

#SHAREorg

SHARE
Security - Compliance - Risk

The Dark Arts of MQ SMF Evaluation

Lyn Elkins
elkinsc@us.ibm.com
Session # 13884
August 13, 2013

SHARE
in Boston



#SHAREorg

SHARE
Security - Compliance - Risk

Code!



IE
in Boston

The witch trial – MQ is broken!



WITCHCRAFT AT SALEM VILLAGE.

Complete your sessions evaluation online at [SHARE.org/BostonEval](https://www.share.org/BostonEval)



Agenda



- Review of SMF 115 and SMF 116 class 3 data
- Hunting down the culprit
 - SMF115 Data
 - Bufferpool behaving badly
 - Volume growth
 - Log manager getting cranky
 - SMF116 Data
 - What queues are being used and how?
 - Pulling the data for one CICS transaction or batch job
 - Long running tasks
- Summary

Complete your sessions evaluation online at [SHARE.org/BostonEval](https://www.share.org/BostonEval)



Objectives

- This session is to delve a bit deeper into how the ATS team use the SMF data to find transaction and WMQ problems, based on situations we've tried to resolve.
- It will bore you to death.

Review of SMF115

- The SMF 115 data is the statistical information produced by a WMQ for z/OS queue manager.
 - Primarily used to track major trends and resolve performance problems with the queue manager
 - Very lightweight
 - Broken down into the major 'managers' within WMQ
 - The 'old' MP1B provides several views into the data:
 - MQ1150 – detailed SMF115 report
 - MQCSMF – extracts specific information from SMF115 and 116 in a column format
 - *Particularly useful for building spreadsheets*
 - The 'new' MP1B provides two views of the data
 - Report from for each manager
 - Comma separated values

Review of SMF116 – Class 3 data

- The SMF 116 data is the accounting information produced by a WMQ for z/OS queue manager.
 - Primarily used to determine what is going on within WMQ workload
 - Heavyweight
 - Broken down into the transactions within WMQ
 - The old MP1B provides several views into the data:
 - MQ1160 – prints the SMF116 class 1 report
 - MQ116S – prints the detailed SMF116 class 3 report, including the queue information
 - MQCSMF – extracts specific information from SMF115 and 116 in a column format
 - *Particularly useful for building spreadsheets*
 - The new MP1B provides:
 - The 'TASK' output
 - *Somewhat like the MQ116S report*
 - *I am currently writing a paper on the differences/similarities*
 - Other files, much like the 'old' MQCSMF output

Finding the documentation



SMF 115 data – Hunting down the culprit Red Flags for bufferpools



- In the next few slides, an analysis of a bufferpool under stress is shown
- First the raw SMF data for two weeks was processed thru the “old MP1B” MQCSMF and the MQ1150 format and print programs
- The Buffer Manager statistics were downloaded into a spreadsheet
- The spread sheet was sorted to find:
 - Non-Zero Short on storage counts
 - Non-Zero DMC counts
 - Percent of free pages
- This showed the areas that needed to be looked at in greater detail, and it became apparent that there were some processing patterns that need evaluation

Complete your sessions evaluation online at SHARE.org/BostonEval



SMF 115 data – Hunting down the Culprit



- **Red Flags for Bufferpools**
 - SOS

QMGR	BP	NumBuf	%now	%low	dwt	dmc	sif	sla	sos
QML2		3 70000	18	0	109	198906	922354		1 50
QML2		3 70000	19	0	68	143872	387873		1 13

- Freepages at 5% or less

Date	Time	QMGR	BP	NumBuf	%now	%low	dwt	dmc	sif	sla	sos
201133408:15:21	QML1			3 70000	98	5	9	27	32557	0	0
201133420:41:19	QML1			3 70000	95	5	2	384	81145	0	0

Complete your sessions evaluation online at SHARE.org/BostonEval



SMF 115 data

- **Red Flags for Bufferpools** - Continued
 - DMC – synchronous write process kicks off

QMGR	BP	NumBuf	%now	%low	dat	dmc	sit	sfa	sos
QML3	3	70000	16	0	58	210092	863991	1	0
QML3	3	70000	22	3	132	36528	1232774	2	0

