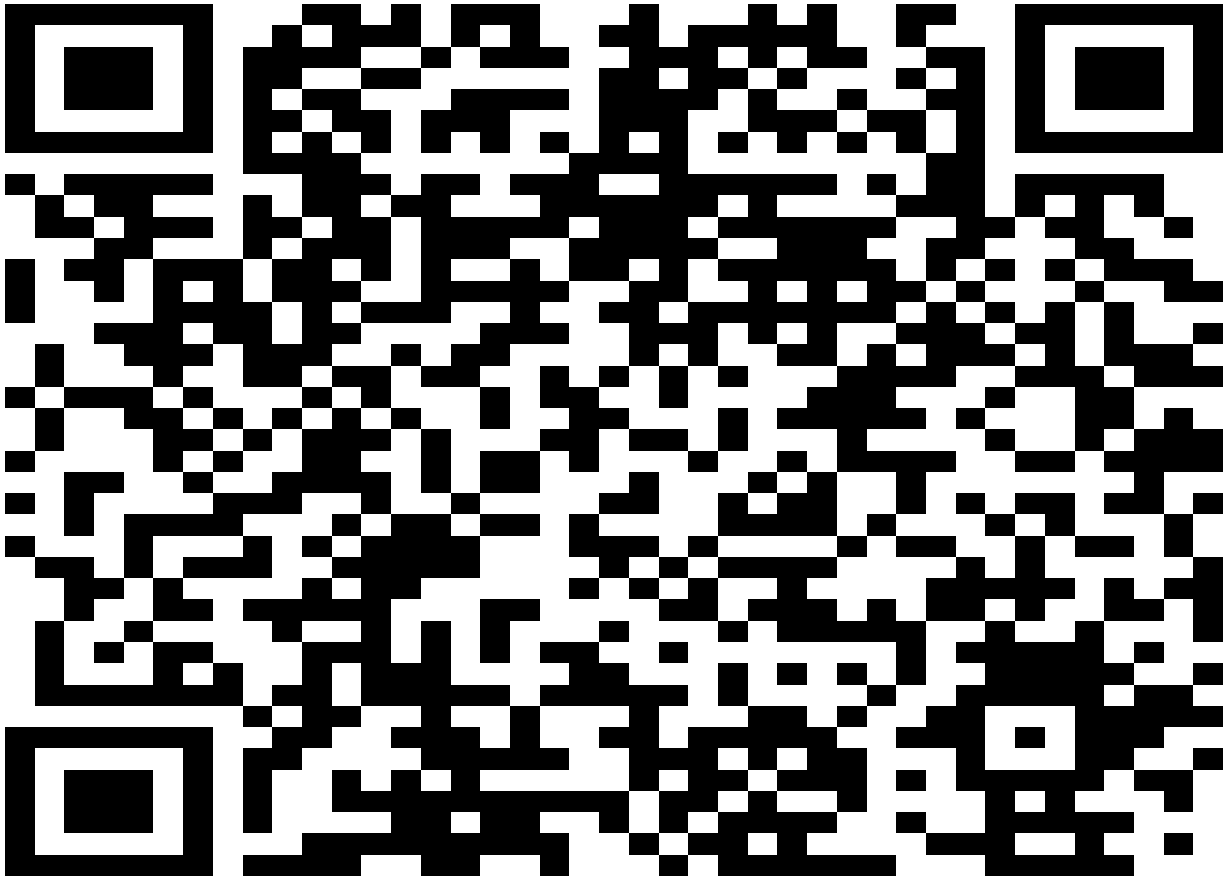
# Paging Dr. MQ - Health Check Your Queue Managers to Ensure They Won't Be Calling in Sick! [z/OS] – Session 17046

*Lyn Elkins*
elkinsc@us.ibm.com

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

# Code!



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

# Is MQ sick? What can we do about it!

# What is a MQ for z/OS Health Check?

- A health check is a point in time evaluation of the queue manager and channel initiator.
  - It is to provide singular focus and evaluation
- It does not replace:
  - Real time monitoring
  - Regular reviews of statistics
  - Regular reviews of accounting data


CRASHED
SLOW
QUEUEI NG
QMGRS
IN MESSAGING TROUBLE?
Better
call
Lyn!
HEALTH
CHECKS
SMF
NOW
CPU

# Objectives

- This session is to delve a bit deeper into how the team formerly known as ATS (now something less easy to remember) perform a health check and how you can perform your own.
- It is almost exciting as watching moss grow.

