The Dark Arts of MQ SMF Evaluation

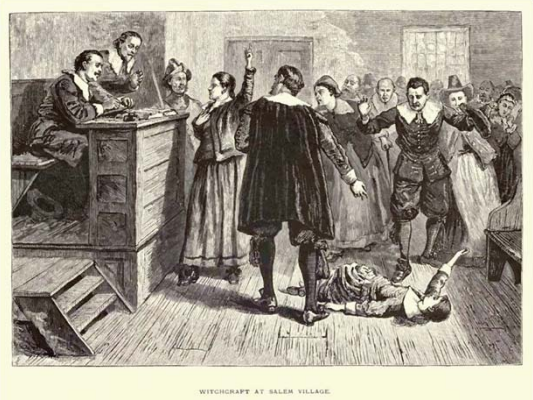
WebSphere software

Lyn Elkins - elkinsc@us.ibm.com

Mitch Johnson – mitchj@us.ibm.com

Wildfire
Technical Hands-On Workshops
IBM zGrowth Team

The witch trial – MQ is broken!



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
    - Other SMF115 data of interest
  - SMF116 Data
    - What queues are being used and how?
    - Can I find out which queues are the most active?
    - Pulling the data for one CICS transaction or batch job
    - Long running tasks
- Summary

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
    - Two records per queue manager per SMF interval (pre V8)
    - At least two records per queue manager per SMF interval (V8)
  - 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 problem

SMF 115 data – Hunting down the Culprit

- **Red Flags for Bufferpools**
  - SOS

QMGR	BP	NumBuff	%now	%low	dwt	dmc	stl	stla	sos
QML2	3	70000	18	0	109	198906	922354	1	50
QML2	3	70000	19	0	68	143872	367873	1	13

  - Freepages at 5% or less

Date	Time	QMGR	BP	NumBuff	%now	%low	dwt	dmc	stl	stla	sos
20113340	8:15:21	QML1	3	70000	98	5	9	27	32557	0	0
20113340	20:41:19	QML1	3	70000	95	5	2	364	61145	0	0

SMF 115 data

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

QMGR	BP	NumBuff	%now	%low	dwt	dmc	sti	stla	sos	
QML3	3	70000	16	0	58	210092	853991	1	0	
QML3	3	70000	22	3	132	36526	1232774	2	0	

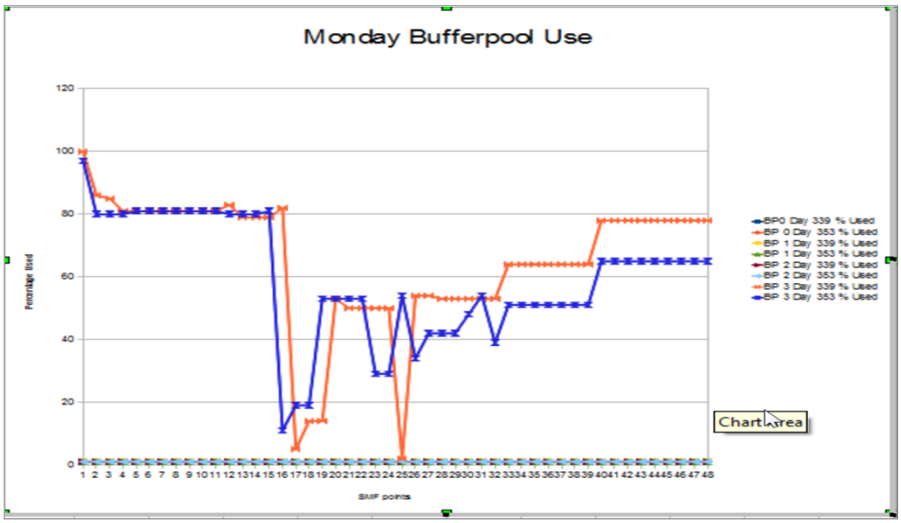
- The DMC count should be used in conjunction with the IMW field from the SMF115 report to see how many synchronous writes were actually performed.

