Getting Started with Performance of MQ on z/OS

17899, Dolphin, Oceanic 5, Weds Aug 12th 2015, 1:45 - 2:45pm

Mayur Raja
IBM Hursley Park, Winchester, UK
mayur_raja@uk.ibm.com

Legal Disclaimer

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Please Note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice, at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

MQ sessions this week

	Monday	Tuesday	Wednesday	Thursday	Friday
08:30			MQ for z/OS, Using and Abusing New Hardware and the New v8 Features	Nobody Uses Files Any More Do They? New Technologies for Old Technology, File Processing in MQ MFT and IIB	Monitoring and Auditing MQ Securing MQ Initiated CICS Workload
10:00	Introduction to MQ - Can MQ Really Make My Life Easier?	MQ for z/OS: The Insider Story	IBM Integration Bus MQ Flexibility	Common Problems and Problem Determination for MQ z/OS	IBM MQ and IBM Integration Bus - from Migration and Maintenance to Continuous Enhancements, How and Why to Stay Current
11:15	Introduction to IBM Integration Bus on z/OS	Introduction to the New MQ Appliance	MQ V8 Hands-on Labs! MQ V8 with CICS and COBOL! MQ SMF Labs!		
12:15					
1:45	What's New in the Messaging Family - MQ v8 and More		Getting Started with Performance of MQ on z/OS	IBM MQ: Are z/OS & Distributed Platforms Like Oil & Water?	
3:15	What's New in IBM Integration Bus	Live!: End to End Security of My Queue Manager on z/OS Application Programming with MQ Verbs	Digging into the MQ SMF Data	MQ Parallel Sysplex Exploitation, Getting the Best Availability from MQ on z/OS by Using Shared Queues	
4:30	MQ Security: New v8 Features Deep Dive	Live!: What's the Cloud Going to Do to My MQ Network? The Do's and Don'ts of IBM Integration Bus Performance	Giving It the Beans: Using IBM MQ as the Messaging Provider for JEE Applications in IBM WebSphere Application Server	Challenge the MQ & IIB Experts?	

Getting Started with Performance of MQ on z/OS

- Audience:
  - MQ Administrators
  - MQ System Programmers
  - z/OS System Programmers
- Familiarity with:
  - MQ
  - Buffer pools
  - Page sets
  - CF Structures
  - SMDS
  - Logs

Notes

Agenda

- Getting started with performance on MQ for z/OS
- Factors to consider when looking at performance issues
- Tools plus techniques for cheking CPU usage
- What can cause high CPU usage in MQ ?
- What can cause delays ?
- Some useful commands

Getting Started with Performance of MQ on z/OS

- An "MQ performance problem" is …

  … usually a manifestation of a problem occurring elsewhere

- You need to prove whether MQ really is at fault or not

- So, what factors should you consider ?

- And, where should you start to look ?

Factors to consider when looking at performance

- Resource Usage
  - CPU utilization
  - Disk I/O (Response time / Throughput)
  - Storage/Buffer Pools
    - Paging ?
  - CF usage
  - Network
  - MQ Channels
- Application design considerations
  - MQ Online V's Batch workloads
  - MQ queue usage
- Metrics
  - End to end response times
  - Cost

Factors to consider when looking at performance

- There are many areas that can affect MQ performance. CPU utilization, disk activity, storage and buffer pool usage. Are you experiencing paging ? Can you add additional buffers ? If you are using CF then how well is your CF being utilised ? Are you seeing network issues ? Do you have adequate TCP/IP buffers defined ?

- One customer said 'You page – you die'.

- Metrics tell you how well your systems are running – are there any problems ?
  - Most common metrics are:
    - End to end response time.
    - Cost.

Notes

Factors to consider when looking at performance

- Did the problem start to occur now or sometime ago ?

- Did you make a recent change to your environment?

- Are your MQ systems meeting their SLAs
  - Is the response time criteria being met ?
    - If not look at where the time is being spent
  - Is it taking longer to empty queues ?
    - Are jobs running slower ?
      - Are messages building up on Queues ?
    - Are jobs using more CPU ?
      - Could they be looping ?
    - Is MQ using more CPU ?

Determining CPU usage

- RMF (or other MVS tools) report CPU used by Jobs

```
                    RMF - Performance Management                   z/OS V2R1 RMF
Selection ===>

Enter selection number or command on selection line.

  1 Postprocessor   Postprocessor reports for Monitor I, II, and III      (PP)
  2 Monitor II      Snapshot reporting with Monitor II                    (M2)
  3 Monitor III     Interactive performance analysis with Monitor III     (M3)

  U USER            User-written applications (add your own ...)          (US)

  R RMF SR          Performance analysis with the Spreadsheet Reporter
  P RMF PM          RMF PM Java Edition
  N News            What's new in z/OS V2R1 RMF

                              T TUTORIAL     X EXIT

  RMF Home Page:    http://www.ibm.com/systems/z/os/zos/features/rmf/

              5650-ZOS Copyright IBM Corp. 1994, 2013.
               Licensed Materials - Property of IBM
```

Notes

Resource Measurement Facility (RMF) is a performance monitor for the z/OS Operating System. It also collects data for long-term performance analysis and capacity planning. The product consists of the following components:

Monitor I Data Gatherer which collects data in adjustable intervals from one minute to one hour. The data is written to SMF data sets. *RMF Monitor I* uses the SMF data record types 70 to 78. In addition a dataspace can be created for the ***Monitor I Data Gatherer*** in which the most recent RMF SMF data records are buffered so that the RMF Postprocessor can access and process them immediately.
Monitor III Data Gatherer collects data for short-term and immediate data analysis. The data is collected in intervals ranging from 10 seconds to 10 minutes. The data is written to *RMF Monitor III* VSAM data sets and internally recorded in a wraparound buffer. The data gatherer also collects some special data for long-term data analysis and records them in RMF SMF records too.
Monitor II is the original snap shot monitor of RMF. This function also records information in RMF SMF record type 79 and provides interfaces for other reporting functions like SDSF. Over the years this function became less important and is today mostly used as a data provider for other reporting components.
RMF Postprocessor is the reporting tool for RMF SMF records of type 70 to 78. It generates a big number of tabular and textual reports for long-term data analysis which provide a lot of detail of z/OS and System z performance.
RMF Spreadsheet Reporter is an extension of the *RMF Postprocessor* and displays selected data in spreadsheet applications on a workstation.
RMF Monitor III Reporter is the display tool of *RMF Monitor III* data. The function is written in ISPF and runs under a TSO session on z/OS. The reporter allows to move forward and backward in time and uses *RMF Monitor III* in-storage buffer and data sets to access the collected data.
RMF PM and the ***RMF Webportal*** are workstation enhancements for *RMF Monitor III*. They allow multiple z/OS systems to be displayed in the same data view. *RMF PM* is a Windows-based application and the *Webportal* allows the user to display screens using a frames-capable web browser.
RMF is significantly enhanced with every release of the z/OS Operating System – that is approximately once a year – with other enhancements occurring in support of new hardware, such as new processor models.

Notes

Determining CPU usage ..

- **MONITOR II**
  - RMF address space reports show CPU time

```
                 RMF - ARD Address Space Resource Data              Line 53 of 79
Command ===> _                                                     Scroll ===> PAGE

                          CPU=  4/  5 UIC= 65K PR=   0           System= MV4C Total
```

09:49:23 JOBNAME	DEV CONN	FF 16M	FF 2G	PRIV FF	LSQA CSF	X M	C R	SRM ABS	TCB TIME	CPU TIME	EXCP RATE	SWAP RATE	LPA RT	CSA RT	NVI RT	V&H RT
DB4CIRLM	0.023	0	22	179	18K	X		0.0	0.38	14799	----	-----	---	---	---	---
RMF	29.68	0	50	58	104	X		0.0	1538	1837	----	-----	---	---	---	---
RSED	0.107	0	89	13	240			0.0	417.8	493.0	----	-----	---	---	---	---
SMTP	3.390	0	28	15	93			0.0	29.37	33.99	----	-----	---	---	---	---
NET	0.640	0	43	61	173	X		0.0	464.8	1520	----	-----	---	---	---	---
DB4CADMT	23.00	0	165	15	316	X		0.0	374.1	492.1	----	-----	---	---	---	---
RMFGAT	175.8	0	100	27	111	X		0.0	101K	101K	----	-----	---	---	---	---
ASCH	0.004	0	53	12	74	X		0.0	33.47	43.34	----	-----	---	---	---	---
ARHJCOMM	13.52	0	28	55	1553	X		0.0	422.9	592.4	----	-----	---	---	---	---
OMEGM2	5.207	0	46	29	152			0.0	0.95	1.63	----	-----	---	---	---	---
AXR	0.022	0	23	17	111	X		0.0	0.03	0.07	----	-----	---	---	---	---
RSED4	0.036	0	134	13	217			0.0	456.1	516.5	----	-----	---	---	---	---
IZUSVR1	3.700	0	740	688	441			0.0	3909	3981	----	-----	---	---	---	---
BUZAGNT1	0.041	0	144	200	243			0.0	996.0	1039	----	-----	---	---	---	---
BUZAGNT1	0.041	0	103	102	172			0.0	899.0	923.3	----	-----	---	---	---	---
NETVAUTO	0.352	0	29	15	161			0.0	20.23	36.92	----	-----	---	---	---	---
NETVSSI	0.008	0	19	4	75	X		0.0	0.00	0.04	----	-----	---	---	---	---
OMEGTEMS	114.2	1	345	304	683			0.0	2991	3020	----	-----	---	---	---	---
DFRMM	0.645	0	25	15	155	X		0.0	0.93	1.04	----	-----	---	---	---	---
DFHSMS	68.55	0	27	23	166			0.0	18.72	25.63	----	-----	---	---	---	---
OMEGCN	0.041	0	22	30	136	X		0.0	5.08	8.41	----	-----	---	---	---	---
OMEGTOM	0.462	1	140	96	218			0.0	27041	27055	----	-----	---	---	---	---
NETVIEW	380.4	0	282	63	512	X		0.0	22675	22847	----	-----	---	---	---	---
ZCT1MSTR	0.350	0	769	91	952	X		0.0	73.49	85.84	----	-----	---	---	---	---
OMEGM2CS	0.019	0	64	11	125	X		0.0	8.22	9.06	----	-----	---	---	---	---
MAYUR	0.601	0	36	20	111			0M	1.13	1.15	----	-----	---	---	---	---
ZCT1CHIN	0.263	0	100	21	267			0.0	36.77	56.75	----	-----	---	---	---	---

Determining CPU usage ..

TCB TIME

The number of seconds of TCB processor time used by the current job step.

Depending on your choice, this field can contain either a total value for the RMF session or a changed (delta) value since the report request.

CPU TIME

The amount of processor (TCB + SRB) time, in seconds, for the current job step, in the form 'nnn.dd', where 'nnn' represents the number of seconds and 'dd' represents hundredths of a second.

Depending on your choice, this field can contain either a total value for the current job step or a changed (delta) value since the last report request.

Note: In the example report on the previous slide we see the total values.