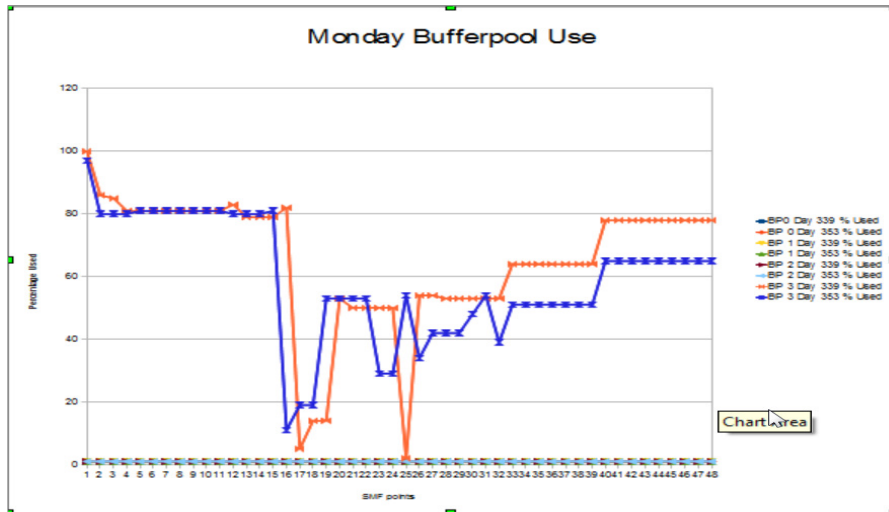
- The DMC count should be used in conjunction with the IMW field from the SMF115 report to see how many synchronous writes were actually performed.

The NEW SMF print – BUFCSV file

- This spreadsheet image was ceated from WMQ V7.0.1 data thru the new MP16 print program
 - Note the data produced is different from the MQCSMF report from the old version. Important fields missing include the important SOS, deferred writes, and synchronous writes fields.

MVS	QM	Date	Time	BP	size	lowest free	# get new pg	# get old pg	# read I/Os	# pg writes	# write I/Os	# sync	w
MPX1	QML1	2010/09/29	15:32:18	0	5000	4980	0	66219	0	0	0	0	0
MPX1	QML1	2010/09/29	15:32:18	1	15000	7233	15302	31695	0	0	0	0	0
MPX1	QML1	2010/09/29	15:32:18	2	40000	5980	39371	32569	116	17400	4350	0	0
MPX1	QML1	2010/09/29	15:32:18	3	20000	3281	18921	29093	0	0	0	0	0
MPX1	QML1	2010/09/29	15:32:18	4	30000	29999	0	68	0	0	0	0	0
MPX1	QML1	2010/09/29	15:32:18	5	30000	29999	0	134	0	0	0	0	0
MPX1	QML1	2010/09/29	15:32:18	9	20000	2583	17521	22273	338	4976	1244	0	0

SMF115 – Bufferpool Trends and Analysis



Complete your sessions evaluation online at SHARE.org/BostonEval

in Boston

SMF115 – Bufferpool Trends and Analysis - Notes



- In the chart shown two high volume days were compared to see if there was a pattern to the BP use.
- BP 0, 1 and 2 showed almost no utilization.
- BP 3 was in very heavy use, some of the time.
- BP 3 is under some stress.
- Having multiple days worth of data is vital, had there just been one heavy day it may have been an anomaly. Data from longer periods of time, when compared like this can be very useful in tracking usage, etc.
- In this case there was a clear pattern of overuse of bufferpool 3, in further evaluation the SMF116 data showed that all the queues that were being used for this queue manager were defined on the same pageset/bufferpool. By moving some of the queues to another resource pool, the stress was reduced, work flowed faster and the CPU usage was reduced.
- In attempting to replicate the issues, the information on the previous slides was used to create the charts, but also to show that charting the pattern might be helpful in the evaluation.