# What data needs to be collected?

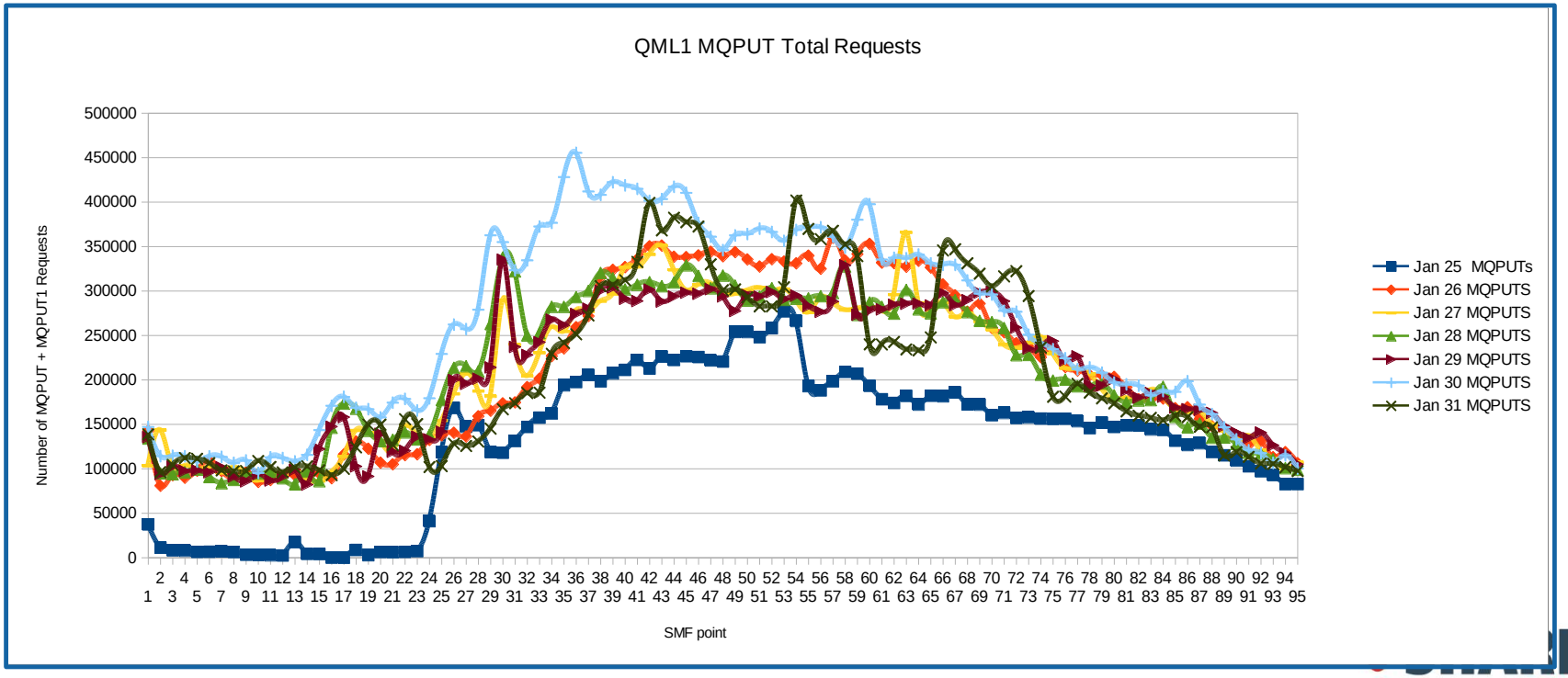
- The data typically requested for an MQ for z/OS health check typically includes:
  - One week of MQ Statistics data – SMF 115
    - For customers on MQ V8 this should include classes 1 and 4
    - For customers prior to V8, if gathering data about channels using the MQCMD program (or similar) if using is also helpful
  - One high volume day's JES logs for the queue manager and channel initiator

# What data needs to be collected? More

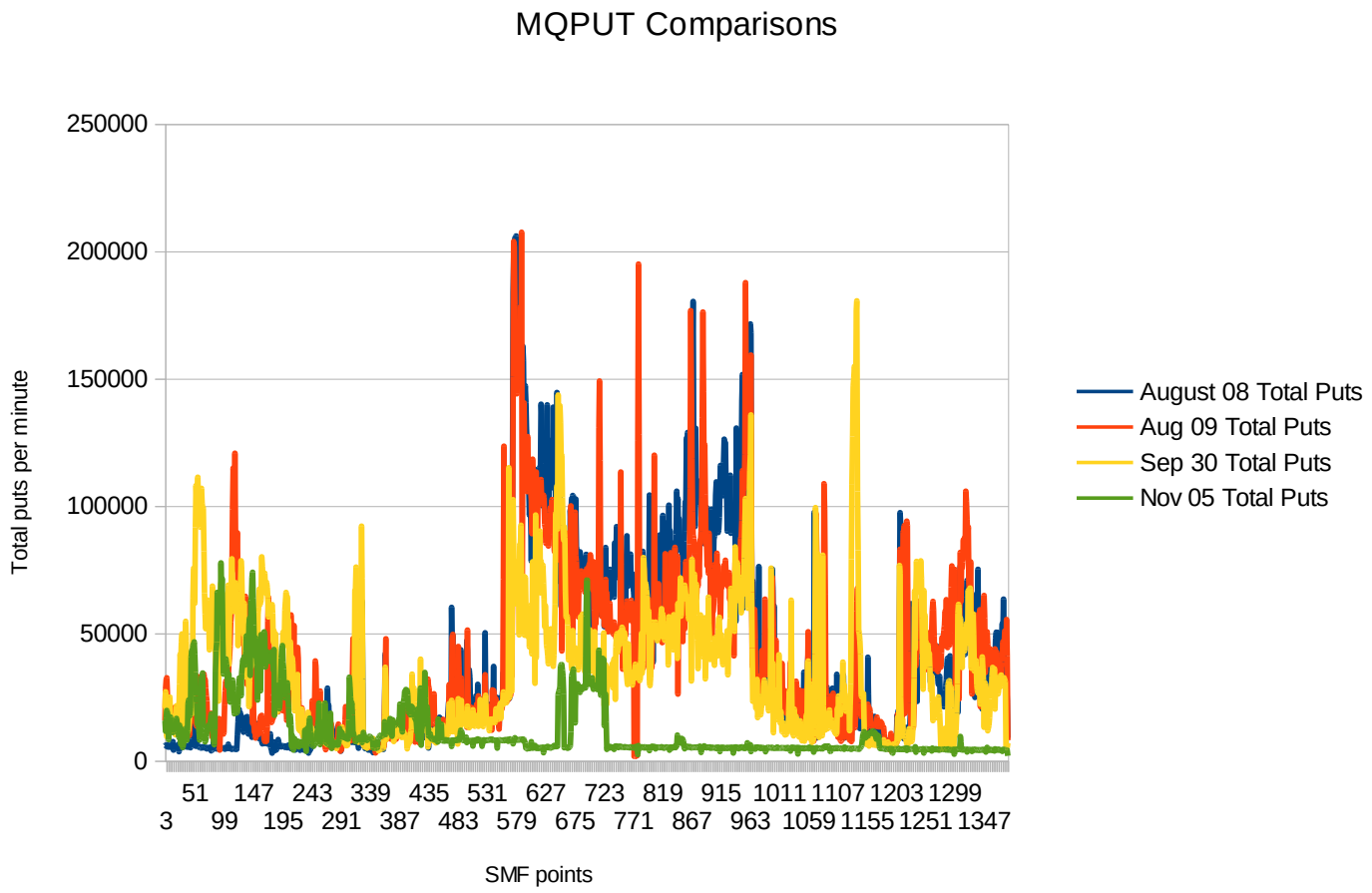
- Class 3 accounting for one or more busy, but not peak SMF intervals
  - If there are multiple peaks during a day, this data should be gathered during either the typical ‘ramp up’ or ‘ramp down’
  - A good cross section of tasks using MQ resources is necessary to evaluate the application’s use of MQ
- If on V8, Class 4 accounting for one or more busy but not peak SMF intervals.