Notes

Determining CPU usage ..

- **Monitor III**
  - RMF PROCU report shows processor usage by jobname

```
                      RMF V2R1   Processor Usage                      Line 1 of 27
Command ===> _                                                    Scroll ===> CSR

Samples: 100     System: MV4C  Date: 08/12/15  Time: 10.03.20  Range: 100    Sec

                  Service      --- Time on CP % ---    ----- EAppl % -----
Jobname    CX  Class        Total    AAP     IIP        CP     AAP     IIP

RMFGAT     SO  STC          1.000  0.000   0.000     1.000
OMEGTEMS   SO  STCUSER      0.850  0.000   0.000     0.850
NETVIEW    SO  STCFAST      0.619  0.000   0.000     0.619
XCFAS      S   SYSTEM       0.449  0.000   0.000     0.449
GRS        S   SYSTEM       0.180  0.000   0.000     0.180
DB4CIRLM   S   SYSSTC       0.119  0.000   0.000     0.119
WLM        S   SYSTEM       0.106  0.000   0.000     0.106
ZCT1MSTR   S   SYSSTC       0.096  0.000   0.000     0.096
JES2       S   SYSSTC       0.083  0.000   0.000     0.083
CATALOG    S   SYSTEM       0.065  0.000   0.000     0.065
ZFS        S   SYSSTC       0.065  0.000   0.000     0.065
IZUSVR1    SO  STC          0.045  0.000   0.000     0.045
DB4CMSTR   S   SYSSTC       0.039  0.000   0.000     0.039
OMVS       S   SYSTEM       0.036  0.000   0.000     0.036
TCPIP      SO  SYSSTC       0.030  0.000   0.000     0.030
CFZCIM     SO  STC          0.019  0.000   0.006     0.019
RRS        S   STC          0.015  0.000   0.000     0.015
*MASTER*   S   SYSTEM       0.008  0.000   0.000     0.008
CONSOLE    S   SYSTEM       0.006  0.000   0.000     0.006
BUZAGNT1   O   OMVSFAST     0.006  0.000   0.000     0.006
MAYUR      T   TSO          0.005  0.000   0.000     0.005
HZSPROC    SO  STCUSER      0.003  0.000   0.000     0.003
DB4CDBM1   S   SYSSTC       0.003  0.000   0.000     0.003
NET        SO  SYSSTC       0.003  0.000   0.000     0.003
SMSPDSE    S   SYSTEM       0.002  0.000   0.000     0.002
SMSPDSE1   S   SYSTEM       0.002  0.000   0.000     0.002
DB4CADMT   SO  STCUSER      0.001  0.000   0.000     0.001
```

Determining CPU usage ..

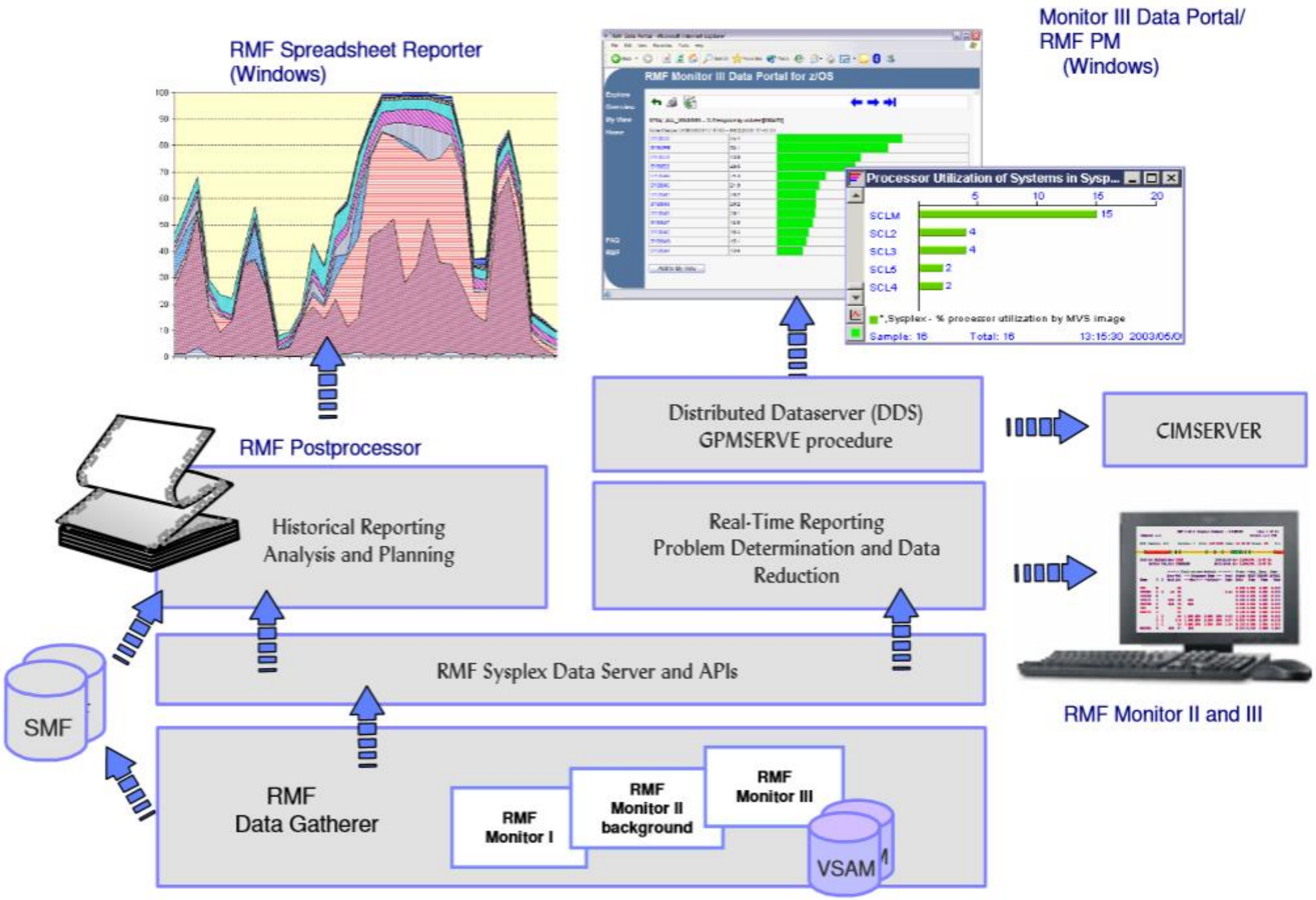
The new Processor Usage report (PROCU) displays all jobs that were using a general or special purpose processor during the report interval. RMF reports the jobs by descending overall CPU time. The report gives you information about the percentage of CPU time on general purpose CPs consumed on behalf of the job. In addition, the percentage of CPU time used by work that is eligible for being offloaded to a Application Assist (zAAP) or Integrated Information (zIIP) processor is shown. This can be used to understand the benefit of adding a zAAP or zIIP into the configuration.

The EAppl fields display the percentage of task, SRB and enclave CPU time consumed within the address space on general (CP) or special (zAAP and zIIP) purpose processors.

The report can be selected with option <1A> from the RMF Resource Report Selection Menu or by the command PROCU.

Notes

RMF Spreadsheet reports and Data Portals



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE in Orlando 2015

RMF Spreadsheet reports and Data Portals

For further details, see:

1) The RMF flyer at:

http://www-03.ibm.com/systems/z/os/zos/features/rmf/product/rmfflyer.html

2) The RMF Spreadsheet Reporter at:

ftp://public.dhe.ibm.com/eserver/zseries/zos/rmf/RMF_SpreadsheetReporter.pdf

3) Effective zSeries Performance Monitoring Using Resource Measurement Facility redbook at:

http://www.redbooks.ibm.com/redbooks/pdfs/sg246645.pd

Notes

How to check CPU usage ?

- **CICS/IMS statistics**
  - Reports CPU used by transaction
  - CICS also reports how much time was used in MQ and in DB2
  - Also, see APAR:

    PM82576: ALLOW CORRELATION OF CICS TYPE 110 SMF RECORDS WITH WEBSPHERE MQ TYPE 116 SMF RECORDS

How to check CPU usage ?

PM82576: ALLOW CORRELATION OF CICS TYPE 110 SMF RECORDS WITH WEBSPHERE MQ TYPE 116 SMF RECORDS

CICS sets a 27 byte network UOWID and extracts from this a 22 byte correlation token which it passes to DB2. DB2 in turn sets it in DB2 Type 101 SMF records.

CICS also passes the 27 byte network UOWID to IBM MQ so that IBM MQ can extract from it the 22 byte correlation token and set it in IBM MQ Type 116 SMF records. MQ sets it in the QWHCTOKN field of the SMF 116 records.

Note: If you have a CICS transaction that issues calls to both MQ and DB2, with this APAR fix you can now correlate activity across all three subsystems.

Notes

How to check CPU usage ? ...

- **SMF data captured by MQ**
  - Queue Manager and CHINIT capture the following:
    - SMF 115 – Statistics records
    - SMF 116 – Accounting records
  - Enabled via:
    - **CSQ6SYSP** macro
      - SMFSTAT=YES
      - SMFACCT=YES
    - **START TRACE** command
      - START TRACE(STAT) DEST(SMF) CLASS(1)
      - START TRACE(ACCTG) DEST(SMF) CLASS(4)
  - SMF data gathering interval is controlled via:
    - **CSQ6SYSP** macro
      - STATIME=0|integer (in mins) - default 30 mins
    - SET SYSTEM STATIME(0|integer) (in mins) command

SMF Statistics data captured by MQ

SMF Record	SubType	Class	Data Captured	Size	CPU overhead	Remarks
115	1	1	Storage and Logs	<1K	Low	Always gather. Useful for trend analysis.
115	2	1	Number of messages, Buffer and paging information, Locks, Queue-Sharing Group information related to the coupling facility and DB2, Topics	<7K	Low	Always gather. Useful for trend analysis.
115	3	2 and 3	Detailed information about storage usage in Queue Manager address space			
115	215	1	Bufferpool Information (IBM MQ for z/OS V8 with OPMODE=NEWFUNC).			
115	231	4	System information for CHIN address space and tasks.		Low	Always gather. Useful for seeing task usage.

- Queue Manager and CHINIT capture the following:

SMF Accounting data captured by MQ

SMF Record	SubType	Class	Data Captured	Size	CPU overhead	Remarks
116	0	1	Time spent processing MQ API calls, number of MQPUT/MQGET calls.			
116	1	3	Data for each task at thread and queue level.	Large	High	Only enable when required. But you may want to run once a month.
116	2	3	Additional queue-level accounting data.	Large	High	Only enable when required. But you may want to run once a month.
116	10	4	Accounting Data for Channels	Large if lots of channels	Low	But, be careful with Clients that connect and disconnect frequently as this will result in a lot of data being gathered.

- Run 116 Class 3 at least once a month to gather data
    - Useful for comparing data from a good data with data from a bad day !