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 counts, 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
MPX1	QML1	2010/09/29	15:32:18	1	15000	7233	15302	31695	0	0	0		0
MPX1	QML1	2010/09/29	15:32:18	2	40000	5980	39371	32569	116	17400	4350		0
MPX1	QML1	2010/09/29	15:32:18	3	20000	3281	18921	29093	0	0	0		0
MPX1	QML1	2010/09/29	15:32:18	4	30000	29999	0	68	0	0	0		0
MPX1	QML1	2010/09/29	15:32:18	5	30000	29999	0	134	0	0	0		0
MPX1	QML1	2010/09/29	15:32:18	9	20000	2583	17521	22273	338	4976	1244		0

SMF115 – Bufferpool Trends and Analysis



SMF 115 data

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

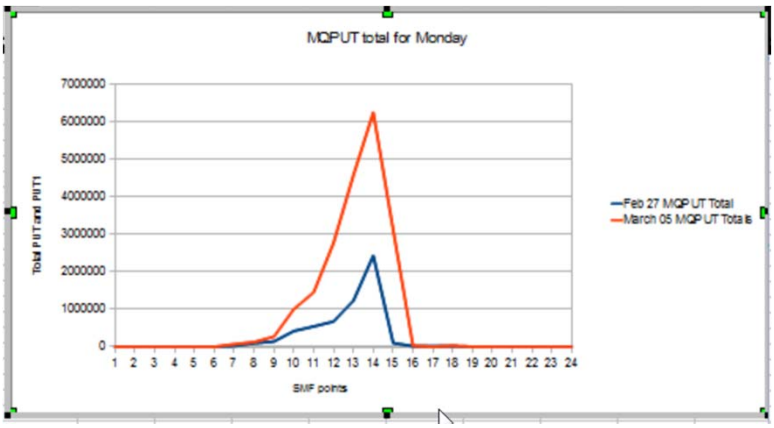
QMGR	BP	NumBuff	%now	%low	dwt	dmc	stl	stla	sos
QML4	2	70000	53	19	0	0	46571	0	0
QML4	3	70000	98	20	0	0	46028	0	0
QML4	3	70000	75	20	0	0	0	0	0

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	Inq1	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

Message Manager - Trend Chart



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

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

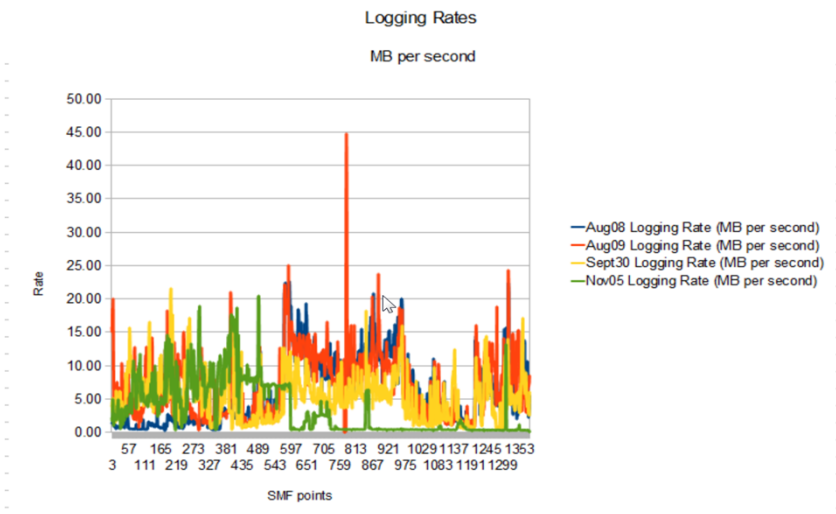
QMGR	wr_wait	wr_nwait	Aug09 Force Writes	Aug09 Log Buffer Waits	read_buf	read_act	read_arc	r_delay	N_CheckP	Aug09 Num I/O	Aug09 Control Intervals Written	paging
QML1	0	569925	339	1	0	0	0	0	0	22020	241748	0
QML1	0	621641	337	0	0	0	0	0	0	23758	230944	0
QML1	0	753611	363	1	0	0	0	0	0	27490	285402	0

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

Logging Rates - Charted



Log Manager CSV file from new MP1B

- 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.

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

SMF 115 data – CF Statistics

A	B	C	D	E	F	G	H	I	J	K	L
Date	Time	QMGR	CFN	CFname	Num_E	Avg_E_T	%Redrive	Num_M	Avg_M_T	%Redrive	Num_full
2013351	00:08:05.45	QML1	0	CSQ_ADMIN	10398	13	0	16	615	0	0
2013351	00:08:05.45	QML1	2	APPLPRD01	36852	23	0	1327	44	0	0
2013351	00:08:05.45	QML1	3	APPLPRD02	5137	33	0	0	0	0	0
2013351	00:08:05.45	QML1	4	APPLPRD03	63	17	0	2	16	0	0

- The CF data from MQ should be used in conjunction with the Coupling Facility Activity Reports (CFRM).
- The average elapsed time is reported in microseconds, and in this example is low. As it happens the CF in use is 'local' – in the same CEC, so they should be low.
- Recommendation is to chart the values over time, like the other statistics looking for anomalies and use patterns.