# What is a ‘busy but not peak period’?

- A sample week of MQPUT and MQPUT1 may illustrate patterns of use



# Another example of workload patterns



# How do I find a ‘busy but not peak period’?

- Keep in mind (aka do as I say not as I do)
  - There may be multiple well defined peaks during a day
    - Batch and online workloads are typical
    - Multiple samples may be required to demonstrate all use
  - Some queue managers may have less well defined peaks – the message rates are more constant then the illustrated queue managers
  - Others may have well defined and predictable peaks

# Finding the Symptoms

- To find out how healthy an individual queue manager is, there are a number of important steps:
  - Review the JES logs
    - Look for specific messages that indicate issues, some examples:
      - CSQJ032E - APPROACHING END OF THE LOG RBA RANGE OF FFFFFFFFFFFF. CURRENT LOG RBA IS .
      - CSQJ111A - OUT OF SPACE IN ACTIVE LOG DATA SETS
      - CSQJ112E - INSUFFICIENT ACTIVE LOG DATA SETS DEFINED IN BSDS
    - Calculate the number of log switches during a peak interval using the CSQJ033I message

# Finding the Symptoms - continued

- JES log evaluation continued:
  - Track the storage use by capturing the CSQY220I messages
    - Example:
      CSQY220I QML1 CSQSCTL Queue manager storage usage: 110 local storage: used 597MB, free 1141MB: above bar: used 1GB, free >10GB
  - Look for long running applications:
    - Long running UOWs reported via CSQJ160I
    - Log shunting reported via CSQR026I
  - Look for messages that you have never seen before!
    - Your real time alerting system may not be looking at all the important messages.

# Finding symptoms - 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
  - SupportPac MP1B, no matter which version, provides at least 2 views into the MQ statistics
    - Report form
    - Column or CSV form
      - This is used to import into spreadsheets for analysis
    - Both are necessary because the CSV form may be missing fields you need to review when looking for a problem.

# Finding the symptoms - 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
      - Differences between new and old for a CICS transaction are documented in https://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD106135
    - Other files, much like the ‘old’ MQCSMF output

# Determining the health

# SMF 115 – using the data

- The CSV file or the column format can be downloaded from z/OS and then imported into a spreadsheet for evaluation.
- For example, the log data loaded into a spreadsheet looks as follows:

| Date | Time | QMGR | wr_wait | wr_nwait | Aug08 Force Writes | Aug08 Log Buffer Waits | read_buf | read_act | read_arc | r_delay | N_CheckP | Aug08 Num I/O | Aug08 Control Intervals Written | paging |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2011220 | 00:02:01.06 | QML1 | 0 | 69242 | 447 | 0 | 0 | 0 | 0 | 0 | 0 | 6726 | 20658 | 0 |
| 2011220 | 00:03:01.05 | QML1 | 0 | 74069 | 403 | 0 | 0 | 0 | 0 | 0 | 0 | 6866 | 22446 | 0 |
| 2011220 | 00:04:01.04 | QML1 | 0 | 72830 | 453 | 0 | 0 | 0 | 0 | 0 | 0 | 6726 | 22550 | 0 |
| 2011220 | 00:05:01.08 | QML1 | 0 | 70513 | 478 | 0 | 0 | 0 | 0 | 0 | 0 | 6672 | 20870 | 0 |
| 2011220 | 00:06:01.07 | QML1 | 0 | 77959 | 489 | 0 | 0 | 0 | 0 | 0 | 0 | 7166 | 23458 | 0 |
| 2011220 | 00:07:01.06 | QML1 | 0 | 60920 | 483 | 0 | 0 | 0 | 0 | 0 | 0 | 6342 | 17732 | 0 |
| 2011220 | 00:08:01.06 | QML1 | 0 | 65818 | 444 | 0 | 0 | 0 | 0 | 0 | 0 | 6662 | 18960 | 0 |
| 2011220 | 00:09:01.03 | QML1 | 0 | 68325 | 476 | 0 | 0 | 0 | 0 | 0 | 0 | 6740 | 19982 | 0 |
| 2011220 | 00:10:01.08 | QML1 | 0 | 50712 | 478 | 0 | 0 | 0 | 0 | 0 | 0 | 6000 | 12088 | 0 |

# SMF115 – Using the data

- By itself the SMF 115 data for many of the internal component managers are useful, but some manipulation of the raw data can make it much more consumable and useful.
- Examples include:
  - The message manager summaries as shown earlier
  - Calculating the log data written per second
  - Charting, making it easier to spot patterns and trends
- Some of the calculated data is done for you in the ‘new’ MP1B, but beware – it uses hard coded values for intervals, etc. which can skew results.

