MQ SMF: Statistics, Accounting, and Lies
Session 28146

Lyn Elkins – WSC – elkinsc@us.ibm.com

Mitch Johnson – WSC – mitchj@us.ibm.com

"There are three kinds of lies: lies, damned lies, and statistics."

Agenda

MQ SMF – the answer or not

My favorite RFEs

Examples of lies, or misleading results

Summary

MQ SMF Data- the Answer

❖ I have spent longer on MQ SMF than anything I have worked on at IBM
   ❖ Well perhaps travel arrangements since they enforce these new tools
❖ MQ SMF has taught me more about the way it works than any session
❖ MQ SMF exposes patterns of use and operation
   ❖ A break in those patterns can be the answer to a problem or an indicator of trouble brewing

Why do I have to know about MQ SMF data?

❖ We have tools to look at this for us
  ❖ Tools can be wrong
    ❖ How many years has MP1B been published, by MQ developers?
      ❖ And within the last 6 months a calculation error was found
  ❖ Tools can exclude data
    ❖ Sometimes it is a matter of catching up
    ❖ Sometimes it is a decision
      ❖ It doesn't evaluate the accounting data
    ❖ Looking across a multiple queue managers can be a problem

❖ Machine Learning is still more at the hype stage than reality
  ❖ Someday <pick your AI name of choice> will be able to do it all

My current favorite RFEs on SMF

❖ Publish the MQ Statistics and MQ Accounting SMF data in addition to creating the SMF Records
  ❖ Like the Statistics and Accounting information is published on distributed
  ❖ https://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&CR_ID=134864

❖ IBM MQ for z/OS - Queue Statistics
  ❖ New subtype of statistics records dealing wiht the API requests against individual queues
  ❖ Much of the time the task detail is less critical than the queue information
  ❖ https://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&CR_ID=135074

Examples of Lies – or at least misleading results

- ➢ Generated messages
- ➢ The last LPAR added
- ➢ There are no problems with my bufferpools
- ➢ Other Lies, omissions, misleading things, etc.

Generated Messages – those pesky things

❖ Generated messages are messages that the queue manager itself writes when:
  1) A trigger event occurs
  2) A Performance event occurs
  3) Any other kind of detected event

❖ They are included in the Data Manager put count
  ❖ The put count from the DM include:
    ❖ All the MQPUT + MQPU1 requests from the MM that are passed to the DM
      ❖ Not passed are the put and put1 that are put to waiting getter
    ❖ All the queue manager generated message
  ❖ Note: I have not studied this is detail, generated trigger messages do show up in this data and while I have been told that performance events and report messages are as well I've not verified.