SMF Statistics and Accounting data

Data captured for each class:

- Stats class(1) - basic queue stats (count of messages, log stats etc.)
- Stats class(4) - Chinit stats (TCB information etc.)
- Accounting class(1) - number of messages put or got, total CPU time
- Accounting class(3) - task and queue data (CPU used, elapsed timed, where the delays are)
- Accounting class(4) – data like Display Channel Status for each channel

When Accounting data is captured:

- Accounting class(1)
  - When task ends –this could be days in duration
- Accounting class(3)
  - When statistics are collected, or when task ends.
- Accounting class(4)
  - When statistics are collected, when a channel ends, or when the CHINIT ends

Notes

SMF 115 (subtypes 1 and 2) Statistics

- Two records cut per SMF interval per queue manager
  - SMF 115 subtype 1
    - Storage Manager and Log Manager
  - SMF 115 subtype 2
    - Buffer Manager, Message Manager, Data Manager, CF Manager, DB2 Manager, Topic Manager, Lock Manager
- Lightweight (subtype 1 < 1K, subtype 2 < 7K)
- Negligible CPU cost

- **Recommendation:**
  - Always gather and examine this data
  - Useful to store for trend analysis

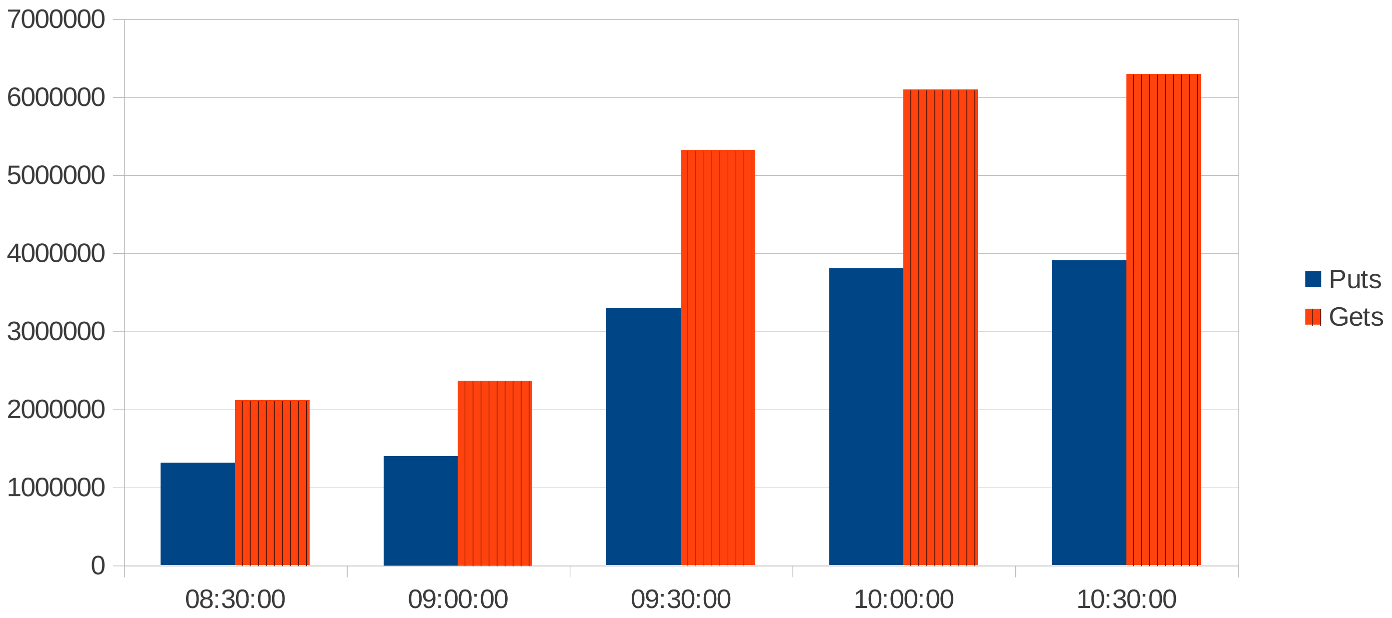
SMF 115 (Subtype 2) Message Manager Statistics

- MQ API Call counts

```
Message manager : QMST
  MQOPENs   374549,  MQCLOSEs   375694,  MQGETs   5014956, MQPUTs     4564331
  MQPUT1s    89707,  MQINQs      88650,  MQSETs         0, Close_all        0
  MQSUBs         0,  MQSUBRQs        0,  MQCBs          0, MQCBs            0
  MQCTLs         0,  MQSTATs         0,  Publish        0
```

Monitor Puts and Gets

- Number of MQPUT, MQPUT1 and MQGET requests



- Number of MQGETs > Number of MQPUTs + MQPUT1s
  - Servers waiting for messages
  - Applications in MQGET with wait, waiting for reply messages

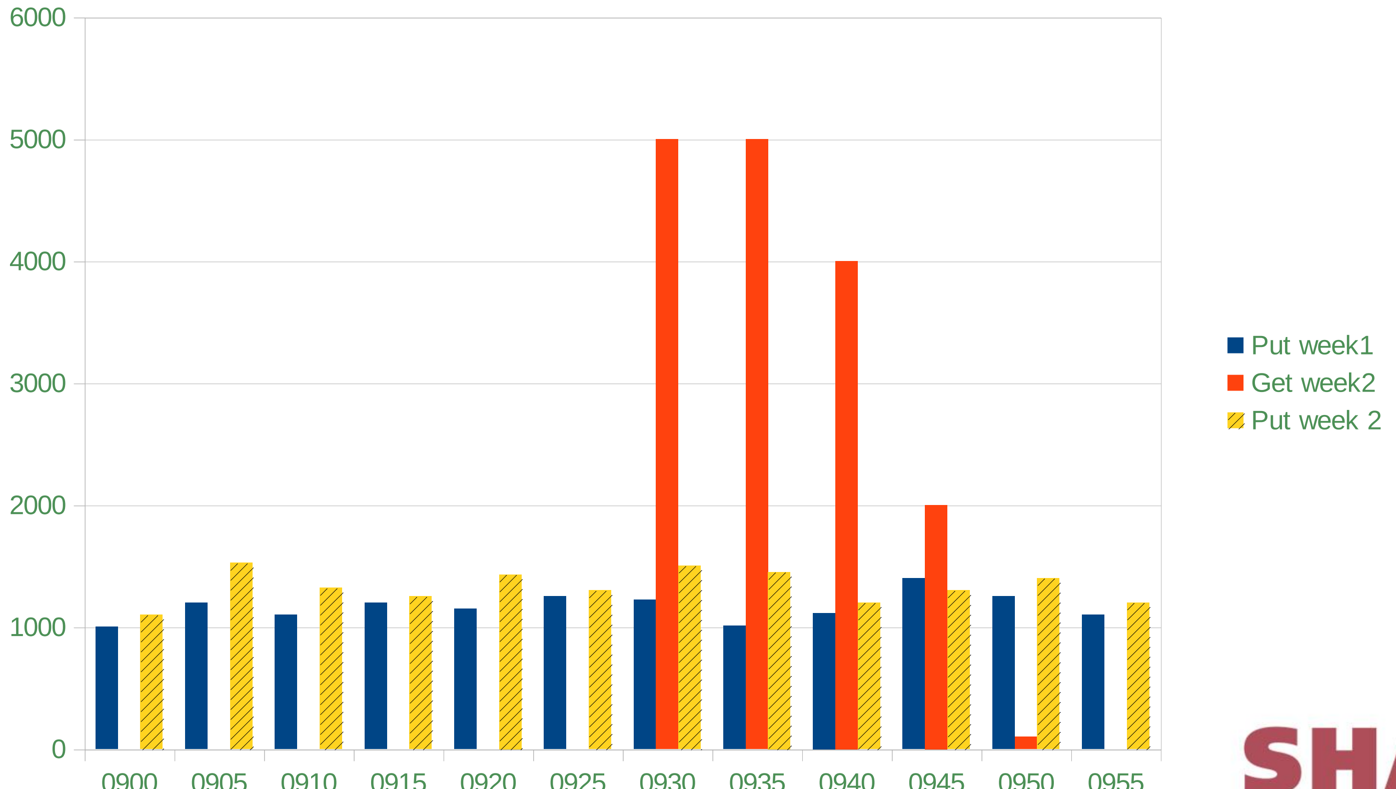
Monitor Puts and Gets

- You should take the peak hour in the week, such as Monday morning 10:00 to 11:00, and plot the data on a week by week basis.
- This data was taken from a test system.
  - It shows the start-up and shutdown of a long test. In normal use, this graph will have more variation.
- A useful metric is the number of puts and gets.
  - The number of puts is the measure of work going through.
  - The number of gets can vary. For example too many servers will lead to more gets, with some gets not getting messages.
  - You would expect a 2 * get for a get from the reply queue.
  - Under the covers the first get returns with no message available and so waits. When the reply arrives, the internal get is retried and the message retrieved.

Notes

Long Term Monitoring

- Plot peak hour every week (for a year)
- Week 2 has higher Put requests than week 1
  - Get jobs are not running at start



Long Term Monitoring

This is based on a customer's data.

The yellow data is taller than the blue data. This shows week 2 has more traffic than week 1.

It also shows the get jobs were started very late. This resulted in a build-up of messages on the queue. When the get jobs were eventually started, they processed lots of messages. It is better to have the get job started before the put jobs to ensure That messages don't build up on the queue. In fact, it may be better to use Trigger Type FIRST to kick off the get jobs when the first message is placed on the queue.

This customer had almost 1 million messages on the queue.

Notes

How to check CPU usage ? ...

MQ Accounting Trace (SMF 116, Class 3)

```
Open name LYN.LOGQ.Q11                                           Object type:Local Queue
Base name LYN.LOGQ.Q11                                           Base type  :Queue
Queue indexed by NONE
First opened 12-09-2010 17:30:34.17
Last closed  12-09-2010 17:36:19.60
Page set ID              63,  Buffer pool         3
Current opens             0,  Total requests      4157
Generated messages :            0
Persistent messages: GETs             0,  PUTs        4155,  PUT1s          0
Put to waiting getter: PUT             0,  PUT1           0
PUTs: Valid        4155,  Max size      35712,  Min size      17856,  Total bytes    139 MB
-MQ call-               N         ET          CT         Susp        LOGW     PSET Epages   skip expir
 Open    :              1         40          39            0
 Close   :              1          5           5            0
 Put     :           4155      10948         193         9457         915
-Logging: Total-count  Total-elapsed   Force-count Force-elapsed
 MQPUT            326       3.801852          318      2.852607
Maximum depth encountered           385
```

- Gives you CPU data
- Similar data can be obtained for get requests.

How to check CPU usage ? ...

MQ Accounting Trace (SMF 116, Class 3)

Detailed information about Queue's:

- Pageset and bufferpool
- Number of valid put requests
- Record size range, you can calculate the average size
- Total elapsed time and cpu time for the requests
- N = number or calls, ET = Elapsed time, CT = CPU time
- Maximum depth

Notes

How to check CPU usage ? ...

