitions.

RDO parameter RUNAWAY defaults to "SYSTEM" that uses the SIT ICVR value.

However, if you code a value (500 - 2700000) instead of using "SYSTEM", then CICS will use the assigned value for the specific transaction.

Multi-Tasking (DTIMEOUT)

Giving up control:

DSSR WAIT_MVS	Extended External ECB	VSAM I/O JOURNAL I/O
DSSR WAIT_OLDW	Non-Extended External ECB	JOURNAL FUNCTIONS
DSSR WAIT_OLDC	Internal CICS ECB	TS BUFFERS TS STRINGS
DSSR SUSPEND	Wait for CICS resource	STORAGE ENQ

Deadlock Timeout (DTIMEOUT)

RECOMMENDATION

Is to specify DTIMOUT and SPURGE for all transactions that do not have any updates to resources such as:

Inquiry
Menu
Browse
Best Defense Against stall Conditions
SOS

DISPATCHER STATISTICS

TRANSACTION MANAGER STATISTICS

Total number of transaction (user + system)	77135
Current MAXTASKS limit	30
Current Number of active user transactions	1
Current number MAXTASK queued user transactions	0
Times the MAXTASKS Limit reached	1
Peak number of MAXTASK queued user transaction.	0
Peak number of active active user transaction	30
Total number of active user transactions.	7717
Total number of MAXTASK delayed user transactions	0
Total MAXTASK queuing time	000-00:00:00
Total MAXTASK queuing time of currently queued user transaction:	00:00:00

DISPATCHER STATISTICS

Start time.	06:45:29
Current Number of tasks	23
Peak number of tasks.	44
Current ICV time (msec).	1000
Current ICVTSD time (msec).	500

TCB STATISTICS

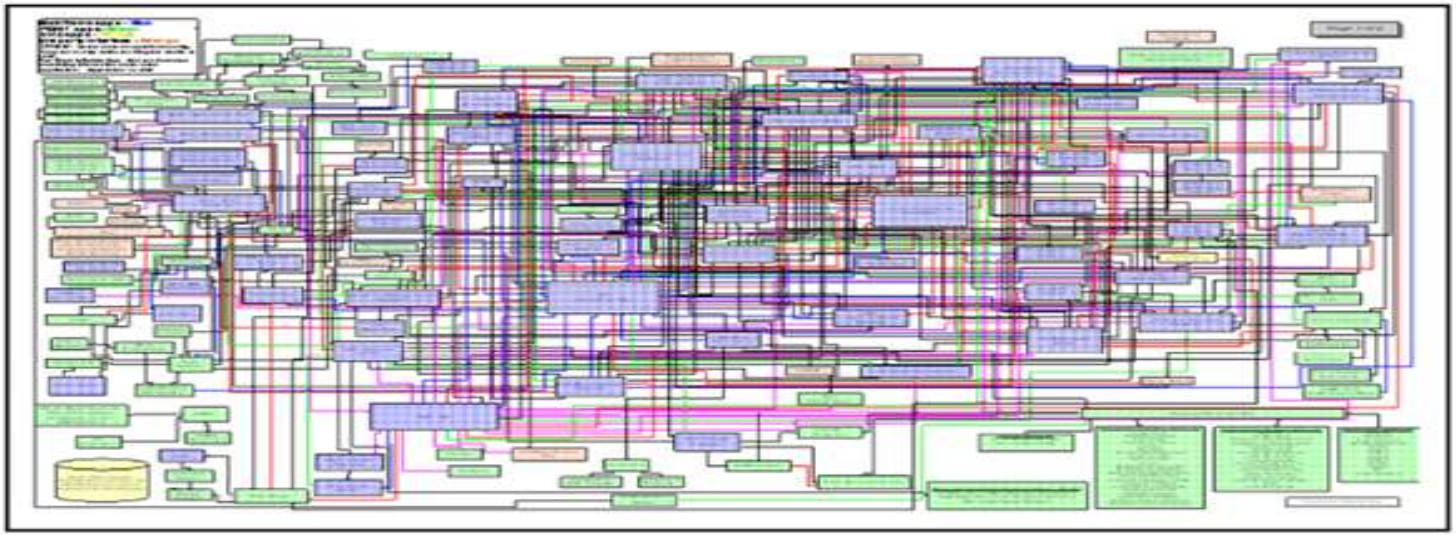
Mode	MVS Waits	Accum Time In MVS Wait	Acccum Time Dispatched	Accum CPU Time/TCB
QUASI	2569503	13:18:16.2439	00:53:59.1999	00:52:53.7309
RESOURCE	4990	14:00:43.4698	00:11:31.9738	00:01:49.5175

Systems of Record can be complex systems built on decades of continuous & incremental development

Change?