❖ There is a generated messages count in the WQ records – NGEN field
  ❖ But what does it really count?
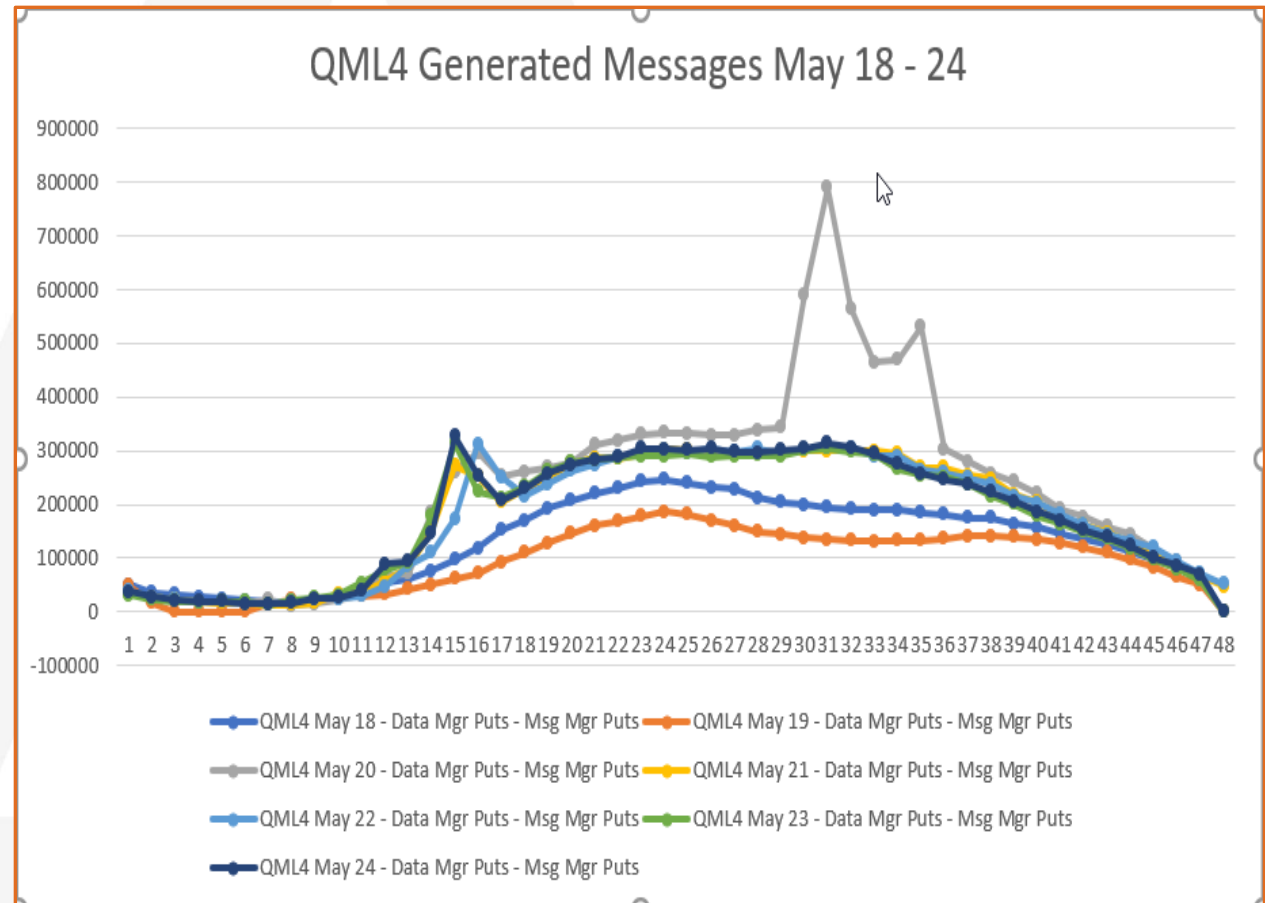
Generated Messages – So where is the LIE?

❖ Generated messages are counted correctly in the WQ records?
- ❖ On z/OS the generated messages count only includes trigger messages
  - ❖ Performance events, etc. are not included
- ❖ On z/OS only trigger events associated with PRIVATE queues are counted
  - ❖ Shared queue accounting does not reflect trigger events
- ❖ On distributed, everything is counted except the trigger messages

Last LPAR added problem – Generated Messages

❖ Customer extended their Queue Sharing group, added a new LPAR
  - ❖ New Queue manager was added to the LPAR and QSG on May 1
  - ❖ New BMP to process messages was added on May 18
  - ❖ On May 20, customer experienced what was perceived as an 'outage'
    - ❖ Response time went from sub-second to multi-second to minutes for some transactions
❖ Opened a PMR against MQ, IMS, and z/OS
  - ❖ No real problems found in any subsystem
  - ❖ Statistics, accounting and dumps failed to show any real issue in the new LPAR
❖ Resolution process included stopping the BMP that was added on May 18
 - ❖ Queue depths began dropping
❖ So adding #8 must be the problem!

Generated Messages – So where is the LIE?

