MQPERF1

MQ Statistics and MP1B – More Statistics Reports and New SMF data (V8 & 9)

MP1B – Other statistics reports

- In this presentation the following reports will be discussed:
  - Data Manager
  - Pageset statistics
  - Channel Initiator
    - Adapter Tasks
    - Dispatcher Tasks
    - SSL Tasks
  - Queue Sharing Group Reports
    - CF and CFCSV
    - SMDS
    - DB2

Data Manager Report

- The data manager name is a bit misleading to me – it includes a lot about the queue manager access the queues, and some on the API
- Internally the data manager will call the buffer manager to access to the messages
- A sample report looks like this:

```
Data Manager statistics
MPX1,QML3,2018/11/14,01:13:58,VRM:900,
   Obj Cre        657, Obj Puts          2,  Obj Dels      1314,  Obj Gets       4190
   Locates      38171, Stgclass          0,  Enum           240
   Msg Gets     16049, Msg Puts     18122
   Lock MM          1, Rel MM            1,  Delete MM        0
   Read Ahead:IO    0,:Buffer            0,  Gets:disk        0,        BP       3611
MPX1,QML3,2018/11/14,01:28:56,VRM:900,
   Obj Cre        443, Obj Puts          0,  Obj Dels       886,  Obj Gets       3833
   Locates      23031, Stgclass          0,  Enum           233
   Msg Gets     13038, Msg Puts     14556
   Lock MM          0, Rel MM            0,  Delete MM        0
   Read Ahead:IO    0,:Buffer            0,  Gets:disk        0,        BP       3172
```

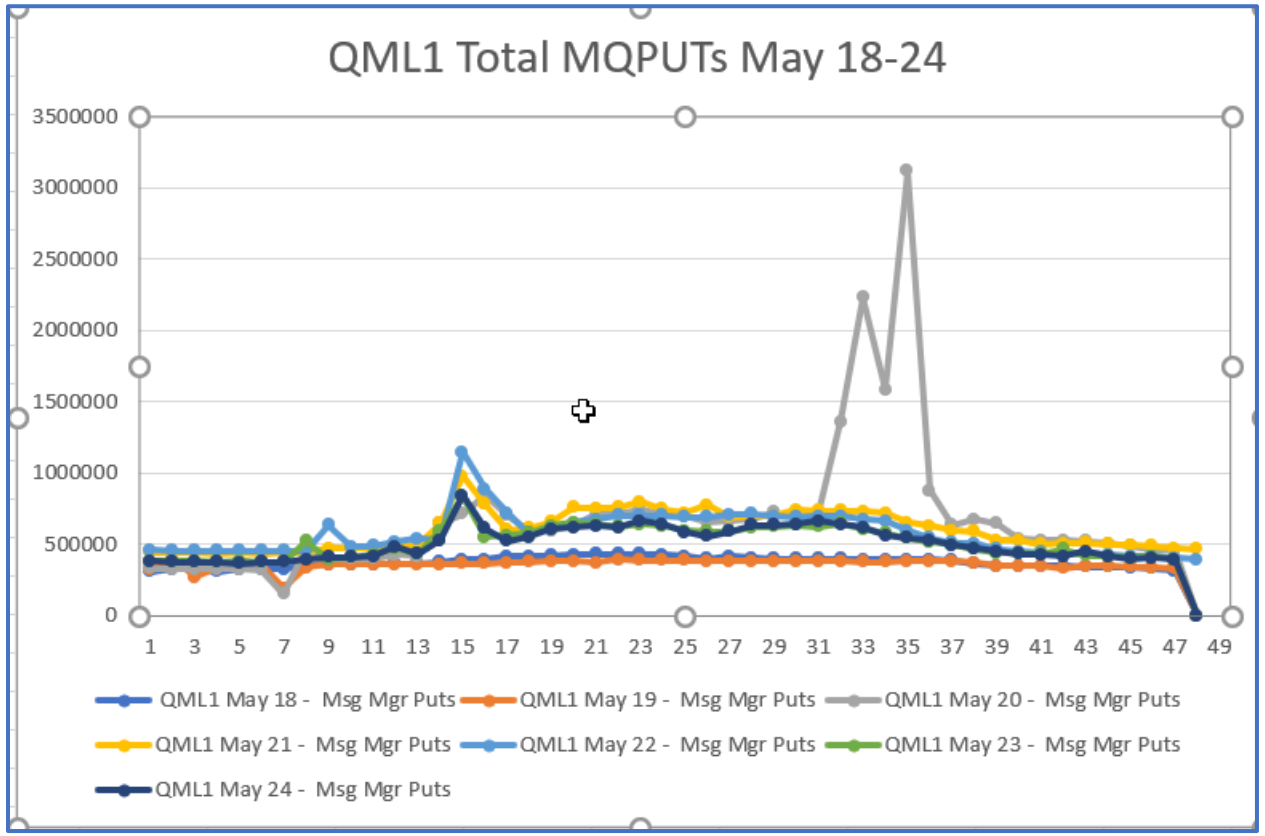
Data Manager Report – fields of interest