Complete your sessions evaluation online at SHARE.org/BostonEval



SMF 115 data

- **Yellow Flags for Bufferpools**
 - Consistently Approaching/Achieving 20 % Free pages

QMGR	BP	NumBuf	%now	%low	swt	dmc	sif	sifa	sos
QML4	2	70000	53	19	0	0	46571	0	0
QML4	3	70000	98	20	0	0	46028	0	0
QML4	3	70000	75	20	0	0	0	0	0

Complete your sessions evaluation online at SHARE.org/BostonEval

Bufferpool Use - Warnings

- In the data shown, two bufferpools were approaching the 20% freepage threshold.
- At 20% the async write task is initiated, which is not catastrophic, but if it can be avoided it should be.
- In this case, when several weeks worth of data were examined the 20% threshold was being broken on a regular basis. After evaluation for the SMF116 class-3 data it was found this was batch oriented workload, and messages were expected to queue up for long periods of time this was not a problem. It is something to watch.

Complete your sessions evaluation online at SHARE.org/BostonEval

SMF 115 data – Hunting down the Culprit

- Message Manager Information
 - Good indication of queue manager usage
 - This is only a count of API calls, not one of successful calls
 - Volume trends can be approximated from the MQPUT and MQPUT1 calls, as these are generally successful
 - MQGETs may or may not have data returned

QMGR	Open	Close	Get	Put	Put1	inq	inql	Set	Total API calls	Total Puts
QML1	160	151	2,925,084	3,417,313	0	1	0	0	6,342,709	3,417,313
QML1	248	228	2,256,084	3,150,666	0	5	0	0	5,407,231	3,150,666
QML1	897	895	3,468,114	3,093,355	0	50	0	0	6,563,311	3,093,355

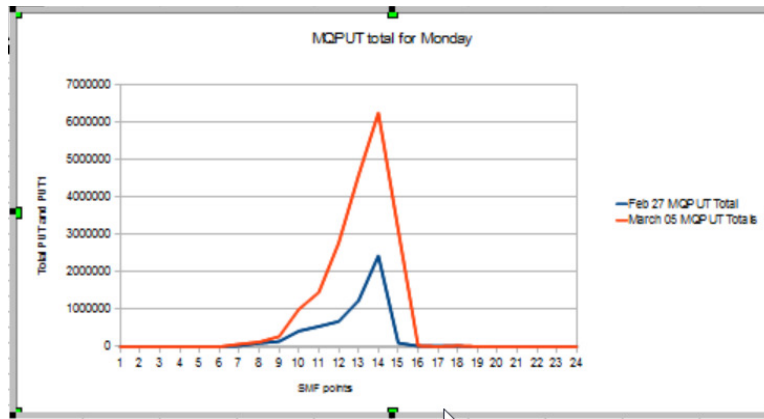
Complete your sessions evaluation online at SHARE.org/BostonEval

Message Manager Statistics

- This data was taken from the message manager output from the old MQCSMF format and print program.
- Two columns were added to calculate the Total API calls and Total Puts (sum of MQPUT and MQPUT1 calls)
- When charted over a few weeks an upward curve was noticed.

Complete your sessions evaluation online at SHARE.org/BostonEval

Message Manager - Trend Chart



Complete your sessions evaluation online at SHARE.org/BostonEval



Message Manager Trend Chart



- Two days data is not really a trend analysis, but it's a start
- If more Mondays are charted, a real trend may emerge and show that volume is increasing allowing a good admin to plan for additional workload.
- This is an overall count for the queue manager, individual queue activity can be evaluated from the SMF116 class 3 data.

Complete your sessions evaluation online at SHARE.org/BostonEval



New MP1B Print program – Message Manager



MVS	QM	Date	Time	Puts	Put1s	Gets
MPX1	QML1	2010/09/29	15:32:18	36070	0	30659
MPX1	QML2	2010/09/29	15:32:19	21725	0	16433
MPX1	QML3	2010/09/29	15:32:38	20289	0	16237

Complete your sessions evaluation online at SHARE.org/BostonEval



SMF 115 data – Hunting down the Culprit



- Log Manager Information
 - Good indication of persistent messaging use
 - As has been mentioned before some of the counts are not complete, the checkpoints does not include those from queue manager switching