❖ Generated messages as seen from the differences in the MQPUT+MQPUT1 from the message manager and the puts reported in the data manager

QML4 Generated Messages May 18 - 24

Legend:
- QML4 May 18 - Data Mgr Puts - Msg Mgr Puts
- QML4 May 19 - Data Mgr Puts - Msg Mgr Puts
- QML4 May 20 - Data Mgr Puts - Msg Mgr Puts
- QML4 May 21 - Data Mgr Puts - Msg Mgr Puts
- QML4 May 22 - Data Mgr Puts - Msg Mgr Puts
- QML4 May 23 - Data Mgr Puts - Msg Mgr Puts
- QML4 May 24 - Data Mgr Puts - Msg Mgr Puts

Last LPAR added problem – What really happened

❖ The total MQPUTs and MQPUT1 request for the week
  ❖ Anything look familiar?

QML4 Total MQPUTs May 18-24
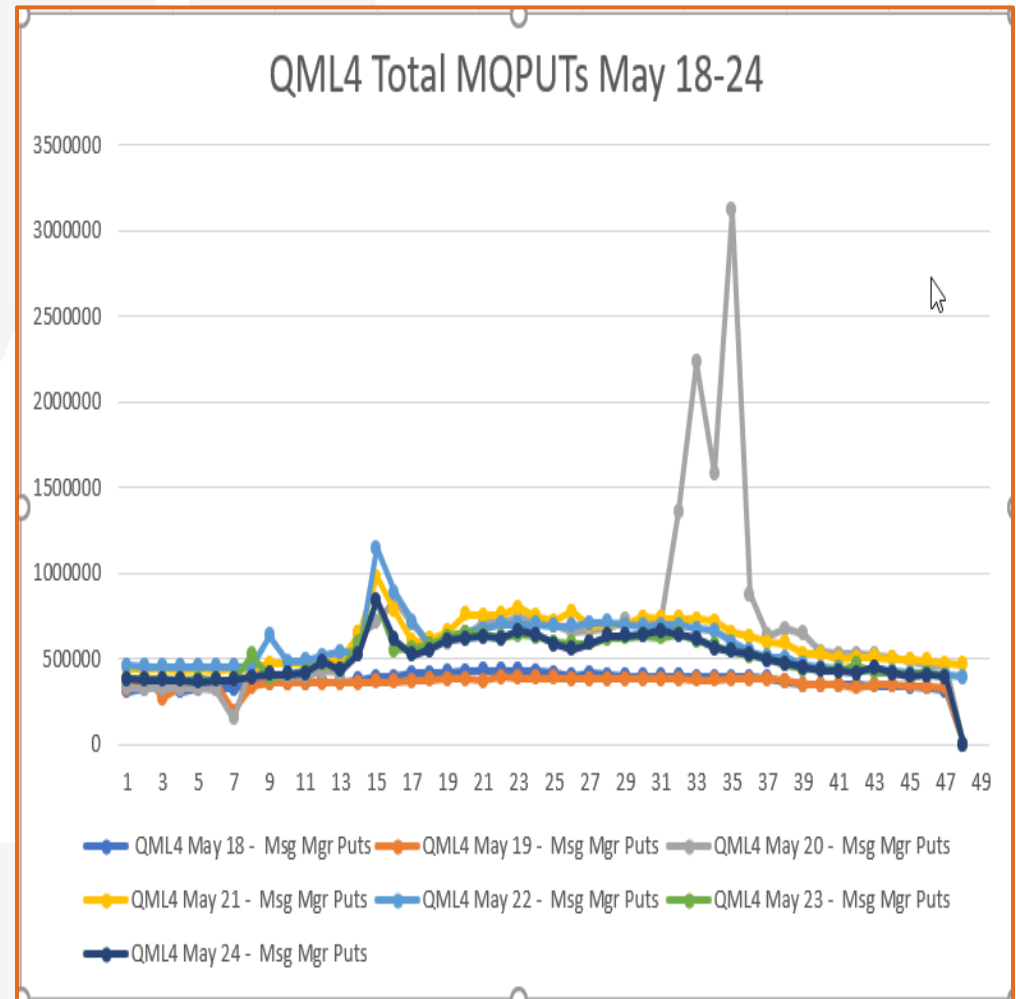
My Buffer pools, there are no problems with my buffer pools!