IBM Transaction Analysis Workbench

SMF Type 116, Class 3

```
                  SMF Type=116-1 WebSphere MQ Accounting Class 3 Summary (By TRAN,QUEUE)

MQACCT4  Printed at 10:50:30  2/03/2011  Data from 09:00:40 03/03/2010 to 09:59:52 03/03/2010


SSID: SYSB  Type: CICS     Name: CICSSYSP  Tran: TRTI      Threads:         2
  Other     Avg Count               6.0   Avg Elapsed    0.000116  Avg CPU      0.000112

  In-MQ Time (Total)      Elapsed:  0.000233    CPU:    0.000224
  In-MQ Time (Average)    Elapsed:  0.000116    CPU:    0.000112

SSID: SYSB  Type: CICS     Name: CICSSYSP  Tran: TRTL      Threads:         4

  In-MQ Time (Total)      Elapsed:         0    CPU:           0
  In-MQ Time (Average)    Elapsed:         0    CPU:           0

  Queue: APPLICATION_A_REQUEST
  QType: LOCAL    IType: NONE       GDisp: Q_MGR    QCount:         4

               Count      Elapsed       CPU       Susp Elp  JnlWrt Elp   PS Req's  PS Rd Elp   Expired    Page Skip  Msgs Skip
             _________  _________  _________  _________  __________  _________  _________  _________  _________  _________
    OPEN        15.0   0.000019   0.000009
    CLOSE       15.0   0.000002   0.000002
    INQ         15.0   0.000009   0.000008

    In-MQ Time (Total)      Elapsed:  0.001861    CPU:    0.001222
    In-MQ Time (Average)    Elapsed:  0.000465    CPU:    0.000305

  Queue: APPLICATION_B_REQUEST
:
```

CPU time in MQ

CPU time in MQ API

IBM Transaction Analysis Workbench

What does Transaction Analysis Workbench V1.3.0 do?

IBM Transaction Analysis Workbench for z/OS is a tool for analyzing problems with the performance or behavior of z/OS-based transactions. Transaction Analysis Workbench provides a platform for investigating logs and other historical data collected during transaction processing and system operations.

Transaction Analysis Workbench enables you to interactively explore formatted, interpreted, and easily customizable views of log records from various types of log file. You can use Transaction Analysis Workbench to identify and analyze transaction problems quickly, without requiring an expert understanding of log data structures and the relationships between log records.

Notes

IBM Transaction Analysis Workbench

Transaction Analysis Workbench can help you analyze the following types of log record, listed here by system or subsystem.

CICS®
- CICS monitoring facility (CMF) performance class (SMF type 110, subtype 1, class 3)
- CICS trace entries that have been written to a z/OS® generalized trace facility (GTF) data set or the CICS auxiliary trace (DFHAUXT)

DB2®
- DB2 log
- DB2 trace records that have been written either to a GTF data set or to SMF (as SMF type 100, 101, or 102)

IMS™
- IMS log and trace
- IMS transaction index
- IMS monitor and DB monitor
- Common Queue Server (CQS) log stream
- IMS Connect event data, collected by IMS Connect Extensions
- IMS Connect transaction index: created by IMS Performance Analyzer from IMS Connect event data in an IMS Connect Extensions journal
- Tivoli® OMEGAMON® XE for IMS Application Trace Facility (ATF) journal
- Internal resource lock manager (IRLM) long lock detection (SMF type 79, subtype 15)

WebSphere® MQ
- WebSphere MQ log extract
- WebSphere MQ statistics (SMF type 115, subtypes 1 and 2)
- WebSphere MQ accounting (SMF type 116)

WebSphere Application Server for z/OS
- WebSphere Application Server for z/OS request activity performance statistics (SMF type 120, subtype 9)

z/OS
- Selected SMF record types applicable to problem analysis (including RMF™, APPC, job-level accounting, and VSAM activity), in either log streams or dumped SMF data sets OPERLOG, the z/OS operations log (log stream)

Notes

How to check CPU usage ? …

- **Strobe** (from Compuware)
  - **Autostrobe**
    - Automatic analysis of a mainframe job if it exceeds a threshold
  - **iStrobe** and **Strobe**
    - Analyse program performance at point when threshold exceeded

- **IBM Application Performance Analyser**

- **Both tools:**
  - Sample the address space
  - Report which areas of code are hot
  - Report DB2 and MQ usage

- **Also IBM Tivoli Monitoring Tools**

Strobe for IBM MQ (from Compuware)

Strobe for WebSphere® MQ is a component of the Strobe Application Performance Management product line. It helps IT professionals to measure and improve the efficiency and responsiveness of their IBM z/OS® based applications using IBM Websphere® MQ. Strobe for WebSphere® MQ pinpoints individual MQ calls causing excessive wait time or CPU time; summarizes wait time and CPU time for MQ message queues by call type; supplies information on MQ call options and message attributes; identifies counts of MQ calls by command; and identifies program statements calling resource consumptive MQ system services.

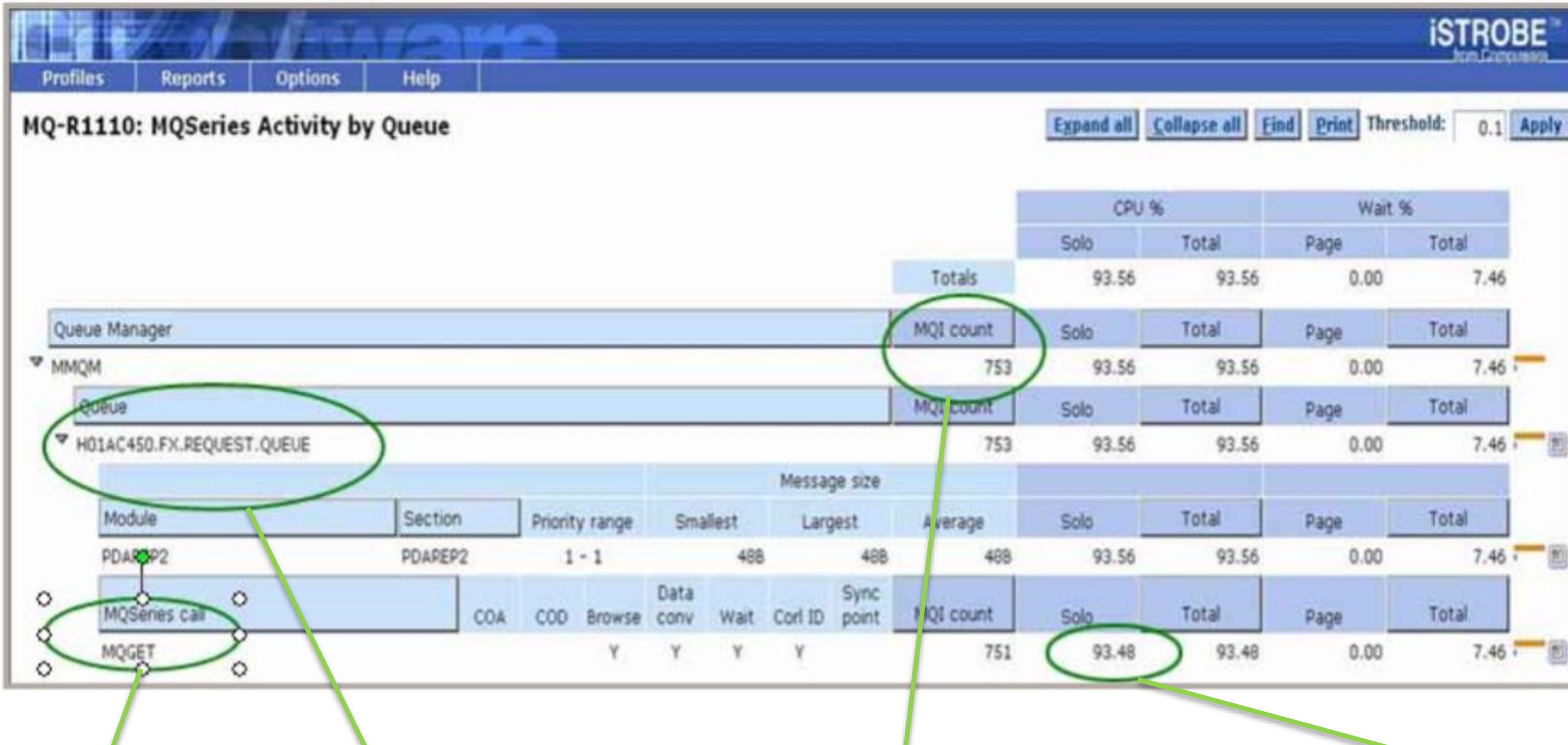
Performance data from Strobe for WebSphere® MQ is fully integrated with information from Strobe and other Strobe options in a single performance profile. Users gain a complete picture of application performance and can quickly locate and eliminate application inefficiencies and improve response times. With Strobe for WebSphere® MQ users can identify and improve resource consumptive MQ calls, evaluate and improve queue efficiency, and avoids excessive use of MQ system services.

Compuware AutoStrobe provides a capability to automatically analyze a mainframe job when it exceeds its usual performance thresholds. Compuware iStrobe and Strobe for WebSphere MQ then allow the program performance to be analyzed at exactly the point of the excessive elapsed time issue.

Notes

Strobe for IBM MQ (from Compuware)