QMGR	wr_wait	wr_nwait	Writes	Aug09 Force	Aug09 Log Buffer Waits	read_buf	read_act	read_arc	r_delay	N_CheckP	Aug09 I/O Num	Aug09 Control Intervals Written	paging
QML1	0	569925	339	1	0	0	0	0	0	0	22020	241748	0
QML1	0	621641	337	0	0	0	0	0	0	0	23758	230944	0
QML1	0	753611	363	1	0	0	0	0	0	0	27490	285402	0

Complete your sessions evaluation online at SHARE.org/BostonEval





Message Manager – New MP1B output

- This is from the Message Manager CSV file.
- Note that all it reports are the MQPUT, MQPUT1 and MQGET requests.
- If you are looking for any of the other requests, the TASK report (from the MP116 data) must be used.

Complete your sessions evaluation online at SHARE.org/BostonEval



SMF 115 data – Hunting down the Culprit

- Log manager – I/O rate
 - The I/O rate is calculated as
 - The number of CIs written * 4096 (CI size)
 - Divided by 1 M (1024*1024)
 - Divided by the number of seconds in the interval
 - The I/O rate is the throttle for many queue managers

Aug08 Control Intervals Written	Aug08 Logging Rate (MB per second)	Aug09 Control Intervals Written	Aug09 Logging Rate (MB per second)	Sept30 Control Intervals Written	Sept30 Logging Rate (MB per second)	Nov05 Control Intervals Written	Nov05 Logging Rate (MB per second)
20658	1.34	241748	15.74	58938	3.84	33492	2.18
22446	1.46	230944	15.04	70570	4.59	25822	1.68
22550	1.47	285402	18.58	46630	3.04	27688	1.80
20870	1.36	266212	17.33	79076	5.15	76658	4.99
23458	1.53	307780	20.04	53588	3.49	74088	4.82

Complete your sessions evaluation online at SHARE.org/BostonEval





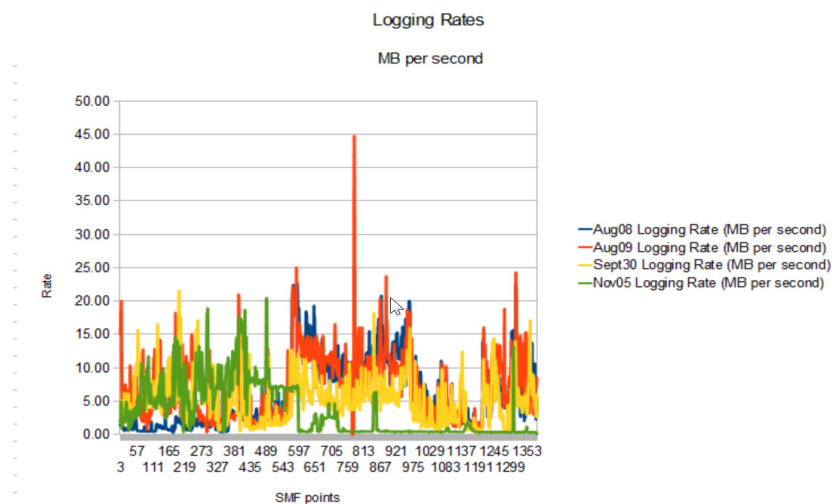
Log Manager Statistics

- This data was taken from the log manager output from the old MQCSMF format and print program.
- The log buffer waits indicates the number of times during the interval there were not free log buffers. This is somewhat tunable, but most production environments have it set to the recommended 40,000. If this count goes very high and the maximum number of buffers are allocated, then the queue manager may be saturated.
- Another critical factor is the I/O rate that can be achieved

Complete your sessions evaluation online at SHARE.org/BostonEval



Logging Rates - Charted



Complete your sessions evaluation online at SHARE.org/BostonEval





Logging Rates Charted - Notes

- These rates were charted from days when there were reported slowdowns.
- Notice the huge spike in the Aug 09 data, that is an anomaly caused by a change to the SMF interval in the middle of the day. It is left in to illustrate that spikes do happen and should be investigated. They may not indicate a sudden growth rate, but can indicate a problem with the data itself.
- In looking at the data, the logging rate is frequently at the 20/25 MB per second rate. For the environment, this was quite high. It was discussed with the capacity planning team. This is continually being monitored, there may be workload that has to shift to another queue manager in the near future.