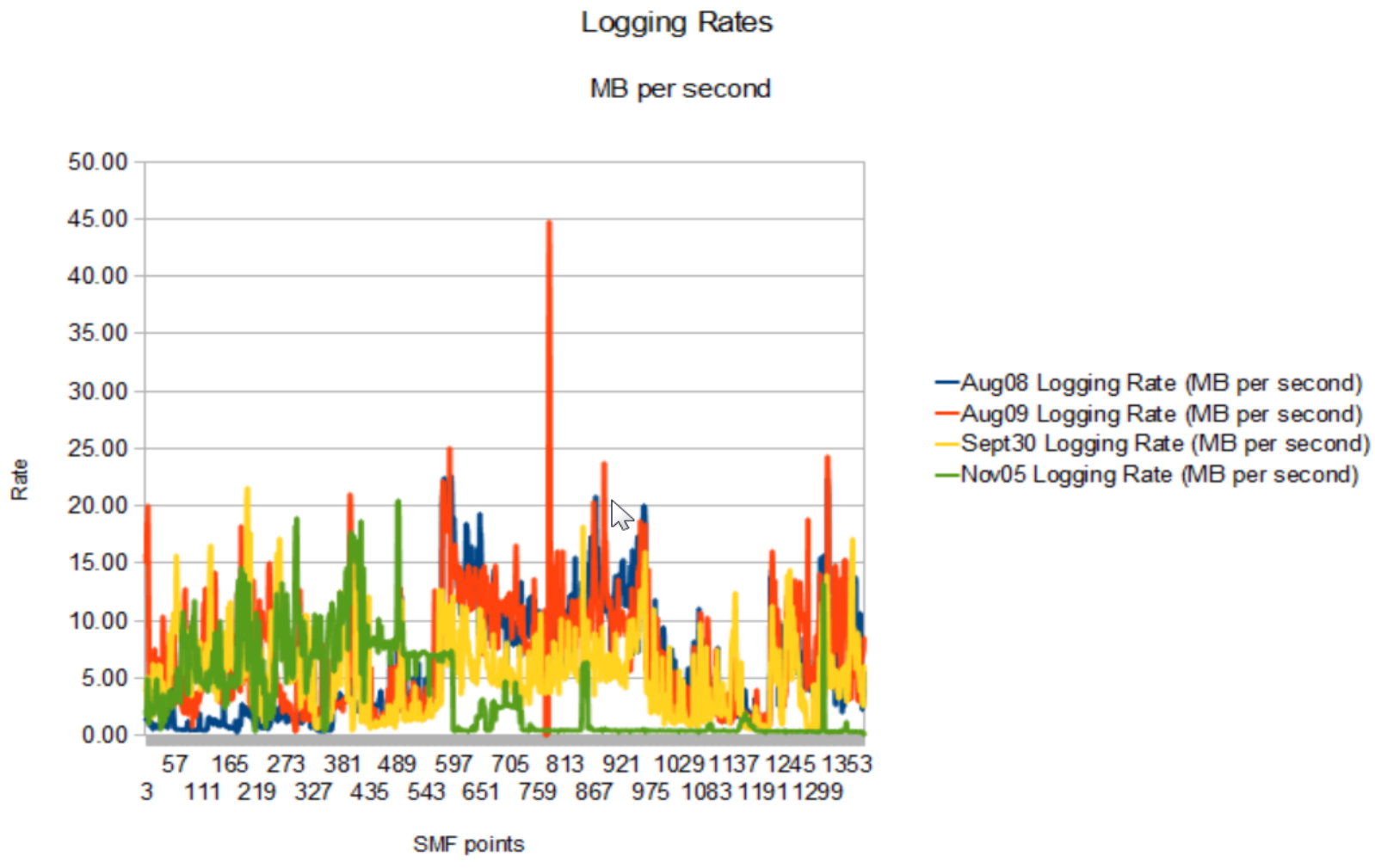
# SMF115 – Using the data – continued

- Example of using a spreadsheet to calculate the log writes:

| Num I/O | Num_CI_W | paging | | Seconds | Interval Duration | | Log writes in MB per second |
|---|---|---|---|---|---|---|---|
| 142446 | 142460 | 0 | | 980 | | | |
| 13352 | 13352 | 0 | | 1877 | 897 | | 0.06 |
| 12420 | 12420 | 0 | | 2775 | 898 | | 0.05 |
| 10956 | 10956 | 0 | | 3670 | 895 | | 0.05 |
| 10484 | 10484 | 0 | | 4567 | 897 | | 0.05 |
| 9274 | 9274 | 0 | | 5463 | 896 | | 0.04 |
| 8946 | 8946 | 0 | | 6359 | 896 | | 0.04 |
| 8864 | 8864 | 0 | | 7256 | 897 | | 0.04 |
| 1476 | 1476 | 0 | | 8154 | 898 | | 0.01 |