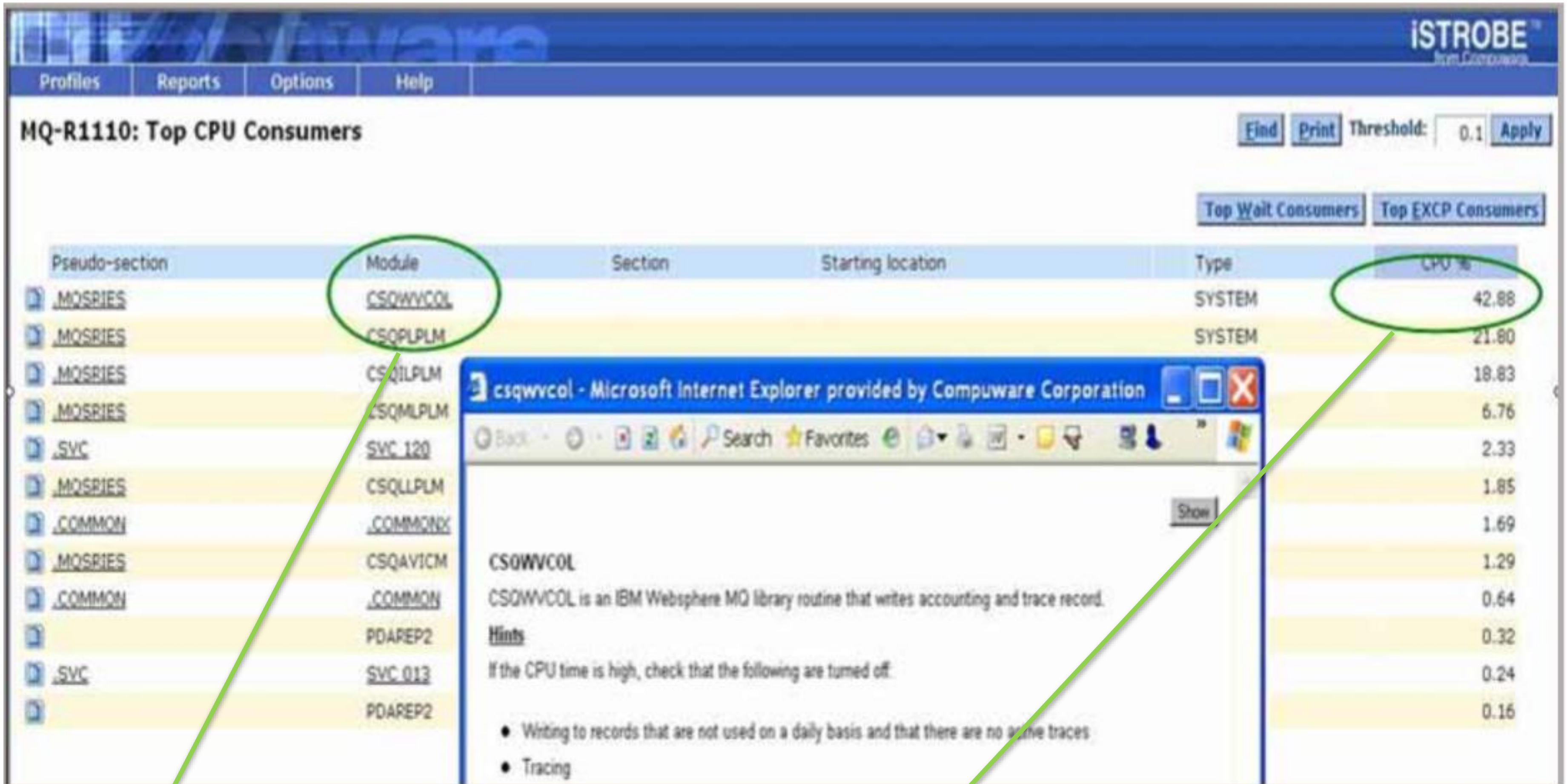
- Activity by Queue shows high CPU usage by MQGET calls



MQGET   Queue Name   MQI Count = 753   CPU = 93.48%

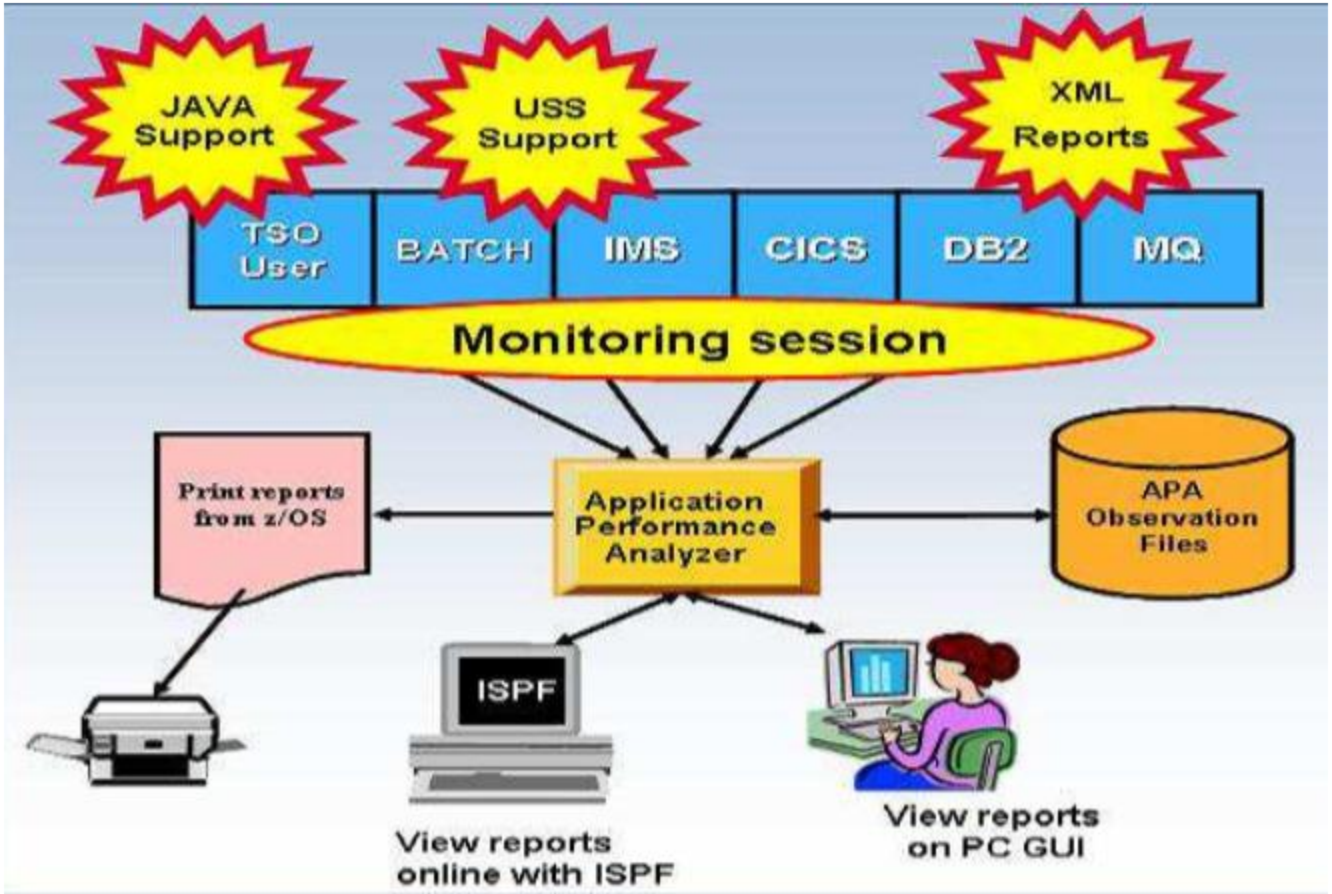
Strobe for IBM MQ (from Compuware)

- High CPU due to MQ Accounting Trace being active



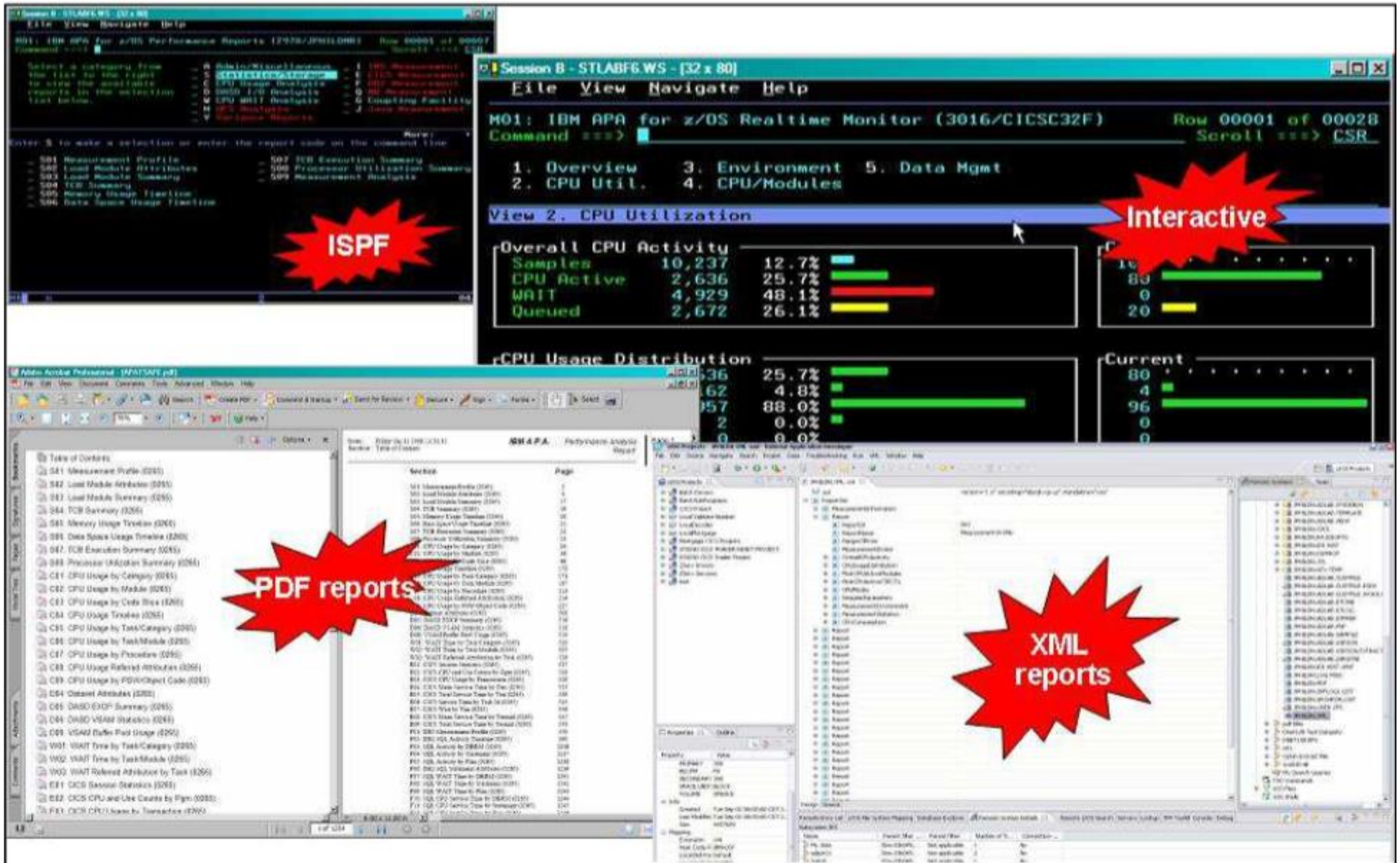
CSQWVCOL

IBM Application Performance Analyser (APA)



IBM Application Performance Analyser (APA) ..

- **CPU usage, service time and wait time by MQ Queue, Request or Transaction**



IBM Application Performance Analyser V13.1

CICS

Support for IBM CICS Transaction Server, including V5, enables you to monitor specific CICS transactions using wildcard transaction prefixes or by termed selection. With this support, you can trace transactions during critical situations, rather than waiting to review data collected Periodically.

IMS

Support for IMS applications, including V13, means you can have IMS application-performance data-on-call time and service-call time for DL/I. You can also choose to trace all IMS calls. Over 20 IMS reports are now provided, helping to improve your ability to analyze the influences of IMS on application performance.

DB2

Support for DB2, including V11, delivers relevant information for performance analysis and tuning, including SQL statements and processor usage by SQL statement as well as for IBM DB2 stored procedures written in Java™. You can also choose to trace all DB2 calls.

WebSphere MQ

WebSphere MQ support provides information about CPU usage by queue, by request and by transaction in WebSphere MQ. Application Performance Analyzer also provides service time by queue, request and transaction, as well as wait time, for the same categories.