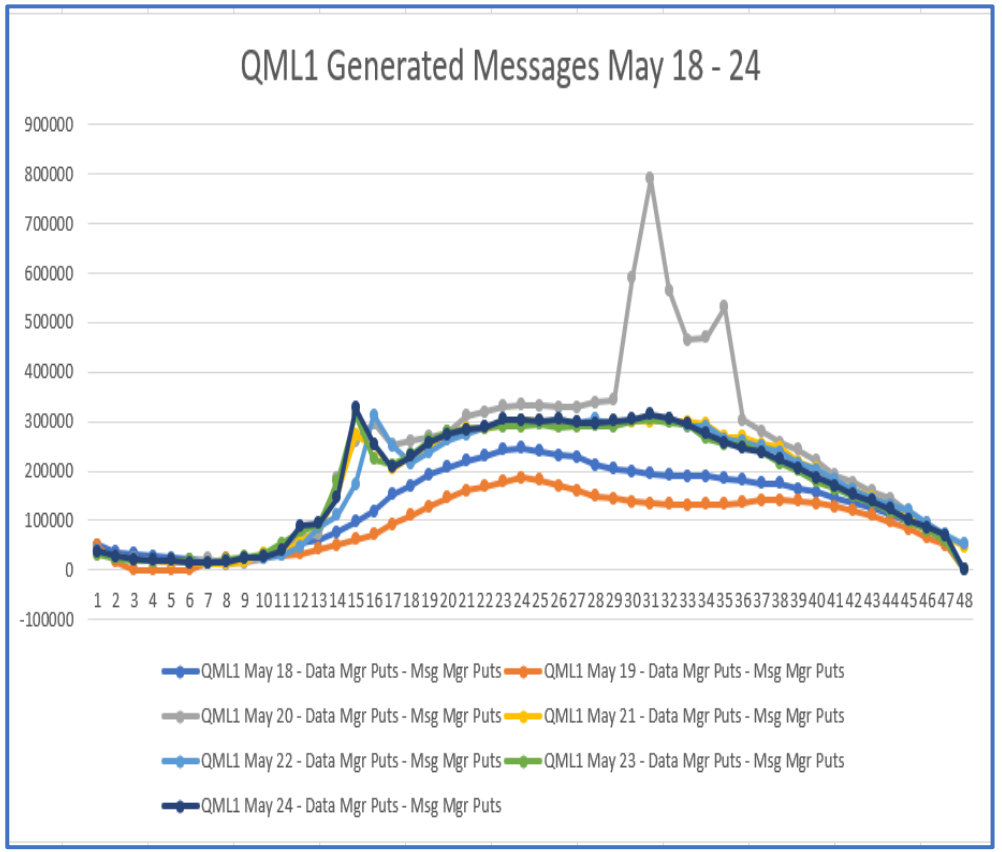
- These reports are from the CSQDQIST macro
  - At this time of composition, the MP1B documentation has an incorrect macro name, CSQDIST
- Obj Cre –objects created during this interval - QISTDCRE field
  - In the sample the numbers seem quite high for a production queue manager
    - This is likely the result of temporary dynamic queues being created, often by monitoring tools.
    - Verify the tasks creating the queue with task accounting records.
    - Reducing this creation activity can result in lowered CPU costs – use permanent queues where possible.
- Obj Puts – objects altered – QISTDPUT field
- Obj Del – objects deleted – QISTDDEL field
  - Again this surprisingly high number is likely due to the use of temporary dynamic queues
  - Note – looking into a possible problem.
- Obj Gets –object definition retrieval for a display or alter – QISTDGET
- Locates – object locates for an MQOPEN, display, or alter – QISTDLOC
- Stgclass – requests to alter a storage class definition – QISTALST
- Enum – requests to find an object - QISTENUM

Data Manager Report – fields of interest - continued

- Msg Gets – The number of MQGET requests – QISTMGET
- Msg Puts – The number of MQPUT+MQPUT1s – QISTMPUT
- Lock MM, Release MM, Delete MM – are counting actions performed against Marked Message when browsing with marking
- Read Ahead I/O – counted when messages are being read from a pageset on sequential processing