❖ When above the bar and more buffer pools were introduced, way back in V8, bufferpools problems went away.

  ❖ I've moved the BPs above the bar and made them huge
  ❖ Never seeing any SoS or Sync Write Threshold being hit any more
  ❖ So why am I having slowdowns?

Buffer pool lies

❖ Big bufferpools can cause some issues:
   ❖ Internal contention, especially when there is a mixture of batch and online work, does not show up in the statistics
   ❖ Increasing queue depth causes latches when internal pointers have to be extended or moved
   ❖ Smaller buffer pools may have hidden this because the deferred write threshold was hit more often, causing message offloads
   ❖ If there is I/O going on, unless the buffer pool is page fixed the internal latching can be longer due to the software page fixing going on

BP Lies – the Latching problem

BASE_NAME	PAGESET_ID	BUFFERPOOL_ID	PUT_PAGESET_ACCESS_COUNT	LONGEST_LATCH	MAX_LATCH_WAIT_TIME_S	MAX_LATCH_WAIT_TIME_US	MAX_LATCH_WAIT_ID
ELKINSC.QUEUE1	2	1	62	0000004807	5	731717	19
ELKINSC.QUEUE2	2	1	62	0000004806	3	394963	19
ELKINSC.QUEUE3	2	1	62	0000004807	2	769386	19
ELKINSC.QUEUE1	2	1	63	0000004806	2	701325	19
ELKINSC.QUEUE2	2	1	0	0000004806	2	701325	19
ELKINSC.QUEUE3	2	1	0	0000004806	2	701325	19
ELKINSC.QUEUE4	2	3	0	0000004806	1	701842	19
ELKINSC.QUEUE5	2	3	0	0000004806	1	701842	19
ELKINSC.QUEUE4	2	3	0	0000004806	1	701842	19
ELKINSC.QUEUE5	2	3	0	0000004806	1	701842	19

```
SELECT Q.QMgr, Correlation, Base_Name, Pageset_ID, BufferPool_ID, Get_Pageset_Count,
       Get_Messages_Skipped_Count, Get_Messages_Expired_Count,
       Put_Pageset_Access_Count, Put1_Pageset_Access_Count, Correl,
       Longest_Latch, Max_Latch_Wait_Time_s, Max_Latch_Wait_Time_us, Max_Latch_Wait_ID,
       Start_Time_Date, Start_Time_Time

FROM MQSMF.WQ Q , MQSMF.WTAS WTAS

WHERE
      (Q.QMgr = ''QML1'' AND
       Correlation = Correl AND
       Longest_Latch > ''00000000000000000'' AND
       (Max_Latch_Wait_Time_us > 25000 OR
        Max_Latch_Wait_Time_s >0))
```

Other Lies

- ❖ CSQ4SMFD always presents the truth
  - ❖ Not quite, about six months ago we found a major bug with the presentation of the Log Manager data :
    - ❖ When running a v910 SMF formatter against a v910 queue manager, the fields 'QJSTSLPTU' AND 'QJSTIOSQU' are not included in the SMF dump.

    - ❖ When running a v910 SMF formatter against a v710 queue manager, there are various additional fields that are not being outputted from 'QJSTIOCOUNT' down to 'QJSTIOMAXSUSL'.
  - ❖ New APAR has been added, although the PTF is not yet closed is should be soon.
    - ❖ **PH15885 -** https://www-01.ibm.com/support/docview.wss?uid=swg1PH15885

  - ❖ Also, and older fix on dates from CSQ4SMFD
    - ❖ PI68790: WMQ OUTPUT FROM THE CSQ4SMFD RETURNS VALUES THAT ARE NOT IN THE VALID DATE RANGE

Other Statistics, accounting, omissions, and lies

- ❖ Channel Accounting:
  - ❖ Does not include any CPU use
    - ❖ Some of us are particularly interested in the cost of the MQCONN/CONNX and MQDISC
  - ❖ Is not produced like task accounting
    - ❖ There is a request to change the behavior to act like the task accounting records