- The formulas used to calculate seconds and duration:
  - Seconds =SUM(HOUR(B2)*3600,(MINUTE(B2)*60),SECOND(B2))
  - Interval duration =(Q3-Q2)
  - Interval crossing a date =SUM((86399-Q96),Q97)
- The formula for log writes in MB/Second
  - =ROUND(((N3*4096)/(1024*1024))/(R3),2)
  - Where N3 = the number of CI Writes column and R3 is the Interval duration

# Logging Rates - Charted



# Reviewing the SMF 115 data – Are there problems?

- While not always the culprit, tuning the buffer pools has been a never ending cycle of opportunities
  - Lazy queue definitions
    - I copy what works, might not be what is best for the queue manager
  - Workload pattern changes
    - What flows today, might become a log jam without any warning
      - Well except the business knew they were opening 2600 new branches on Tuesday

# SMF 115 data – Urgent symptoms – Call 911

- **Unless there has been an unexpected spike in volume, this should never been seen!**
  - SOS
  - Freepages at 5% or less

| 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 | 88 | 143872 | 387873 | 1 | 13 |

| Date | Time | QMGR | BP | NumBuff | %now | %low | dwt | dmc | stl | stla | sos |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2011334 | 08:15:21. | QML1 | 3 | 70000 | 98 | 5 | 9 | 27 | 32557 | 0 | 0 |
| 2011334 | 20:41:19. | QML1 | 3 | 70000 | 95 | 5 | 2 | 384 | 61145 | 0 | 0 |

# SMF 115 data – Urgent symptoms

- **Red Flags for Bufferpools** - Continued
  - DMC – synchronous write process is requested
    - Note that it did not run this often, but this is the number of times that conditions were right!

| QMGR | BP | NumBuff | %now | %low | dwt | dmc | stl | stla | sos |
|---|---|---|---|---|---|---|---|---|---|
| QML3 | 3 | 70000 | 16 | 0 | 58 | 210092 | 853991 | 1 | 0 |
| QML3 | 3 | 70000 | 22 | 3 | 182 | 36526 | 1232774 | 2 | 0 |

# SMF 115 data – Clairvoyance – Ask Doctor

- **May be approaching a problem**
  - Consistently Approaching/Achieving 20 % Free pages

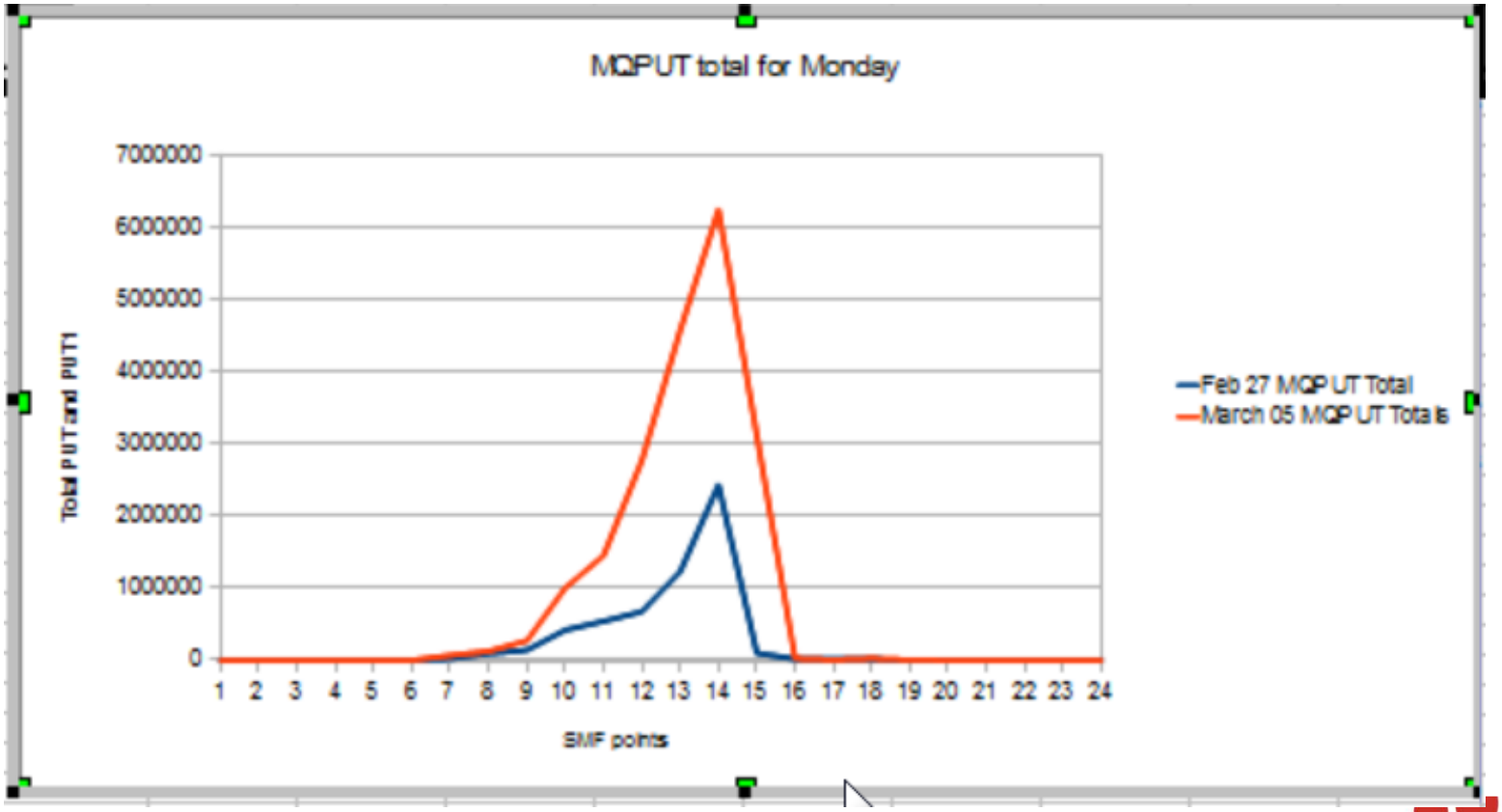
| QMGR | BP | NumBuff | %now | %low | dwt | dmc | stl | stla | sos |
|---|---|---|---|---|---|---|---|---|---|
| QML4 | 2 | 70000 | 53 | 19 | 0 | 0 | 46571 | 0 | 0 |
| QML4 | 3 | 70000 | 98 | 20 | 0 | 0 | 46028 | 0 | 0 |
| QML4 | 3 | 70000 | 75 | 20 | 0 | 0 | 0 | 0 | 0 |