(See: ftp://public.dhe.ibm.com/software/htp/pdtools/info/DatasheetAPA.pd)

Notes

Strobe and IBM APA

- Strobe/ Application Performance Analyser:
  - Are near real time
  - Sample the batch/CICS address space, for example 10 or 100 times a second.

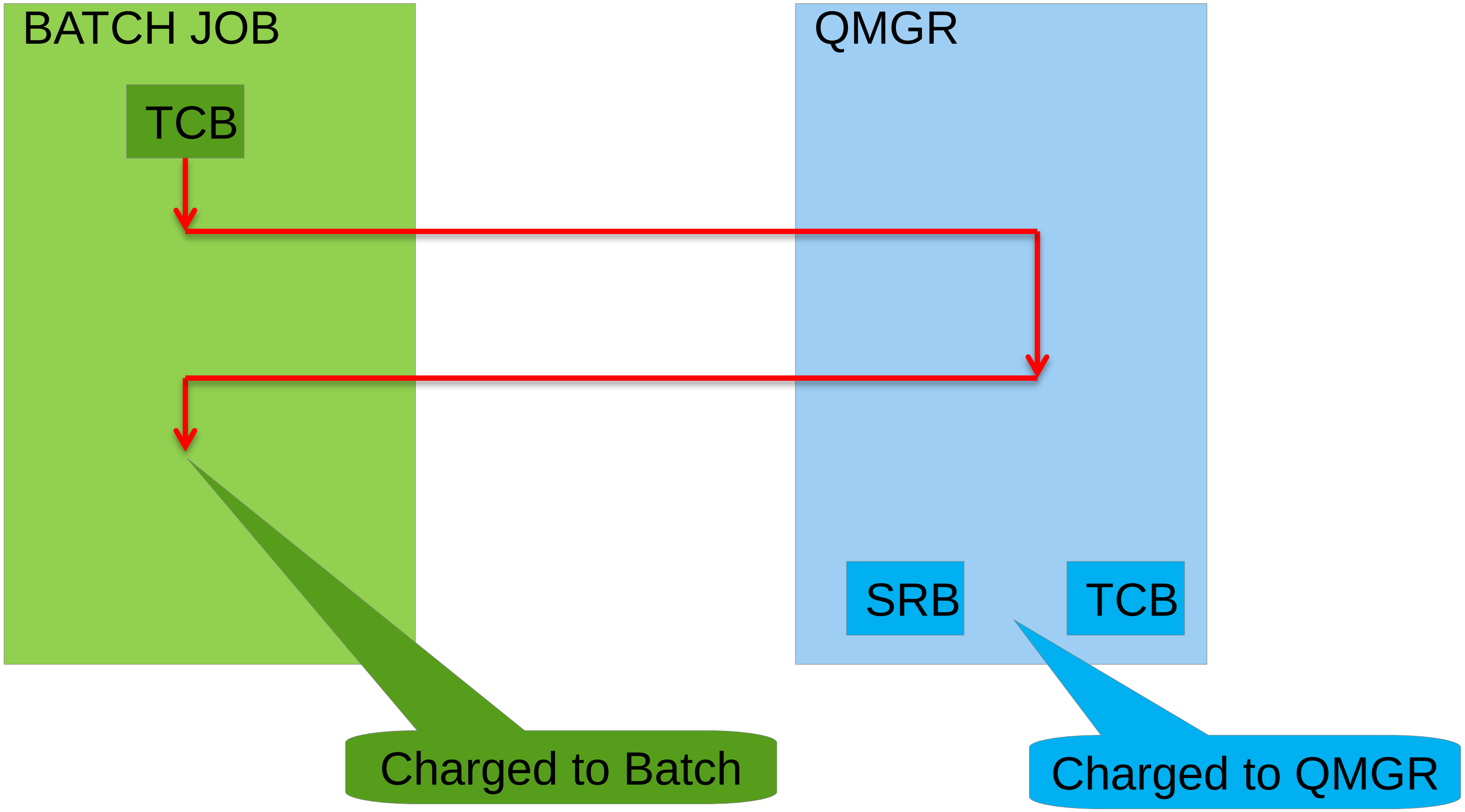
- Strobe tells you hot CSECTs and sometimes the hot instructions

- Application Performance Analyser has little value for the QMGR (or subsystems) as it works best for applications (jobs)

- The tools are very useful for a quick check.

Notes

Where is the CPU being used ?



Where is the CPU being used ?

- Most MQ work is done under the user's TCB (i.e. in a Batch or CICS address space) and is hence charged to the Batch or CICS job. Some work, such as Commit processing, is done on a SRB or TCB within the QMgr and hence gets charged to the QMgr address space.

- If there is a shortage of CPU then the batch job may/can be delayed.

Notes

High CPU due to application design issues

- Use of **Trigger EVERY** with medium to high throughput
  - Additional messages produced
    - Cost of starting and ending transaction
  - A long running transaction is better in most cases
  - May need logic to monitor and start more instances

- Use of **dynamic queues** for reply to queues
  - It is expensive to create and destroy dynamic queues
  - Often better to have a common, predefined reply to queue

- **Polling queues** for messages
  - Some users wrongly code a wait interval of 0.1 seconds and loop around
  - Use MQGET with wait, with a long wait interval
  - On z/OS can use MQGET with signal
  - Now also have MQCB (callback function)

- **Not clearing** the MQMD **MsgId** or **CorrelId** field
  - Results in a search of the queue
  - Gets worse with increased queue depth

High CPU due to application design issues

- Trigger EVERY is good for <= 1 message per second. If you have > 1 message arriving per second, CPU usage with trigger EVERY will be high.

- A Long running transaction is better. User Trigger FIRST to get the instance of the serving application started.

- Could check if the curdepth of the request queue is greater than a certain value and start another instance of the serving application if required. The application can be terminated when the curdepth drops to a certain value.

- For CICS transactions, you can use CPSM to route work to a CICS region with available capacity.

- Consider a reply to queue with same name as a CICS region (so only one LPAR uses it). Or have a queue per application like PAYROLL_REPLY.

- Broker logic with many flows (MQPUT, commit, MQGET)
  - Need to convert data tree into XML, MQPUT, MQGET, read XML into data tree.
  - Very expensive as each hop does commit.

Notes

What can cause high CPU usage in MQ ?

- Non-indexed Queues
  - Many messages scanned but skipped
  - **CSQI004I** MQ7A CSQIMGE3 Consider indexing CP0000 by MSGID for BATCH connection PAICEP7A, 36345 messages skipped
- Too many server applications and not enough messages
  - Results in failed MQGET calls and hence CPU usage
- Still using DB2 blobs for large messages to Shared Queues ?
  - This does not perform well
  - Consider moving to Shared Message Datasets (SMDS)
    - Almost 10 fold improvement in performance

What can cause high CPU usage ?

- Too many servers.
  - If you have a 100 servers waiting for work, and one message comes in, one server will get the message. The other 99 will not. So, there will be 99 no message available errors.
- It is a balance between too many, and not enough, servers.
- Can use automation based on queue depth to start more servers, or have some application logic that queries the queue depth and starts more server instances when the queue depth is high.

Notes

Causes for high CPU usage in the CHINIT

- Clients doing MQCONN, MQOPEN, MQGET/MQPUT, MQCLOSE, MQDISC
  - MQCONN and MQDISC calls are expensive
  - Consider keeping clients connected if possible
  - Can you use connection pooling ?
- Data conversion ?
  - Try to convert on the target queue manager (e.g. use MQGET with convert option)
- Small channel BATCHSZ
  - Consider using a BATCHSZ > 50
  - BATCHLIM greater than the default
- Some TLS ciphers are more expensive
  - See MP16 for full details
- Channel start/stop messages
  - V8 introduced message suppression
- Client should use listener with INDISP QMGR instead of GROUP
  - Eliminate DB2 requests
- What are Channel exits doing ?
  - Use DIS CHS EXITTIME

Causes for high CPU usage in the CHINIT

- A customer had coded an MQ interface layer which resulted in an MQCONN and MQDISC call for every request. This was very expensive ! Try to keep client connected if possible,

- Some customers believe that setting BATCHSZ to 1 reduces CPU and improves throughput. This is actually incorrect. It is better to have a larger BATCHSZ as this results in less confirm flows. Some customers run with a BATCHSZ of 1000 and a BATCHLIM of 2MB.

- Some TLS cipher specs are more expensive. Refer tp the MP16 performance report for more details.

- It is cheaper not to produce the channel start/stop message. MQ V8 introduces the ability to suppress some messages.

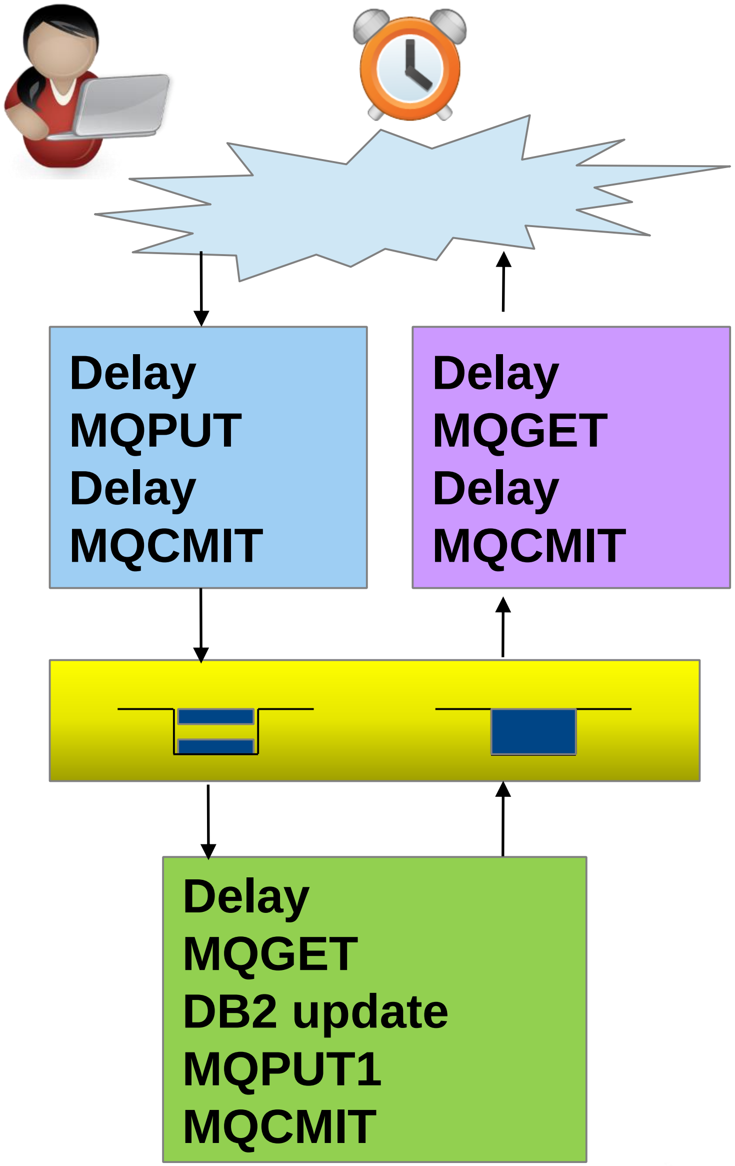
  MQ 8 introduces the EXCLMSG parameter which can be used to suppress a subset of console messages before the queue manager issues them. For example to suppress the messages issued when a SVRCONN channel starts (CSQX511I) and stops (CSQX512I), issue the following command:

  SET SYSTEM EXCLMSG(X511,X512)

- **Note:** For client connections, in order to use the extended transactional client, the server needs to be V701 or higher.