Other Statistics, accounting, omissions, and misleading things

❖ Data Manager Reporting – Object creates and Deletes:
  ❖ Object creates are typically low, unless Temporary or Permanent Dynamic queues in heavy use
  ❖ Recent discovery – 2 to 1 ratio of Object Deletes to Object Creates
    ❖ This has been observed before, but we now know the cause
    ❖ When a TD queue is closed it becomes eligible for deletion
      ❖ If there is an explicit syncpoint done, the queue is deleted as part of that work
      ❖ If there is no explicit syncpoint, the delete is attempted but does not complete as the transaction still holds and interest in the TD queue
      ❖ After the transaction fully ends, a scavenger task can delete and clean up the TD queue

Other Statistics, accounting, omissions, and misleading things

❖ Data Manager Reporting – Object creates and Deletes:

  ❖ Another recent discovery – a MUCH higher ratio of Object Deletes to Object Creates

  ❖ Queue manager is counting every delete attempt

  ❖ We suspect that the channels using the TD queues are abending, causing this effect.

  ❖ Asking customer to open Case on this

Jan 10 - Object Creates Vs Object Deletes
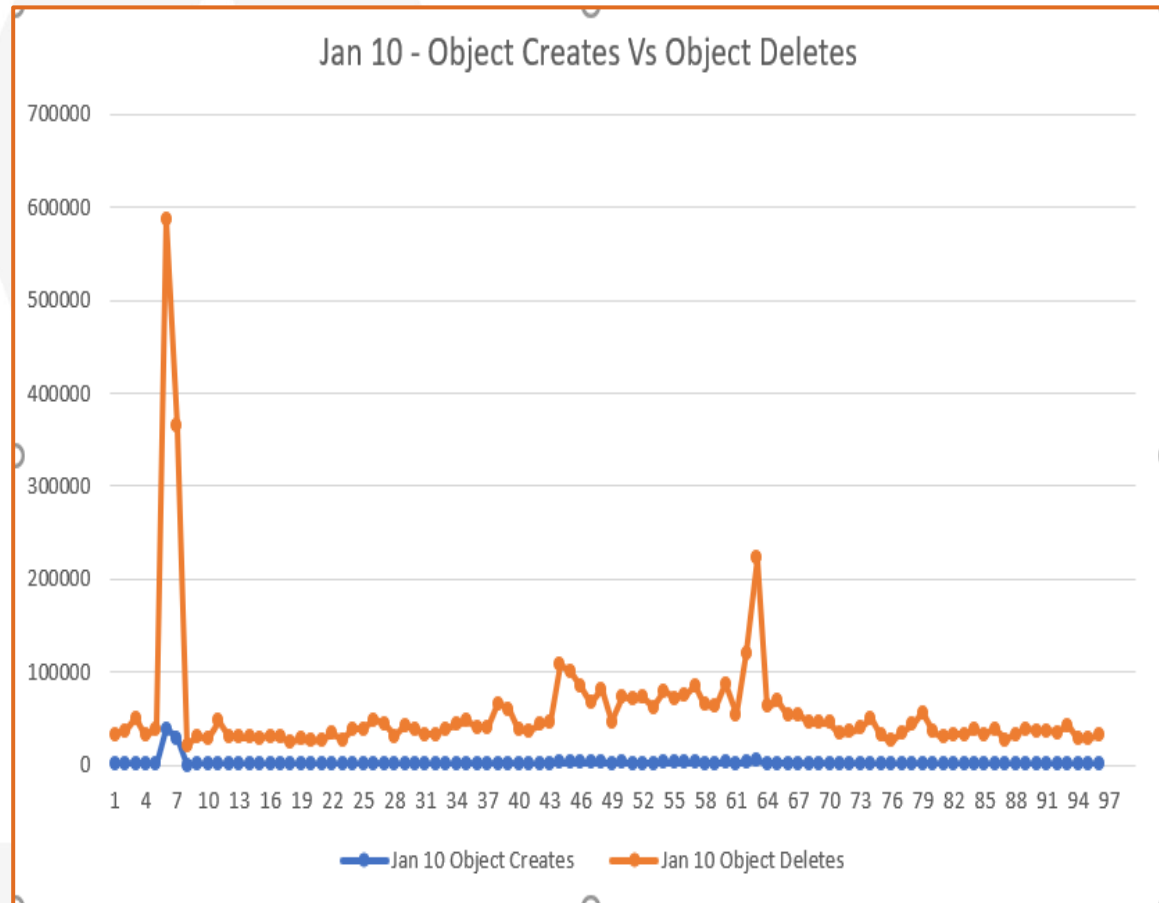
Other Statistics, accounting, omissions, and misleading things