# SMF 115 data – Reviewing the Message Manager information for gross use numbers

- 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



# Looking for more symptoms? Try the SMF116 Class 3 data

- Reviewing this copious data can feel like searching for the spell to turn lead into gold. Better to think of it as panning for gold
- As an MQ 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 – Looking for things that make us less healthy

- The SMF 116 class 3 data may provide information about the health of the queue manager and the applications using MQ
  - For example,  this data can be used:
    - Identify heavily used queues that can benefit from splitting across resource pools
    - Identify shared queues that should be on the same structure
    - Identify queues that should be indexed
    - Identify overly long response times, typically from I/O
    - Identify applications that are consuming excessive CPU due to bad verbs

# What queues are being used and how?

- SMF116 class 3 data shows the use of queues
- Helpful because even as an MQ admin, it may be a challenge to find out where the queues are.
  - For example, if an application opens an alias, that points to another…this data provides the base name of the object
- Some specific opportunities for prevention:
  - Non-indexed queues
  - High volume request/reply queues in same resource pool
  - Overuse of Temporary dynamic queues

# Queue Indexing – an opportunity to reduce the CPU fever

- 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 4583.730054,  Min  257.434901, Avg 3958.326341
-MQ call-          N         ET         CT         Susp       LOGW       PSET Epages   skip expire
 Get    :          28        384        369           0          0          0      0     3505      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 15:16:50.35
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, Su cessful 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      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
  - Non-indexed 28 messages at 369 CPU microseconds
  - An average of 270 microseconds less expensive per get
  - What's a microsecond or two between friends?
- Comparing the number of pages that had to be skipped
  - Indexed = 0
  - Non-indexed = 3585

# Another infection – concentration of resources

- 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

# GET queue use from SMF 116 class 3 data

| Queue | Get | ValidGet | Bytes | MaxGet | MinGet | MaxTOQ | Total Gets | Total Valid Gets | Total Bytes | Average Message Size | Buffer pool/ Structure | Page Set |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TESTQ1,QL | 1 | 1 | 15000 | 15000 | 15000 | 1.1E+04 | 5 | 5 | 75000 | 15000 | 3 | 3 |
| TESTQ2,QL | 2 | 1 | 161 | 161 | 161 | 2.1E+03 | 330 | 165 | 120777 | 731.98 | 3 | 3 |
| TESTQ3.QL | 55 | 54 | 5886 | 109 | 109 | 3.2E+04 | 55 | 54 | 5886 | 109 | 3 | 3 |
| TESTQ4.QL | 1 | 1 | 128 | 128 | 128 | 3.5E+05 | 50 | 50 | 6400 | 128 | 3 | 3 |
| TESTQ5.QL | 1 | 1 | 128 | 128 | 128 | 3.2E+05 | 50 | 50 | 6400 | 128 | 3 | 3 |
| TESTQ6.QL | 1 | 1 | 128 | 128 | 128 | 2.4E+05 | 50 | 50 | 6400 | 128 | 3 | 3 |
| TEST.INITQ1 | 48 | 24 | 16416 | 684 | 684 | 4.4E+02 | 352 | 176 | 120384 | 684 | 1 | 1 |
| TEST.INITQ2 | 64 | 32 | 21888 | 684 | 684 | 2.5E+02 | 3826 | 1913 | 1308492 | 684 | 1 | 1 |
| TEST.INITQ3 | 170 | 85 | 58140 | 684 | 684 | 3.0E+02 | 1650 | 825 | 564300 | 684 | 1 | 1 |
| TEST.INITQ3 | 20484 | 10252 | 7012368 | 684 | 684 | 6.3E+04 | 106138 | 53142 | 36349128 | 684 | 1 | 1 |
| TEST.INITIQ4 | 172 | 86 | 58824 | 684 | 684 | 1.4E+04 | 1466 | 733 | 501372 | 684 | 1 | 1 |
| SYSTEM.CHANNEL.INITQ | 77 | 34 | 26656 | 784 | 784 | 4.7E+02 | 2216 | 989 | 775596 | 784.22 | 1 | 1 |
| SYSTEM.CHANNEL.SYNC( | 2 | 2 | 856 | 428 | 428 | 2.1E+08 | 189 | 176 | 75328 | 428 | 1 | 1 |

# PUT queue use from SMF 116 class 3 data

| Queue | Number valid PUTs | Number of Bytes Put | Average Message Size | Buffer pool/ Structure | Page Set |
|---|---|---|---|---|---|
| TESTQ3.QL | 7 | 1282 | 183.14 | 3 | 3 |
| TESTQ4.QL | 54 | 5886 | 109 | 3 | 3 |
| ECHO.QL | 48 | 6144 | 128 | 3 | 3 |
| SYSTEM.CHANNEL.INITQ | 4 | 3456 | 864 | 1 | 1 |
| SYSTEM.CHANNEL.SYNCQ | 174 | 74472 | 428 | 1 | 1 |
| SYSTEM.CLUSTER.COMMAND.QUEUE | 2 | 1000 | 500 | 1 | 1 |
| SYSTEM.COMMAND.INPUT | 919 | 1257221 | 1368.03 | 1 | 1 |