Notes

Where is time being spent ?



- Delay before MQPUT
  - Long MQPUT request
- Delay after MQPUT
  - Long MQCMIT
- Delay before MQGET
  - Long MQGET
  - Long Database update
  - Long MQPUT1
  - Long MQCMIT
- Delay before MQGET
  - Long MQGET
- Delay before MQCMIT
  - Long MQCMIT

Where is time being spent ?

- You need to consider the end to end time. MQ may be just a small part of it.
- We have seen instances where there is an MQPUT, MQGET with wait for 20 seconds, followed by an MQCMIT. Because no message arrived, the MQCMIT occurred 20 seconds after the MQPUT. There were messages in the input queue which could not be got because they were MQPUT in syncpoint. When the message was got, the time the message was put was extracted from the header, and showed 'a long time in MQ'. This was an Application coding problem.
- Long MQPUTs or MQCMITs typically add milliseconds to a request. For example increasing the time from 1 ms to 5 ms. If there are big times like 10 seconds, this is likely to be an application problem.
- If you see response times increase by 10 ms, this is a small problem. If you see response times increase by more than 1 second, this is a big problem.
- Delays in a serving application could be caused by long DB2 requests, or it could be that CICS/IMS is not dispatching the serving application.
- Also, remember that MQ channels are applications and then you also have the network time !

Notes

Where is time being spent ?

- Most people measure from the top.

  For example, web traffic may go through data power boxes, MQ on distributed, then into MQ on z/OS.

  If there are messages queued up on the server, is this an MQ issue ?

  No, it is most likely an application issue, it is not fast enough to remove messages for some reason.

- A customer “just” added an MQPUT call into a CICS transaction and this caused the response time to increase by 10 ms !

- This was because the transaction now had 2 phase commit and each part of the commit processing took 5 ms.

Notes

Consider your MQ Application environments

- **Online processing**
  - Messages processed within seconds
  - End to end response time under 1 second
  - Typically small/medium sized messages
  - Queues typically have 10's of messages
  - Performance critical (needs to be tuned well)
  - Work can be queued to balance workload (10s of msgs/sec)
- **Batch processing**
  - Messages last for hours
  - Typically medium/large messages
  - Deep queues (millions of messages)
  - Performance is not critical
- **Most customers have both environments**
  - Ensure that you isolate Online and Batch workload !
    - For example, use separate channels
    - Don’t want batch to hold up online workload !

Workload isolation

- Check 'batch' workload is isolated from 'online' workload

- Use different:
  - Pagesets
  - CF Structures
  - Buffer Pools
  - Channels

- Use a dedicated QMGR if dealing with large messages

Workload isolation

- Isolate your online and batch work.
- Put short and long lived messages in different page sets and use different buffer pools.
- Maintain a list of the MQ resources used by applications
  - If a resource is impacted, you can tell which applications are affected

Notes

Who are the people ?

Business/Strategist

Development

Infrastructure

Notes

Who are the people ?

- 10 years ago, the infrastructure team was in charge
  - Would not let an application into production unless capacity planning had been done
  - Would tune the infrastructure for a very dynamic workload
- Nowadays, things are more agile
  - Want application written this week and in production next week, with no disruption to existing environment
    - May be a case for just adding additional capacity (CPU) to meet the demand
  - No idea of message size, or throughput
    - Hard to plan for this
  - Application developers are in charge
  - New applications written in Java for mobile phone!
  - Most Cobol programmers have left
    - Hence, some application bugs are hard to fix

Notes

What can delay the (server) application ?

- Are messages building up on queues ?
  - Is this just a peak workload ?
  - Is there a problem with the serving application ?

- Are there enough application instances ?
  - Are CICS/IMS limiting the number of transaction instances ?

- Is the problem in CICS/IMS/WAS ?
  - Are transactions slow to run ?
  - Check statistics

- Is the application busy ?
  - Doing database updates ?

What can delay the (server) application

If the messages are queued up on a queue – it is usually the application problem.

A common cause of 'problems' is the applications are not running properly. This could be due to lack of CPU, or the system is paging.
With batch jobs RMF can report if the job is delayed for CPU.
With CICS transactions, CICS may not be delayed for CPU – but CICS may not be dispatching the transactions because of CICS giving higher priority to other transactions.

Last week the transaction could process work fast enough. This week it cannot keep up. This could be caused by a DB2 change – the DB2 updates are taking longer. It doesnt matter what the problem is – you know where to look next.!

Trigger could be slow because CICS issued the EXEC CICS START TRANSACTION – but CICS is delaying that transaction.

Notes

What can slow down commit processing ?

- Longer I/O requests (1ms → to 5 ms) due to:
  - Increased load
  - Slower DASD
  - Mirrored DASD in use

Use RMF or MQ statistics to verify

Other things to check

- If MQPUT/MQGET calls are taking longer
  - Use MQ accounting class(3) to determine why
  - Check Buffer Pool and CF usage – doing pageset I/O ?
    - Need more buffers → 64 bit buffers
  - Do messages need to be persistent ?

- If commit processing is taking long
  - Use MQ accounting class(3) to determine why
  - Use RMF to look at log response time.
  - Could be due to DASD or CF problems

- If you suspect network problems
  - Check for additional traffic on the network
  - Check your network routing properties
  - Check TCP/IP Send/Receive buffer size settings

Some things to check

- Remember small problems are when the response time increases by under 10 ms. Big problems are when the response time varies by 100s of ms.

- If the applications are short of CPU, do not even look at MQ performance – get the CPU shortage issue fixed first.

- MQPUT and MQGET requests can be delayed – for example if buffer pools fill up – this will typically be a small problem .. additional buffer pools can be defined.

- If commits take longer, this could be due to DASD or CF problems.

- Network problems are usually big problems. Check for additional network traffic. Response time may be fine during the week but may drop at weekends due to cross site backups swamping the network.

  One customer’s TCP routing table had a problem. Traffic was being routed via their remote site to get to a machine that was physically located just 2 meters away !

  Also check the size of TCP/IP send/receive buffers.

Notes

What can delay MQ channel traffic ?

- A slow network can have a major impact
  - Use TCPIP Ping and netstat
  - Get a TCPIP packet trace
  - Use MQ Display commands
- Message Retry
  - Queue full at the remote end
  - Retry after a period
  - Messages queued on transmission queue
- Interference
  - Same channel used for different applications
  - Application 1 has more traffic
    - Impacts Application 2
  - Application 1 gets queue full
    - Application 2's messages are delayed.

What's happening to my Channel ?

- Measure your round trip time using PING

```
//RUN1GB  EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
 ping winmvsac (count 1000 verbose length 32768
/*
```

Ping statistics for winlnxn7.hursley.ibm.com (9.20.4.136)

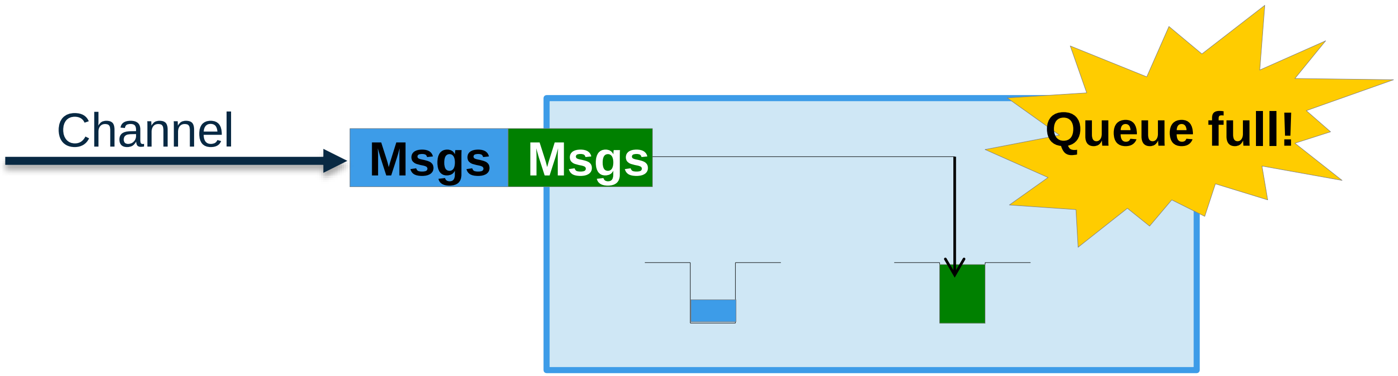
Packets: Sent=1000, Received=1000, Lost=0 (0% loss)

Approximate round trip times in milliseconds:

Minimum=1.03 ms, Maximum=10.19 ms, Average=1.13 ms, StdDev=0.50 ms

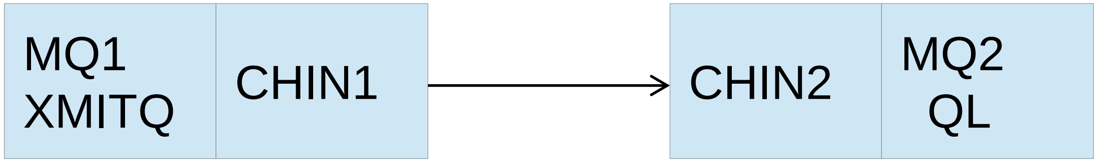
Seen customers with over 80 milliseconds across pacific.

Message retry



- The green queue is full
  - Messages destined for the green queue wait for a period
    - MRTMR(1000) – time between retries
    - MRRTY(10) – number of retries
- The blue queue has capacity
  - Messages destined for the blue queue are delayed !