Fix problems?

Optimize?

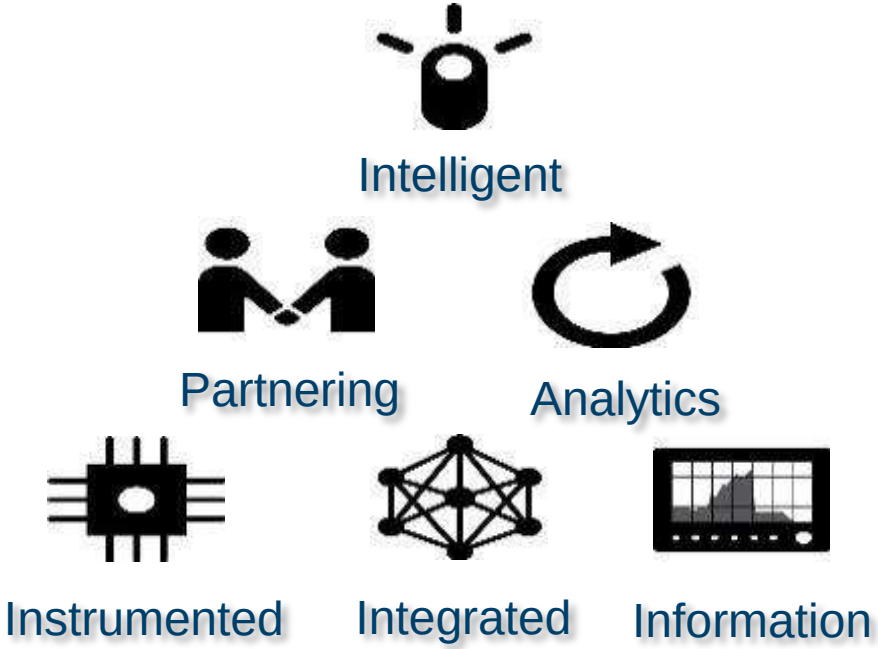


Modernize?

Where do I start?

Re-use?

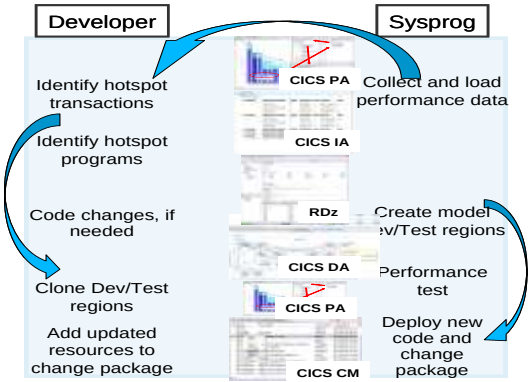
CICS Operational Insight

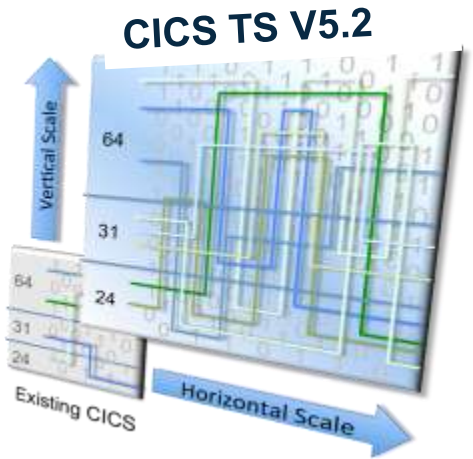


Try it out at
https://cicsoi.mybluemix.net/
Have your say on new insights on
https://ibm.biz/cicsoi-forum

Three steps to CICS Operational Efficiency

Optimize applications, systems, and processes to achieve more with less





Step 1

Threadsafe analysis and implementation to reduce CPU utilization by up to 20%

Step 2

Workload Management to cope with mobile scalability and availability needs

Step 3

Region **Consolidation** to reduce management overhead and CPU utilization (up to 10%)

Summary

- There are many areas in CICS that can be tuned
- To achieve the best results, tuning must be on-going
- Always start at the top and work your way down
- Set reasonable objectives
- Measure and publish the results