❖ Db2 BLOB Use – Not called out in MP1B reporting
  ❖ Db2 BLOBs may be used for message bodies over 63K that cannot be stored in the CF
  ❖ BLOB use has been discourages since MQ V7.1 when Shared Message Data Sets (SMDS) became available
    ❖ CPU consumption and throughput
  ❖ While BLOBs continue to be allowed, there is talk that this may be deprecated at some point
  ❖ When does 'some' become 'too many' ?

| Date	Blob Inserts	Blob Deletes
1/7/2020	41178	41683
1/8/2020	38584	39817
1/9/2020	39989	40377
1/10/2020	42446	42999
1/11/2020	36439	36994
1/12/2020	43528	44056
1/13/2020	38072	38284

Unfulfilled MQGETs

❖ Message Manager and Data Manager Data
  ❖ MQGETs reported in the MM data is the number of MQGET requests that are issued
  ❖ Message Gets reported in the DM component are the gets passed from the MM to the DM
    ❖ Example: An MQGET against an EMPTY queue is not passed, an MQGET for a non-empty queue is passed but may not be fulfilled because a selector is not matched
  ❖ The difference in the gets is an approximate number of 'unfulfilled gets'
    ❖ It's approximate because there are other factors at play (selectors, etc.)
  ❖ Can be a good leading indicator that there are too many getting applications (or too few!)