# Possibly mild, possibly virulent symptom

- Use (or 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.C1363497285

# 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

# Looking for a specific microbe – or one CICS transaction in the SMF116 data

- 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 queue, transaction, or job
  - Use SORT
    - Remember you will have to omit the short records that the SMF dump program always includes – use the VLSHRT

```
//SYSOUT   DD   SYSOUT=*
//SORTIN   DD   DISP=SHR,DSN=*.SEL.OUTDD1
//SORTOUT  DD   DISP=(,CATLG),DCB=(RECFM=VBS,BLKSIZE=4096,LRECL=32760),
//           DSN=ELKINSC.SMFCSQ5.AUG01A.OUT,
//           SPACE=(CYL,(10,10))
//SYSIN    DD   *
  OPTION VLSHRT
  INCLUDE COND=(19,4,CH,EQ,C'CSQ5')
  SORT FIELDS=COPY
/*
```

# Finding  a transaction

```
//S1        EXEC PGM=SORT
//SYSOUT    DD   SYSOUT=*
//SORTIN    DD   DISP=SHR,DSN=*.SEL.OUTDD1
//SORTOUT   DD   DISP=(,CATLG),DCB=(RECFM=VBS,BLKSIZE=4096,LRECL=32760),
//             DSN=ELKINSC.SMFCSQ5.AUG01A.OUT,
//             SPACE=(CYL,(10,10))
//SYSIN     DD   *
  OPTION VLSHRT
  INCLUDE COND=(109,4,CH,EQ,C'ABCD')
  SORT FIELDS=(109,4,BI,A)
/*
```