MQPUTs and Generated Messages





Pageset Statistics

- Pageset activity is I/O activity, or occasionally the expansion of the pageset.
- Pages written in checkpoint describes I/O activity that is due to log switching or LOGLOAD checkpoints

```
MPX1,QML3,2018/11/14,01:13:58,VRM:900,
  From 2018/11/14,00:59:00.958046 to 2018/11/14,01:13:58.539156, duration   898 seconds.
PS00 BP   0, Pages     89993, Size  351 MB, free    99.8%, used     0.2%, P  0%, NP  0%, #full 0,
PS00 No I/O activity
PS01 BP   1, Pages    332978, Size 1300 MB, free    99.8%, used     0.2%, P  0%, NP  0%, #full 0,
PS01 No I/O activity
PS02 BP   2, Pages    805809, Size 3147 MB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
PS02 No I/O activity
PS03 BP   3, Pages   1042314, Size 4071 MB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
PS03 No I/O activity
PS04 BP   4, Pages     89993, Size  351 MB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
PS04 No I/O activity
PS05 BP   5, Pages     89993, Size  351 MB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
PS05 No I/O activity
PS06 BP   6, Pages    341977, Size 1335 MB, free    92.3%, used     7.7%, P  7%, NP  0%, #full 0,
     Pages written in checkpoint            0
     Pages written not in checkpoint     4672
     Number of stripes                      1
     Put Cursor high                 003F6609
     Expansion type:User
PS06 Type :I/O requests,   Pages, Avg I/O time, pages per I/O, MB/Sec,  busy%
PS06 Write:         292,    4672,         1736,          16.0,     36,     0%
PS06 GET  :           1,       1,          468,           1.0,    8.3,     0%
PS99 BP  99, Pages     53995, Size  210 MB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
PS99 No I/O activity
```

Channel Initiator Statistics – Adapter Tasks

- The Channel Initiator Adapter tasks make the requests to the queue manager
- The default is 8 tasks and the channel initiator uses the first available task to make requests
- More tasks are needed when the last task in the string is busy
- The shortened example shown, has 5 of the 32 allocated tasks
  - In this case the tasks are probably over allocated as the last task used was task 4 and it was never busy

```
MPX1,QML3,2018/11/09,11:11:54,VRM:900,
From 2018/11/09,11:00:47.027983 to 2018/11/09,11:11:54.017670 duration
Task,Type,Requests,Busy %,     CPU used, CPU %,"avg CPU","avg ET"
    ,    ,        ,      ,      Seconds,      , uSeconds,uSeconds
   0,ADAP,   196468,   3.7,     9.402554,  1.4,        48,      125
   1,ADAP,     5302,   0.1,     0.257137,  0.0,        48,      118
   2,ADAP,      114,   0.0,     0.005127,  0.0,        45,      116
   3,ADAP,       15,   0.0,     0.000624,  0.0,        42,      376
   4,ADAP,        5,   0.0,     0.000245,  0.0,        49,       44
Summ,ADAP,   201906,   0.1,     9.665891,  0.0,        48,      124
```

Channel Initiator Statistics – Dispatcher Tasks

- The dispatcher tasks communicate with the network to send and receive data
- As channels start they are bound to a particular task, and will remain with that task until the channel is stopped and restarted.
- In the sample show, task three is busier than the others and has more channels assigned.

```
MPX1,QML3,2018/11/07,00:13:30,VRM:900,
From 2018/11/06,23:58:32.432441 to 2018/11/07,00:13:30.013562 duration
Task,Type,Requests,Busy %,     CPU used, CPU %,"avg CPU","avg ET"
    ,    ,        ,      ,      Seconds,      , uSeconds,uSeconds
   0,DISP,     8399,    0.0,     0.136885,  0.0,        16,       14
   1,DISP,     2272,    0.0,     0.034978,  0.0,        15,       13
   2,DISP,    73714,    0.1,     0.930957,  0.1,        13,       11
   3,DISP,   106318,    0.2,     1.584851,  0.2,        15,       13
   4,DISP,    34828,    0.0,     0.408012,  0.0,        12,       10
   5,DISP,      162,    0.0,     0.017988,  0.0,       111,       98
Summ,DISP,  225693,    0.0,     3.113672,  0.0,        14,       12
   0,DISP, number of channels on this TCB,    6
   1,DISP, number of channels on this TCB,    5
   2,DISP, number of channels on this TCB,    6
   3,DISP, number of channels on this TCB,    9
   4,DISP, number of channels on this TCB,    6
Summ,DISP, number of channels on all TCBs,   33
```