Unfulfilled MQGETs
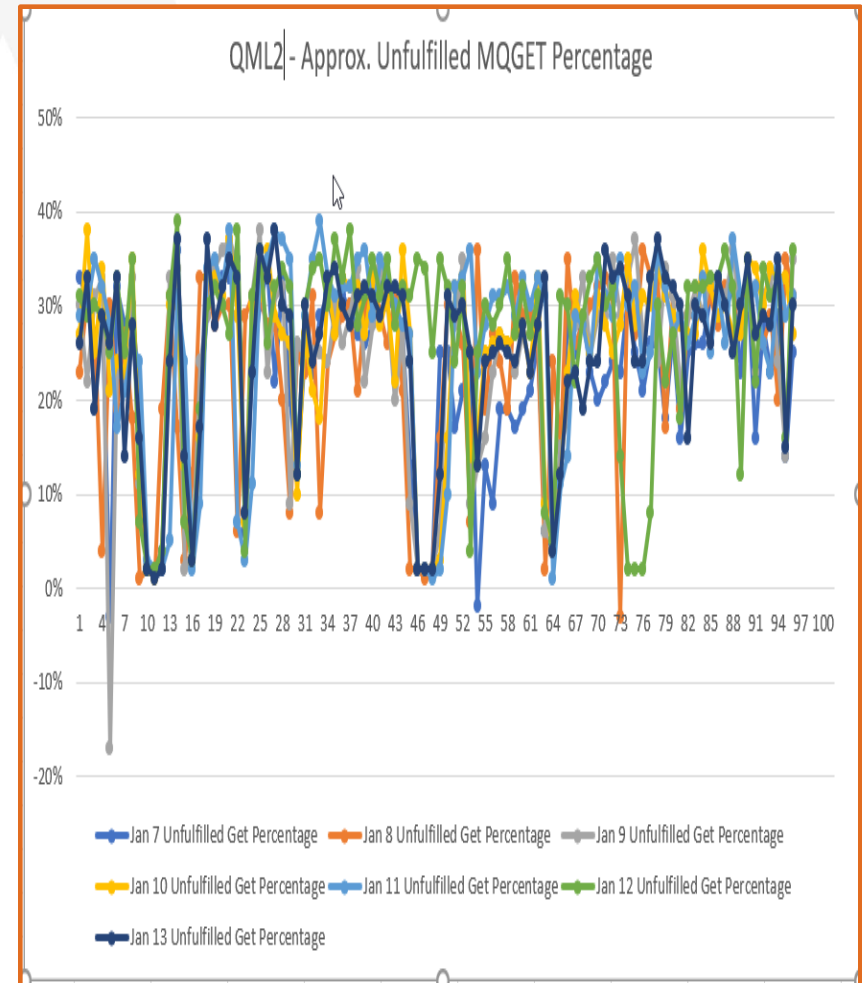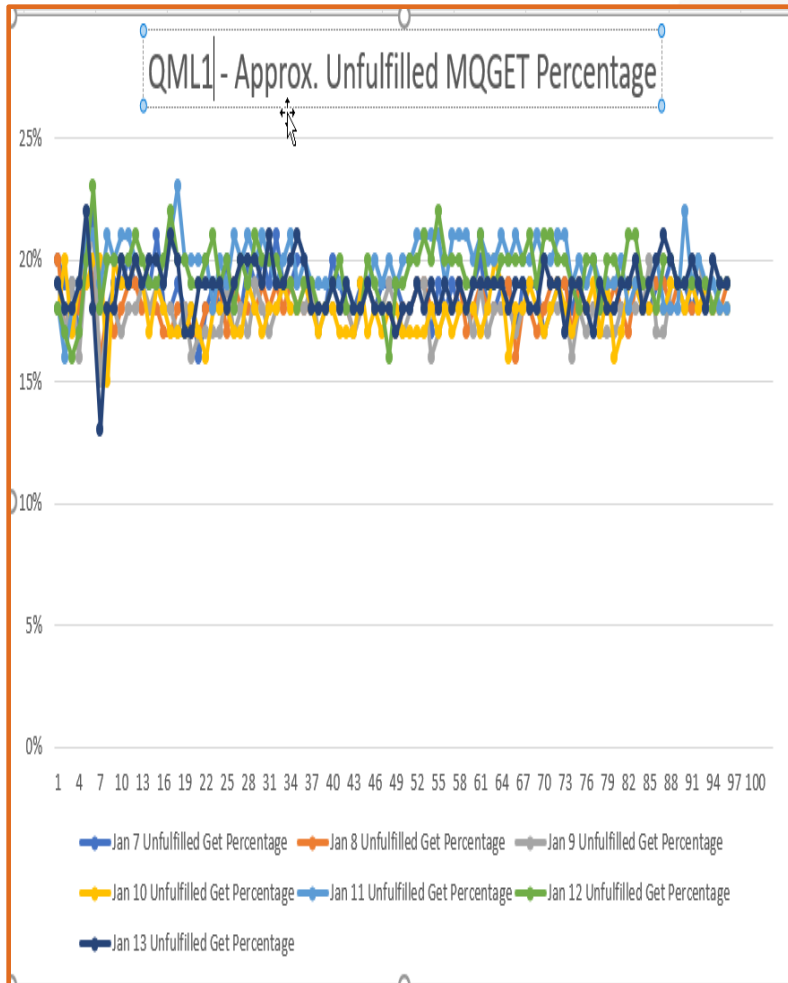
Unfulfilled GETs