Complete your sessions evaluation online at SHARE.org/BostonEval



Log Manager CSV file from new MP1B

z/OS	QM	Date	Time	MB Written	MB/SEC	MB Used	Pages per I/O	Checkpoints
MPX1	QML1	2010/09/29	15:32:18	400	0	399	34	0
MPX2	QML2	2010/09/29	15:32:19	340	0	337	20	0
MPX1	QML3	2010/09/29	15:32:38	441	0	438	30	0
MPX2	QML4	2010/09/29	15:34:02	876	0	864	15	0

- The new MQCSMF print program will calculate the MB/Second written
 - A caution, it uses the number of seconds per SMF interval defined for the run. If you allow this to default, your results are likely to be incorrect.

Complete your sessions evaluation online at SHARE.org/BostonEval



SMF116 Class 3 data

- Reviewing this copious data can feel like searching for the spell to turn lead into gold. It's more like panning for gold
- As a WMQ admin, you have more information at your fingertips about your environment than we at IBM reviewing this data will have. There are a number of things that we do to look for patterns or particular problems that are discussed.



Complete your sessions evaluation online at SHARE.org/BostonEval



SMF116 - Hunting the culprit

- The scenario is simple:
 - 'We are missing our SLAs on some of our transactions'
 - The SMF 115 may or may not show bottlenecks
 - You have over 3M SMF116 class 3 records from one SMF interval to see if you can find the problem
 - And, of course, 'MQ is the problem'

Complete your sessions evaluation online at SHARE.org/BostonEval





What queues are being used and how?

- SMF116 class 3 data shows the use of queues
- Helpful because even as a WMQ admin, it may be a challenge to find out where the queues are
- Some specific problems:
 - Non-indexed queues
 - High volume request/reply queues in same resource pool
 - Overuse of Temporary dynamic queues

Complete your sessions evaluation online at SHARE.org/BostonEval



What queues are being used and how?

- We have seen some specific problems/issues at a number of customers that were addressed with an evaluation of the SMF116 data.
- In this section we are going to show some of the more common ones, and how the SMF data lead to the improvement

Complete your sessions evaluation online at SHARE.org/BostonEval



What queues are being used and how?

- Queue Indexing
 - Messages that are retrieved using an index-able field benefit from being indexed even when the depth is not high.
 - Message ID
 - Correlation ID
 - Token
 - Group ID
 - The greater the depth of the queue the greater the benefit.
 - The SMF116 queue records show when messages are retrieved using a 'known' field

Queue Indexing - Notes

- Queue indexing is unique to WMQ on z/OS
- The use of a proper index can substantially improve performance and CPU consumption, as will be shown
- Anecdotally, we've heard of it making a difference when queue depths were as low as 5 on a busy system
- Often the first report of a problem is when there has been a slowdown elsewhere and queue depths have grown unexpectedly

Non-Indexed Queue retrieval



```

Open name TEAMXX.NON.INDEXED          Object type:Local Queue
Base name TEAMXX.NON.INDEXED         Base type :Queue
Queue indexed by NONE
First opened 12-03-2012 15:12:58.55
Last closed  *** ** ** ** **
Page set ID      4, Buffer pool      3
Current opens   1, Total requests   61
Generated messages :
Persistent messages: GETs      0, PUTs      0, PUT1s      0
Put to waiting getter: PUT      0, PUT1      0
GETs: valid      28, Max size      80, Min size      80, Total bytes  2240
GETs: Dest-S     28, Dest-G      0, Brow-S      0, Brow-G      0, Successful destructive  28
Time on queue : Max 4583.730054, Min 257.434901, Avg 3958.326341
-MQ call-      N      ET      CT      Susp      LOGW      PSET Epages      skip expire
Get      :      28      384      369      0      0      0      3505      0
Inquire:      28      22      21
Maximum depth encountered      258
    
```