Channel Initiator Statistics – SSL Tasks

- The SSL tasks behave much like the dispatcher tasks
- If crypto coprocessors are used and the elapsed time rises, it can mean that the coprocessors are overloaded.
- Monitor the elapsed time – as that is one area that can be influenced by the addition of resources

```
MPX1,QML3,2018/11/07,00:13:30,VRM:900,
From 2018/11/06,23:58:32.432441 to 2018/11/07,00:13:30.013562 duration 897.581120 seconds
Task,Type,Requests,Busy %,     CPU used, CPU %,"avg CPU","avg ET",longest ,date      ,time
   ,   ,        ,      ,      Seconds,      , uSeconds,uSeconds,uSeconds,           ,
   0,SSL ,    5324,   0.0,    0.099386,  0.0,        19,       19,     214,2018/11/07,00:02:55.037480
   1,SSL ,     447,   0.0,    0.006556,  0.0,        15,       14,      62,2018/11/07,00:07:04.459401
   2,SSL ,   40827,   0.1,    1.050829,  0.1,        26,       26,    2413,2018/11/07,00:02:02.937263
   3,SSL ,   52076,   0.1,    0.636646,  0.1,        12,       12,    1011,2018/11/07,00:05:35.853284
   4,SSL ,      71,   0.0,    0.001193,  0.0,        17,       17,     103,2018/11/07,00:04:53.894447
Summ,SSL ,   98745,   0.0,    1.794610,  0.0,        18,       18,    2413,2018/11/07,00:02:02.937263
```

CF Statistics

- Typically we (the WSC) use the CF Activity report, rather than this
  - Potential slowdowns
  - Service time averages from different LPARs
  - Requests from all LPARs
- The fields
  - Structure Full – if the structure is full this count is incremented
  - Max Entries and Max Elements – the highest number of entries and elements during the interval
    - As both are 256 bytes, this might be used to calculate how much of the storage area is used by MQ for messages
  - Single Requests – typically an MQ GET or PUT
  - Multiple Requests – typically a commit
  - Average elapsed time for both
  - Retries for both

QMGR Health – CF Activity

```
STRUCTURE NAME = QSGBUSER            TYPE = LIST    STATUS = ACTIVE
          # REQ    -------------- REQUESTS --------------   ---------------- DE
SYSTEM    TOTAL                 #      % OF  -SERV TIME(MIC)-    REASON      #    %
NAME      AVG/SEC               REQ     ALL      AVG    STD_DEV               REQ    R

MPX1        295K    SYNC      295K    26.9      4.3        1.2     NO SCH      0    0
          492.1     ASYNC        0     0.0      0.0        0.0     PR WT       0    0
                    CHNGD        0     0.0  INCLUDED IN ASYNC     PR CMP      0    0
                    SUPPR        0     0.0                        DUMP        0    0

MPX2        802K    SYNC      802K    73.1     17.8        2.5     NO SCH      0    0
           1339     ASYNC        0     0.0      0.0        0.0     PR WT       0    0
                    CHNGD        0     0.0  INCLUDED IN ASYNC     PR CMP      0    0
                    SUPPR        0     0.0                        DUMP        0    0
```

- We (the WSC) tend to use the CF Activity report rather than the MQ Statistics
- In the example shown above it is easy to see that the MPX2 LPAR is getting a much higher service time (almost times!) than the MPX1 LPAR and that MPX2 is making many more requests.
  - While this may not be a problem, this could indicate some workload skewing that is impacting response times, etc.

CF Statistics – Sample Report

```
MPX1,QML3,2018/11/07,00:28:27,VRM:900,
  CSQ_ADMIN    , Structure-fulls       0
    Single      2936, avg et in uS     11, Retries        0
    Multiple     I  5, avg et in uS    429, Retries        0
    Max entries       1131,  Max elements       1356
  APPPER       , Structure-fulls       0
    Multiple       3, avg et in uS      9, Retries        0
  APPNOP       , Structure-fulls       0
    Single      4719, avg et in uS     18, Retries        0
    Multiple    1595, avg et in uS     19, Retries       25
    Max entries      13638,  Max elements      71272
  CSQSYSAPPL   , Structure-fulls       0
    Multiple     179, avg et in uS     10, Retries        0
```