QML3 - MQGETs Approximate Unfulfilled Requests
August 05 - 09
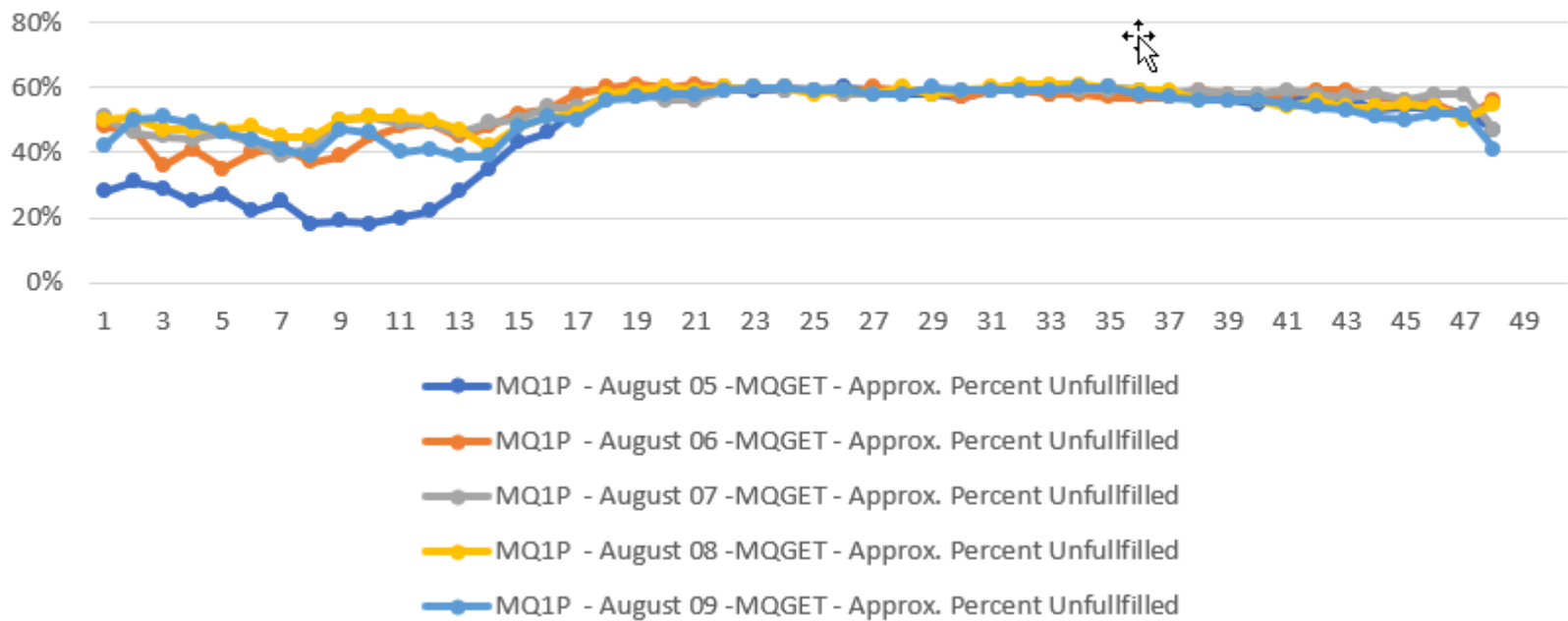
Still my 'favorite' lie – which buffer pool is used

❖ The WQ Accounting records are not always accurate in the buffer pool and page set values
  - ❖ Often they will be the default value of 0, if the storage has not yet been accessed or has not needed to be
    - ❖ For example - Put to waiting getter
  - ❖ Recently learned that when heavy use of TD queues is involved the BP and PSID can be 'leftover' values that do not reflect reality in any way, shape or form
❖ I have an open case on this problem, as it is impacting figuring out how storage is used in heavy TD environments.

Summary

❖ I learn something new with every set of data I look at!

❖ The MQ SMF data is coming under more scrutiny as tools (both IBM and vendor provided) are making more use of the data

   ❖ This has several benefits, the more we find that is missing or inaccurate the better all the tools become

      ❖ Even CSQIBALL

   ❖ This has a downside that people are becoming less familiar with the data

      ❖ As an MQ or performance admin, you need to know what the data is telling you

❖ Finally, everyone is spending a lot of time proving that MQ on z/OS is not a problem

MQ SMF: Statistics, Accounting, and Lies
Session 28146

Lyn Elkins – WSC – elkinsc@us.ibm.com

Mitch Johnson – WSC – mitchj@us.ibm.com