# Finding a Batch job

```
//SYSOUT   DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//SYSIN  DD *
  OPTION VLSHRT
  INCLUDE COND=(73,8,CH,EQ,C'MOVER    '
  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

- Performing your own periodic MQ health checks can help you focus on existing issues, or sharpen your foretelling.

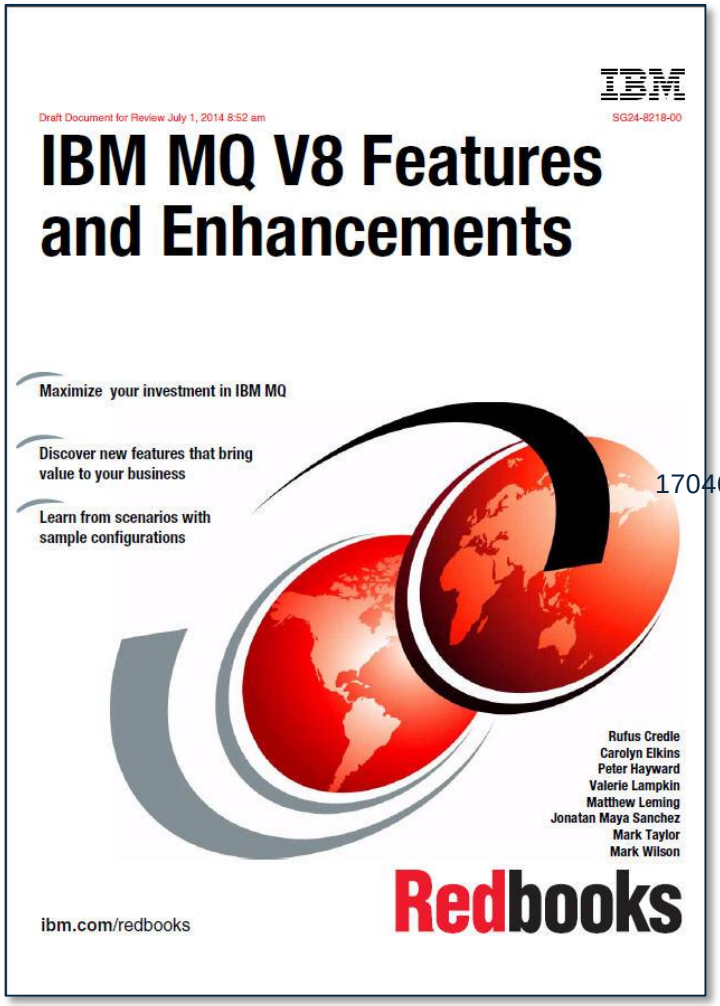
- Thank you

# This was Session # 17046. The rest of the week ……

| | Monday | Tuesday | Wednesday | Thursday | Friday |
|---|---|---|---|---|---|
| 08:30 | | | 17060: Understanding MQ Deployment Choices and Use Cases | 17051: Application Programming with MQ Verbs [z/OS & Distributed] | 16544: Why Shouldn't I Be Able To Open This Queue? MQ and CICS Security Topics |
| 10:00 | 17036: Introduction to MQ - Can MQ Really Make My Life Easier? [z/OS & Distributed] | | 17052: MQ Beyond the Basics - Advanced API and Internals Overview [z/OS & Distributed]<br>17035: MQ for z/OS, Using and Abusing New Hardware and the New V8 Features [z/OS] | 17054: Nobody Uses Files Any More do They? New Technologies for Old Technology, File Processing in MQ MFT and IIB [z/OS & Distributed] | 17057: Not Just Migrating, but Picking up New Enhancements as You Go - We've Given You the Shotgun, You Know Where Your Feet Are [z/OS & Distributed] |
| 11:15 | 17041: First Steps with IBM Integration Bus: Application Integration in the New World [z/OS & Distributed] | | 16732: MQ V8 Hands- on Labs! MQ V8 with CICS and COBOL<br>YOU ARE HERE | 17046: Paging Dr. MQ - Health Check Your Queue Managers to Ensure They Won't Be Calling in Sick! [z/OS] | 17053: MQ & DB2 – MQ Verbs in DB2 & InfoSphere Data Replication (Q Replication) Performance [z/OS] |
| 01:45 | 17037: All About the New MQ V8 [z/OS & Distributed] | 17034: MQ Security: New V8 Features Deep Dive [z/OS & Distributed] | 17040: Using IBM WebSphere Application Server and IBM MQ Together [z/OS & Distributed] | 17062: End to End Security of My Queue Manager on z/OS [z/OS] | All sessions in Seneca unless otherwise noted. |
| 03:15 | 17042: What's New in IBM Integration Bus [z/OS & Distributed] | 17065: Under the hood of IBM Integration Bus on z/OS - WLM, SMF, AT-TLS, and more [z/OS] | 17043: The Do's and Don'ts of IBM Integration Bus Performance [z/OS & Distributed] | 17039: Clustering Queue Managers - Making Life Easier by Automating Administration and Scaling for Performance [z/OS & Distributed] | |
| 04:30 | 17059: IBM MQ: Are z/OS & Distributed Platforms like Oil & Water? [z/OS & Distributed] | 17055: What's the Cloud Going to Do to My MQ Network? | 17044: But Wait, There's More MQ SMF Data Now?!?! - Monitoring your Channels Using V8's New Chinit SMF Data [z/OS] | 17068: Monitoring and Auditing MQ [z/OS & Distributed] | |

# Further information in real books

# And … already available (draft)




17046

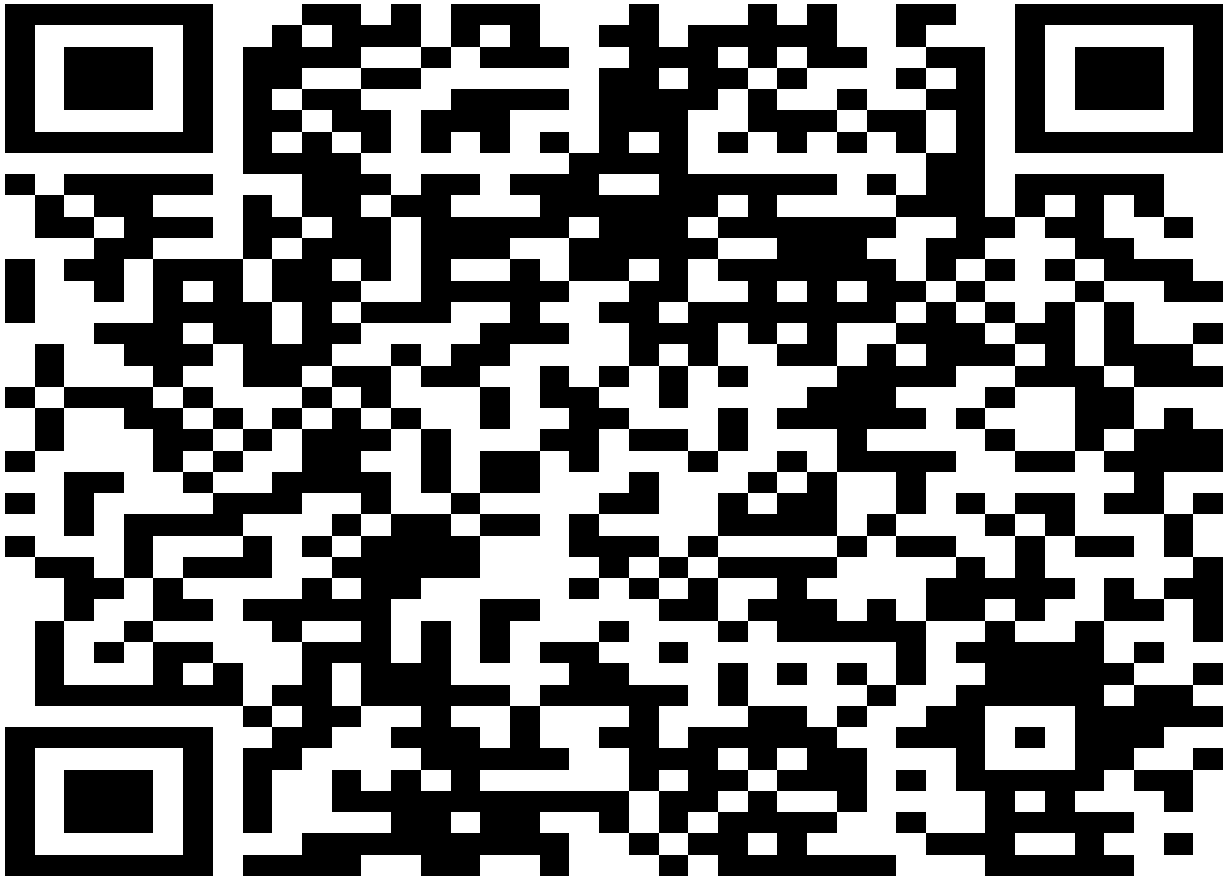







https://www.redbooks.ibm.com/Redbooks.nsf/RedpieceAbstracts/sg248218.html

Because we just have not seen this enough!!! Session # **17046**



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