Check the Channel Status



DIS CHS(...) XQTIME MSGS XBATCHSZ

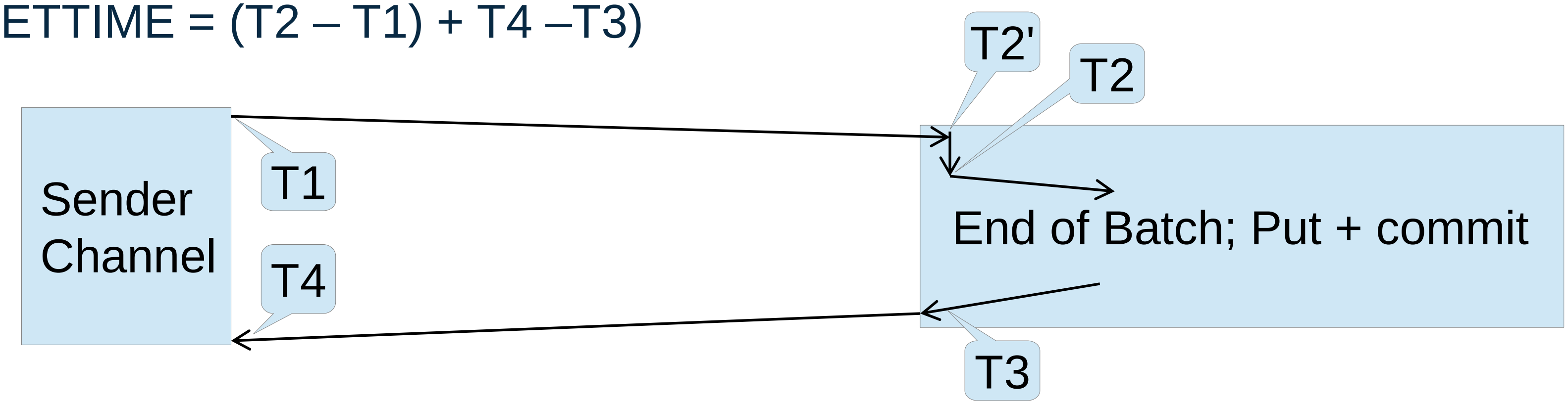
- MSGS(2000)
  - Number of messages sent/received since channel started
- XQTIME(65,53) uSeconds
  - How long was the message on the XMITQ for ?
  - < 100 uSeconds is good
- XBATCHSZ(1,1)
  - Low means channel was waiting for messages to send

What is NETTIME ?

DIS CHS(...) NETTIME

NETTIME is the time (in microseconds) to send an end of batch request and receive a response, minus the time to process the end of batch request.

i.e. NETTIME = (T2 – T1) + T4 –T3)



- Time T2' is when the TCP/IP buffer holds the data
- T2 – T2' is the time between data arriving and the channel being ready to receive it
  - A busy/blocked channel can result in long delays
  - So, the network may well be ok even if the NETTIME is high !
- Generally, a short NETTIME means the network is OK

CHINIT SMF Statistics and Accounting Data

- CHINIT SMF data capture introduced in V8
  - Channel Initiator Statistics (SMF 115, SubType 231)
  - Channel Accounting Data (SMF 116, SubType 10)
- Useful for:
  - Monitoring CHINIT and Channel activity
  - Capacity planning
  - Tuning
- Separate controls from Queue Manager SMF, allows 'opt in'
- Supportpac MP1B updated to format new data

CHINIT SMF Statistics and Accounting Data

- New SMF data for CHINIT address space:
  - **Channel Initiator Statistics** (SMF 115, SubType 231)
    - High level view of activity in CHINIT
      - Number of channels and TCB usage
      - Dispatchers, Adapters, DNS, SSL
        - Do I have spare capacity ?
        - Do I need more or less dispatchers/adapters ?
  - **Channel Accounting Data** (SMF 116, SubType 10)
    - Detailed view of individual channels
      - What work have channels been doing ?
      - Which channels are being heavily utilised ?
    - Controlled by STATCHL attribute on QMgr and Channel definition

What's happening to my queue?

- DIS Q(xx) curdepth

  Tells you the depth of a queue now

- RESET QSTATS(QUEUE1)

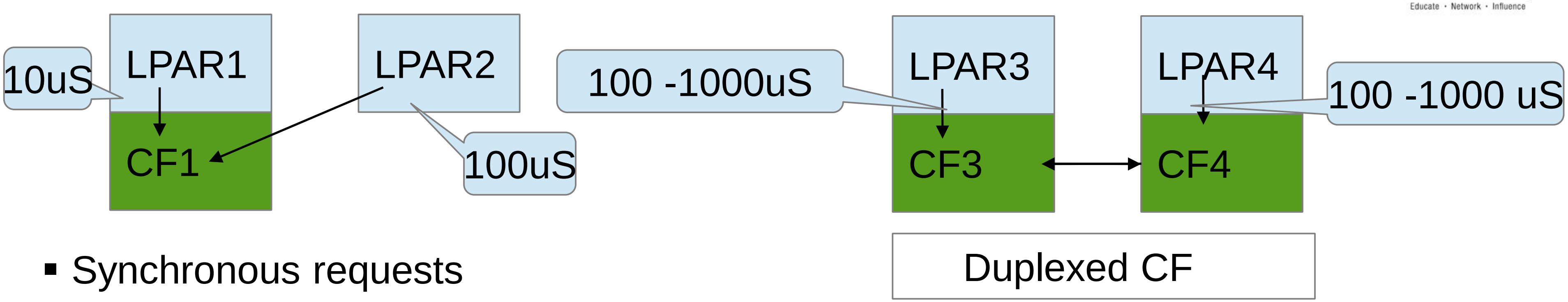
  RESETINT(14)
  HIQDEPTH(6)
  MSGSIN(3482)
  MSGSOUT(3482)

  Processing rate = 3482/14 = 249 messages per second

- Performance events

- Use monitoring product or MQCMD in SupportPac MP1B
  - Issues command periodically and stores output in CSV file

What can make MQPUTs/MQGETs to Shared Queues take longer ?



- Synchronous requests
  - One long instruction - 10 uSeconds
- Asynchronous
  - Issue request, suspend, wait, resume - 800 uSecs
- Use RMF
  - Ratio of Sync and Async requests
  - Response times of Sync and ASync
- Ratio of Sync:Async can change with other work on CF
  - Monitor %CF busy, I/O response times and delays

Coupling Facility Synchronous v'S Asynchronous

System decides which to use. If system wants to use less CPU, it will go Async, But at the cost of throughput.

Synchronous

If the CF is close (within same physical processor) it may take 5 microseonds to get to it.

If it is in a different box it will take longer. So in this case it takes 20 uSeconds.

Asynchronous

If we change the programming model – so we start the I/O then sleep, and get woken up when the I/O has completed, it only uses 10 uSeconds of CPU.

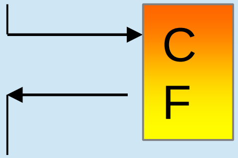
Which is cheaper in terms of CPU usage ?

Clearly 10 uSeconds, so z/OS will go with the one using less CPU.

Notes

Coupling Facility Synchronous v'S Asynchronous

Synchronous
CF Read

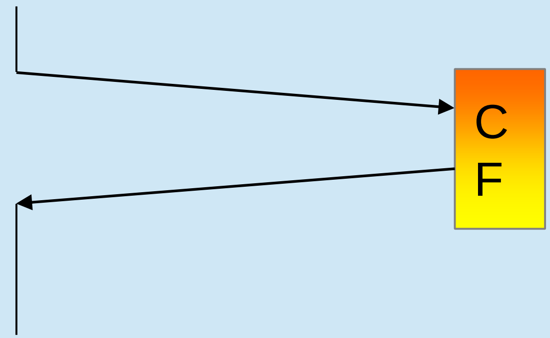


Elapsed time 5 uSeconds
CPU time 5 uSeconds

Synchronous
CF Read

Elapsed time 20 uSeconds
CPU time 20 uSeconds

ASynchronous
Start CF Read
Wait



Post
Elapsed time 100 uSeconds
CPU time 10 uSeconds
Reduced throughput

Which uses less CPU for long distance ?

What's happening to my Buffer pool

- **DIS USAGE command now gives you the buffer pool attributes**

  CSQI065I MQPC Buffer pool attributes ...

Buffer pool	Available buffers	Stealable buffers	Stealable percentage	Page class	Location
0	10000	9981	99	4KB	BELOW
1	10000	9979	99	4KB	BELOW
2	10000	9999	99	4KB	BELOW
3	10000	9994	99	4KB	BELOW
4	50000	49962	99	4KB	BELOW

– 99% free !

What is my logging rate ?

- Log message
  - **CSQJ002I** MQPD END OF ACTIVE LOG DATA SET DSNAME=...
    STARTRBA=0000000E72A1E000 ENDRBA=0000000F32B9DFFF
    0F32B9DFFF−0E72A1E000 = 3074MB
  - Every 2 minutes
  - Logging rate = 25 MB/Second
- Monitor how often your logs fill up!
  - Edit joblog

    x all;f CSQJ002I all; del all x;

    look at timestamps

What do I need to do now ?

- Need to be able to compare a good day with a bad day
- Capture data about the system
  - Typical queue depths at peak time
  - How long messages were on xmit queue
  - Buffer pool usage
  - z/OS view of MQ
  - CF usage
- Which LPARs were using the structures and response times
- Practice collecting the data
  - Everything will be in place and tested in case of a problem

What do I need to do now ?

When IBM is called in to look at a problem it is key to have data from a good day and the bad day.
If we go on site and find 10 problems – such as slow logs, buffer pools filling up, it is hard to tell which is the problem causing today's problem. If we can say the buffers pools were bad yesterday and were bad today – then this is not likely to be the problem.
If today we see deep queues- and usually they only have 10 messages on them this this is good evidence.
If they queues always have lots of messages – then the problem is else where Queue managers on different LPARs can have different response times from the CF – depending how close it is. Is the work no running on a different LPAR?
Has the problem just started or has it slowly developed over time? At the beginning of the session spoke about application response time went from 200 ms to 1000 ms. Customer could say this happened at this time on Thursday.
This was very useful information – it wasn't a slow change, and so is likely to be network or z/OS. It turned out to be a some of the data power boxes were reconfigured so all of the traffic went through one box

One customer had problem where queue built up in the morning for about 1 minute and then back to normal. They had to prepare the commands, as 1 minute was not long enough to type them in.

Notes

Your mileage may vary

- Different implementations have different characteristics
  - Same application in concept
  - Different implementation (DB2, message size, etc.)
- Resources available
  - Storage for buffer pools
  - Some customers have spare capacity, some have none
- Response time from DASD and CF
  - Throughput rates
- General tuning common to all customers
  - Customer specific tuning
- Some customers test at max volume + 25%, some 'Implement and pray'

Should I tune it ?

- Tuning is not always beneficial

  - For example, you could merge two channels to save CPU but..
    - May cause longer response times
    - Large messages may delay delivery of short messages
    - Channel may go down … hence no availability

- You may accept higher costs for improved availability
  - Multiple LPARs
  - QMgrs in a Queue Sharing Groups and Shared Queues

Technology answers to old applications

- Reluctance to change old applications
  - Applications are stable
  - Developers have left
  - A technology solution is needed to handle today's increased workloads
- V8 introduced:
  - 64 bit buffer pools
  - Up to 100 buffer pools (0 to 99)
  - Increased number of pagesets (0 to 99)
- Increase size/number of logs to address increased workloads
  - Stripe Log datasets
- Consider upgrading to newer hardware

Summary of things to check

- Look at queue depths for application queues
  - Large depths → there's a problem elsewhere
- Check Buffer pool < 85% full
  - Otherwise increase buffer pools or move to 64-bit buffer pools
- Check DIS CHS
  - XBATCH < BATCHSZ
  - XQTIME
  - NETTIME
- TCP PING data
- Check response time of the MQ Logs

Any Questions ?

Please complete your session evaluation .. Thank You !

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE in Orlando 2015

Copyright and Trademarks