SMF 115 data – DB2 Statistics

Date	Time	QMGR	Jobname	Count	Avg_ET_T	Avg_ET_S	Max_Ti_T	Max_Ti_S
2013352	10:28:25.19	QML1	QML1CHIN	2	3796	3655	5548	5476
2013352	10:25:26.04	QML1	LYNBTCH	1	3547	3499	3547	3499
2013352	10:25:26.05	QML1	LYNBTCH	1	4323	4303	4323	4303
2013352	10:26:41.31	QML1	LYNBTCH	1	12765	12628	12765	12628

- This data includes the average am maximum times spent on DB2 requests
  - Average elapsed time on the DB2 task
  - Average elapsed time on the DB2 server
  - Maximum elapsed time on the DB2 task
  - Maximum elapsed time on the DB2 server
- Recommendation is to chart the values over time, like the other statistics looking for anomalies and use patterns.

SMF 115 data – SDB2 Statistics

- The additional DB2 information shows the
  - Maximum depth of queues requests into DB2
  - Whether there were deadlocks
  - As with the other statistics, these should be charted to show usage patterns and detect anomalies

Date	Time	QMGR	Max_Depth	Num_deadlock
2013351	00:08:24.40	QML1	1	0
2013351	00:44:52.34	QML1	1	0
2013351	01:14:46.46	QML1	1	0
2013351	01:44:40.57	QML1	1	0

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.

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'

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

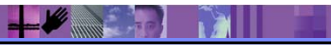
IBM MQ for z/OS Wildfire Workshop

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

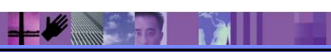
Non-Indexed Queue retrieval

```
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 :          0
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, Su cessful destructive          28
Time on queue : Max 4583.738054,  Min  257.434981, Avg 3958.326341
-MQ call-          N         ET         CT        Susp       LOGW       PSET Epages   skip expire
 Get    :          28        384        369          0          0          0      0      3585      0
 Inquire:          28         22         21
Maximum depth encountered        258
```



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 15:16:58.35
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, Su cessful destructive          27
Time on queue : Max 4788.946117,  Min  422.846309, Avg 4288.437716
-MQ call-          N         ET         CT        Susp       LOGW       PSET Epages   skip expire
 Get    :          27        105         99          0          0          0      0         0      0
 Inquire:          26         21         20
Maximum depth encountered        258
```



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

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

Temporary Dynamic Queues

```
Open name TEAMXX.MODEL                          Object type:Local Queue
Base name AMQ.C9422A60F4386075                  Base type  :Queue
Queue indexed by NONE
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        9,  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          0
 Inquire:          5         17         17
Maximum depth encountered         3
```

Permanent Queues

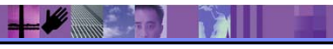
```
== Task token : 12-03-2012 21:24:23.42,  55FE03F0,  55FD0000

Open name TEAMXX.NOT.TEMP                          Object type:Local Queue
Base name TEAMXX.NOT.TEMP                          Base type  :Queue
Queue indexed by NONE
First opened 12-03-2012 21:25:09.23
Last closed  18-10-2019 00:31:46.22
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        9,  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          0
 Inquire:        5        18        18
Maximum depth encountered          3
```


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

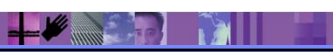
What queues are actually in use?

Date	Time	Jobname	Queue	Get	ValidGet	Bytes	MaxGet	MinGet	MaxTOQ
2013352	10:27:25.50	MPX1CICS	LYNE.TEST1	6	5	1000	200	200	2.1E+04
2013352	10:28:39.49	MPX1CICS	LYNE.TEST1	6	5	1000	200	200	1.6E+04
2013352	10:25:35.53	MPX1CICS	LYNE.TEST2	9	8	2096	262	262	1.3E+04
2013352	10:29:36.56	MPX1CICS	LYNE.TEST2	8	7	1834	262	262	2.2E+04
2013352	10:25:35.53	MPX1CICS	LYNE.TEST2	7	6	1572	262	262	1.3E+04
2013352	10:27:57.52	MPX1CICS	LYNE.TEST2	7	6	1572	262	262	1.4E+04
2013352	10:28:09.53	MPX1CICS	LYNE.TEST2	7	6	1572	262	262	1.6E+04
2013352	10:25:19.51	MPX1CICS	LYNE.TEST2	6	5	1310	262	262	1.1E+04
2013352	10:27:57.52	MPX1CICS	LYNE.TEST2	6	5	1310	262	262	1.4E+04
2013352	10:25:02.29	MPX1CICS	LYNE.ERRORS	7	6	180942	30157	30157	6.5E+03
2013352	10:29:47.74	MPX1CICS	LYNE.ERRORS	7	6	180942	30157	30157	3.7E+03
2013352	10:27:17.05	MPX1CICS	LYNE.INPUT	10	9	2925	325	325	8.6E+04



What is the volume on the different queues?

Queue Name	Bufferpool	Pageset or CF Structure	Number of References	Total Number of GETs	Valid GETs	Bytes Retrieved	Average Message Size
LYN.TEST1		APPLPRD01	1086	2172	1086	1650720	1520.00
LYN.TEST2		APPLPRD02	2843	3882	1069	1710400	1600.00
LYN.TEST3		APPLPRD01	3394	3394	947	1462168	1544.00
LYN.TEST4		APPLPRD01	946	1892	946	1437920	1520.00
LYN.TEST5		APPLPRD03	3191	3191	881	1360264	1544.00
LYN.TEST6	3	4	672	672	672	167448	249.18



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 to use option VLSHRT to omit short records!