Complete your sessions evaluation online at SHARE.org/BostonEval



Non-Indexed Queue retrieval - Notes



- In the SMF record, the fields of interest are:
 - The Queue Indexing
 - The Type of GET request being made, those with a '-S' are for specific messages (Get by correlid, get by message id, etc.)
 - The total CPU expenditure for the successful gets – the 'CT' column highlighted
 - The number of pages skipped while finding matching messages

Complete your sessions evaluation online at SHARE.org/BostonEval



Indexed Queue Retrieval

```

Open name TEAMXX.INDEXED                               Object type:Local Queue
Base name TEAMXX.INDEXED                               Base type :Queue
Queue indexed by CORREL_ID
First opened 12-03-2012 15:16:01.44
Last closed 12-03-2012 15:16:50.35
Page set ID      4, Buffer pool      3
Current opens   0, Total requests   59
Generated messages :      0
Persistent messages: GETs      0, PUTs      0, PUT1s      0
Put to waiting getter: PUT      0, PUT1      0
GETs: Valid     27, Max size     80, Min size     80, Total bytes 2160
GETs: Dest-S    27, Dest-G      0, Brow-S      0, Brow-G      0, Successful destructive 27
Time on queue : Max 4780.946117, Min 422.046309, Avg 4288.437716
-MQ call-      N      ET      CT      Susp      LOGW      PSET Epages      skip expire
Get   :      27      105      99      0      0      0      0
Inquire:      26      21      20
Maximum depth encountered      250
  
```

Complete your sessions evaluation online at SHARE.org/BostonEval

Indexed Queue retrieval - Notes

- In the SMF record, the fields of interest are:
 - The Queue Indexing
 - The Type of GET request being made, those with a '-S' are for specific messages (Get by correlid, get by message id, etc.)
 - The total CPU expenditure for the successful gets – the 'CT' column highlighted
 - The number of pages skipped while finding matching messages

Complete your sessions evaluation online at SHARE.org/BostonEval

Indexed vs Non - comparison

- Comparing the CPU time, both queues with the same max message depth:
 - Indexed 27 messages at 99 CPU microseconds
 - 3.667 ms per message retrieved
 - Non-indexed 28 messages at 369 CPU microseconds
 - 13.18 ms per message
- Comparing the number of pages that had to be skipped
 - Indexed = 0
 - Non-indexed = 3585

What queues are being used and how?

- High volume request and reply queue in the same resource pool
 - This is a case of 'define like' run amok
 - The request queue and reply queue for a high volume application were defined in the same storage class (same bufferpool and pageset)
 - By moving the reply queue to a different storage class, the resource usage was better distributed

High volume request and reply queue in the same resource pool



- Note this is often not seen until there is stress in one or more bufferpools due to volume.

Complete your sessions evaluation online at SHARE.org/BostonEval



What queues are being used and how?



- Overuse of Temporary dynamic queues
 - Often used for responses on both RYO and traditional monitoring tools
 - All queues created will be in the same resource pool
 - Quite expensive in CPU
- Temp dynamic queues are identifiable by their name
 - For example for the MQExplorer uses temporary dynamic queues. The name looks like this

AMQ.MQEXPLORER.1363497285

Complete your sessions evaluation online at SHARE.org/BostonEval



Temporary Dynamic Queues



```

Open name TEAMXX.MODEL
Base name AMQ.C9422A60F4386075
Queue indexed by NONE
Object type:Local Queue
Base type :Queue
First opened 12-03-2012 21:24:16.34
Last closed 23-09-2019 17:52:14.24
Page set ID      0, Buffer pool      0
Current opens    0, Total requests   10
Generated messages :      0
Persistent messages: GETs      0, PUTs      0, PUT1s      0
Put to waiting getter: PUT      0, PUT1      0
PUTs: Valid      3, Max size      0, Min size      9, Total bytes 27
-MQ call-      N      ET      CT      Susp      LOGW      PSET Epages skip expire
Open :      1      850      125      727
Close :      1      113      111      0
Put :      3      106      104      0
Inquire:      5      17      17
Maximum depth encountered      3
    
```