Data Manager Report

SMDS – Shared Message Data Set Statistics

- In the report format there is a huge amount of detail
  - Much of the pertinent data is in the CSV file
- At this point I have not seen many queue managers require tuning for SMDS, but I do expect this to change
  - A recent wave of customers moving client applications on z/OS specifically for availability is ‘my’ leading indicator
- Heavy use or I/O slowdowns may cause delays, leading to tuning opportunities for the buffers

SMDS – Usage CSV report

- In this report, the messages were put on queue manager QML1 and read from QML3
- Note there was no usage reported from QML3
- The Reads reported are from the deletion of message by QML1 after they had been read and committed from QML3

```
z/OS,QM,Date,Time,Duration,StrNum,StrName,What,requests,Pages,Pages/Req,IOTime,WaitTime,IORate,IOBusy,
MPX1,QML1,2018/12/14,13:28:34,73,  2,SMDSMSGS    ,Format,0,0,   0.0,0.000000,0.000000,  0,0,
MPX1,QML1,2018/12/14,13:28:34,73,  2,SMDSMSGS    ,Write ,448,8150,  18.2,0.000920,0.000731, 77,0,
MPX1,QML1,2018/12/14,13:28:34,73,  2,SMDSMSGS    ,Read  ,0,0,   0.0,0.000000,0.000000,  0,0,
MPX1,QML1,2018/12/14,13:28:34,73,  2,SMDSMSGS    ,Other ,0,0,   0.0,0.000000,0.000000,  0,0,
MPX1,QML1,2018/12/14,13:28:34,73,  2,SMDSMSGS    ,Total ,448,8150,  18.2,0.000920,0.000731, 77,0,
MPX1,QML1,2018/12/14,13:30:36,76,  2,SMDSMSGS    ,Format,0,0,   0.0,0.000000,0.000000,  0,0,
MPX1,QML1,2018/12/14,13:30:36,76,  2,SMDSMSGS    ,Write ,0,0,   0.0,0.000000,0.000000,  0,0,
MPX1,QML1,2018/12/14,13:30:36,76,  2,SMDSMSGS    ,Read  ,315,5725,  18.2,0.000935,0.000695, 76,0,
MPX1,QML1,2018/12/14,13:30:36,76,  2,SMDSMSGS    ,Other ,0,0,   0.0,0.000000,0.000000,  0,0,
MPX1,QML1,2018/12/14,13:30:36,76,  2,SMDSMSGS    ,Total ,315,5725,  18.2,0.000935,0.000695, 76,0,
MPX1,QML1,2018/12/14,13:31:36,61,  2,SMDSMSGS    ,Format,0,0,   0.0,0.000000,0.000000,  0,0,
MPX1,QML1,2018/12/14,13:31:36,61,  2,SMDSMSGS    ,Write ,0,0,   0.0,0.000000,0.000000,  0,0,
MPX1,QML1,2018/12/14,13:31:36,61,  2,SMDSMSGS    ,Read  ,97,1775,  18.3,0.000993,0.000768, 72,0,
MPX1,QML1,2018/12/14,13:31:36,61,  2,SMDSMSGS    ,Other ,0,0,   0.0,0.000000,0.000000,  0,0,
MPX1,QML1,2018/12/14,13:31:36,61,  2,SMDSMSGS    ,Total ,97,1775,  18.3,0.000993,0.000768, 72,0,
```

Db2 Statistics

- Db2 statistics should be reviewed periodically, but primarily to make sure that Db2 Blobs are not being overused (hopefully not at all!)

```
MPX1,QML3,2018/12/14,17:29:37,VRM:900,
 Tasks  : Servers    8, Active    9, Conns       0, Discs       0
          HighMax    1, Abend       0,  Requeue       0
               Count  Task avg   Task max   DB2 avg  DB2 max(ms)  (Task-DB2)Avg     Max
 List       :     12          0          0          0          0              0       0
 SCS Select :      1          0          0          0          0              0       0
 Blob Insert:    326          8        251          8        251              0       0
MPX1,QML1,2018/12/14,17:29:49,VRM:900,
 Tasks  : Servers    8, Active    9, Conns       0, Discs       0
          HighMax    1, Abend       0,  Requeue       0
               Count  Task avg   Task max   DB2 avg  DB2 max(ms)  (Task-DB2)Avg     Max
 List       :     12          0          0          0          0              0       0
 SCS Select :      1          0          0          0          0              0       0
 Blob Select:    326          0         11          0         11              0       0
 Blob Delete:    326          1          5          1          5              0       0
```

And there are many more….

- But some are IBM use only
- Others you may have few opportunities to influence queue manager behavior