```
//SYSIN  DD *
OPTION VLSHRT
INCLUDE COND=(73,8,CH,EQ,C'MOVER   ')
SORT FIELDS=(19,4,CH,A)
/*
```


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')
/*
```

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,LT,X'73')
  SORT FIELDS=(19,4,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)
/*
```

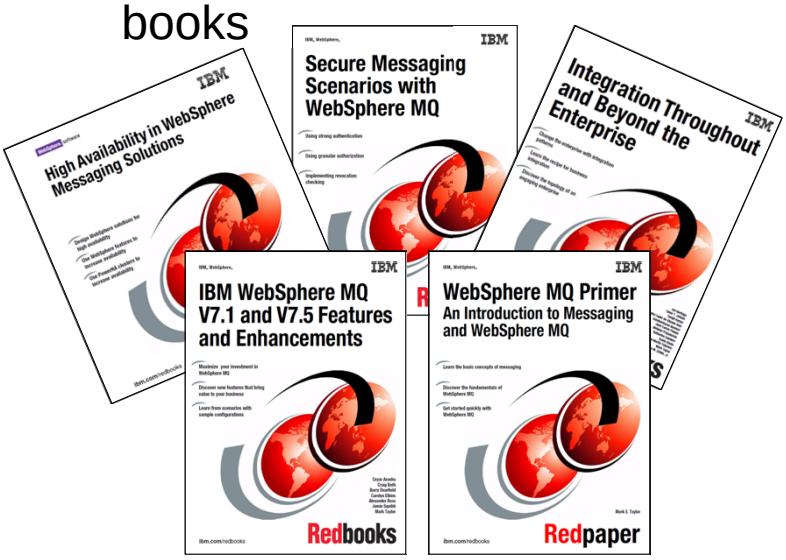

SMF116 and Long running tasks

- 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

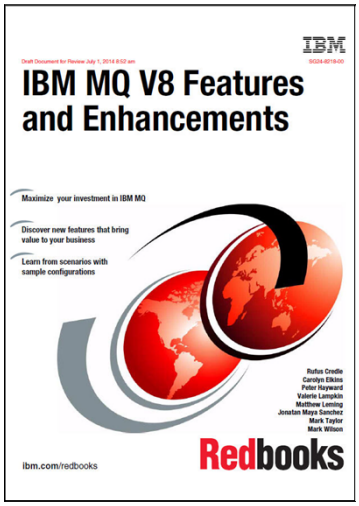
Summary

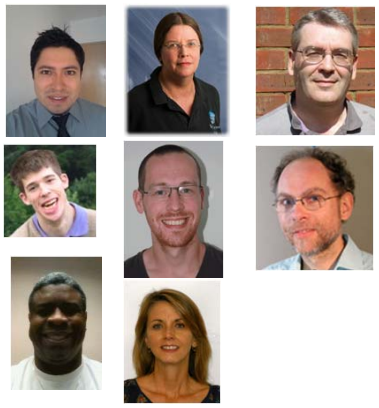
- The SMF data can be used in many ways to find patterns of use, problems with the queue managers, and programming problems.
- There are many other things within the data that are helpful, and more to come with the 7.1 and V8 interpretations and print programs.
- Thank you

Shameless MQ Promotion – real books



And …





https://www.redbooks.ibm.com/Redbooks.nsf/RedpieceAbstracts/sg248218.html