Complete your sessions evaluation online at SHARE.org/BostonEval



Permanent Queues



== Task token : 12-03-2012 21:24:23.42, 55FE03F0, 55FD0000

```

Open name TEAMXX.NOT.TEMP
Base name TEAMXX.NOT.TEMP
Queue indexed by NONE
Object type:Local Queue
Base type :Queue
First opened 12-03-2012 21:25:09.23
Last closed 18-10-2019 08:31:46.22
Page set ID      0, Buffer pool      0
Current opens    0, Total requests   10
Generated messages :      0
Persistent messages: GETs      0, PUTs      0, PUT1s      0
Put to waiting getter: PUT      0, PUT1      0
PUTs: Valid      3, Max size      3, Min size      9, Total bytes 27
-MQ call-      N      ET      CT      Susp      LOGW      PSET Epages skip expire
Open :      1      39      38      0
Close :      1      26      26      0
Put :      3      115      113      0
Inquire:      5      18      18
Maximum depth encountered      3
    
```

Complete your sessions evaluation online at SHARE.org/BostonEval



Temp vs. Permanent

- The CPU cost comparison
 - Verb TDQ Permanent
 - Open 125 38
 - Close 111 26
 - Put 104 113
 - Inquire 17 18
- The Elapsed Time comparison
 - Verb TDQ Permanent
 - Open 850 39
 - Close 113 26
 - Put 106 115
 - Inquire 17 18

Complete your sessions evaluation online at SHARE.org/BostonEval

Hunting down the culprit – finding a transaction in the SMF116

- Many times you want to look at the information from a CICS transaction or batch job
 - No way to turn SMF116 class 3 on for just one TX or job
 - Use SORT
 - Remember you will have 2 passes!
 - *First pass to sort out the 'short' records that the SMFDUMP program applies*
 - *Second pass to pull out the records for the transaction/batch job you want*

Complete your sessions evaluation online at SHARE.org/BostonEval

Finding a specific transaction or batch job



- In a group of millions of records, pulling the information for a specific transaction to 'map' its behavior can be critical in both problem resolution and performance issues
- The SMFDUMP program has few options for getting subsets of the data
- Using a simple sort is a quick solution to dividing up this massive volume into manageable groups

Complete your sessions evaluation online at SHARE.org/BostonEval



Finding a transaction



```
//*  
//* THIS GETS RID OF THE 'FIRST AND LAST' SMF RECORDS THAT CAUSE THE  
//* SORT TO COUGH UP BLOOD  
//*  
//SYSIN DD *  
  OMIT COND=(6,1,CH,LT,X'73')  
  SORT FIELDS=(19,4,CH,A)  
//
```

```
//SYSOUT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//*  
//* SELECT SMF116 BY TRANSACTION  
//*  
//SYSIN DD *  
  SORT FIELDS=(109,4,BI,A)  
  INCLUDE COND=(109,4,CH,EQ,C'ABCD')  
/*
```

Complete your sessions evaluation online at SHARE.org/BostonEval



Finding a Batch job

```
//*  
//* THIS GETS RID OF THE 'FIRST AND LAST' SMF RECORDS THAT CAUSE THE  
//* SORT TO COUGH UP BLOOD  
//*  
//SYSIN DD *  
  OMIT COND=(6,1,CH,CO,STRTG)  
  SORT FIELDS=(1,1,CH,A)  
//
```

```
//SYSUDUMP DD SYSOUT=*  
//*  
//* THIS PULLS THE SMF RECORD FOR A SPECIFIED BATCH JOB  
//*  
//SYSIN DD *  
  INCLUDE COND=(73,8,CH,EQ,C'ELKINSC2')  
  SORT FIELDS=(19,4,CH,A)  
/*
```

SMF116 and Long running tasks

- IF the long running task is started after the Class 3 trace
 - SMF 116 records will be cut at each SMF interval and at task end
- If the task is started before the trace is
 - No records are cut
 - APAR PM58798 has been taken on this



Summary

- The SMF data can be used in many ways to find patterns of use, problems with the queue managers, and programming problems.
- There are many other things within the data that are helpful, and more to come with the 7.1 interpretations and print programs. We hope those will be delivered soon!
- Thank you

Complete your sessions evaluation online at SHARE.org/BostonEval



The Dark Arts of MQ SMF Evaluation



Complete your sessions evaluation online at SHARE.org/BostonEval

