IBM MQ for z/OS - Things to look for when performing an MQ Health Check

John Waldron
IBM

November 2018
Session JL

Agenda

1. Introduction to health-checking MQ
2. MQ JES Message log
3. Display commands
4. SMF statistics and accounting
   - Capture
   - Formatting
   - Statistics (SMF 115)
   - Accounting (SMF 116)

GUIDE SHARE EUROPE UK REGION

Introduction to health-checking MQ

What is a Health Check?

Health Check is a broad term, with different meanings for different audiences.

A reasonable definition for our purposes:

- Evaluation of a working MQ system to assess whether it is:
  - Processing current work efficiently
  - Could handle changes in workload
    - Changing application requirements
    - Outages

Why perform a Health Check?

A Health Check allows you to understand the characteristics of your **queue managers** and their **workload**:

1. Identify areas of inefficiency
   - Maximize processing speed
   - Reduce CPU consumption

2. Anticipate the effect of changes in workload
   - Put in place mitigations
   - Avoid problems

3. Better equipped when a problem occurs
   - Familiarity with tools to diagnose the problem
   - Understanding of what is different from normal
   - Faster resolution

What to look at?

Applications

- MQ API calls
- Message types and sizes
- Scalability

Queues

- Depth – minimum, maximum, typical
- Message age
- Indexing

Message storage

- Bufferpools and pagesets
- CF structures

What to look at (continued)?

Channels

- Throughput
- Batching
- Underlying network

Logging

- I/O rates
- Checkpointing

Where to look?

MQ has many methods for reporting health metrics, from high level overviews to deeply detailed performance data

- MQ JES message log
  - Informational messages reporting aspects of normal processing
  - Warning or error messages when issues are detected
- Display Status commands
  - Fundamental metrics for specific MQ objects (queues, channels, etc.)
- Monitoring data
  - Enhanced metrics for objects

Where to look (continued)?

MQ has many methods for reporting health metrics from high level overviews to deeply detailed performance data

- SMF Statistics
  - Detailed data for key components of MQ
- SMF Accounting
  - Very detailed activity data for individual applications and channels
- Other z/OS components used by MQ
  - DASD, CF, TCP/IP

MQ JES Message log

Health indicators from MQ JES messages

Warning and error messages:

- Only seen when a problem is detected
- Good for identifying and fixing issues

Informational messages:

- Often overlooked
- Give a high-level view of the health of the queue manager
- Can be used to show trends in processing

MSTR message examples

Logging

```
CSQJ002I MQCA END OF ACTIVE LOG DATA SET
DSNAME=MQTST.SUBSYS.MQCA.LOGCOPY1.DS004, STARTRBA=0000000024EA0000
ENDRBA=00000000275FFFFF
```

Checkpointing

```
CSQP018I MQCA CSQPBCKW CHECKPOINT STARTED FOR ALL BUFFER POOLS
```

```
CSQP019I MQCA CSQPDWP2 CHECKPOINT COMPLETED FOR BUFFER
POOL 1, 52 PAGES WRITTEN
```

Storage usage

```
CSQY220I MQCA CSQSCTL Queue manager storage usage:
local storage: used 141MB, free 1340MB: above bar: used 241MB, free
1GB
```

Queue indexing

```
CSQI004I +MQ09 CSQIMGE3 Consider indexing APP.QUEUE1
by MSGID for BATCH connection MSGGET1, 1421 messages skipped
```

CHIN message examples

Queue manager to queue manager channels

```
CSQX500I +MQ09 CSQXRESP Channel CLUSB.TO.MQ09 started
connection 192.168.0.2
```

```
CSQX501I +MQ09 CSQXRESP Channel CLUSB.TO.MQ09 no longer active
connection winmvs3c (192.168.0.2)
```

SVRCONN channels

```
CSQX511I +MQ09 CSQXRESP Channel SYSTEM.ADMIN.SVRCONN started
connection 192.168.0.55
```

```
CSQX512I +MQ09 CSQXRESP Channel SYSTEM.ADMIN.SVRCONN no longer active
```

Storage usage

```
CSQX004I +MQ09 CSQXSPRM Channel initiator is using 23 MB of local
storage, 1554 MB are free
```

Display Commands

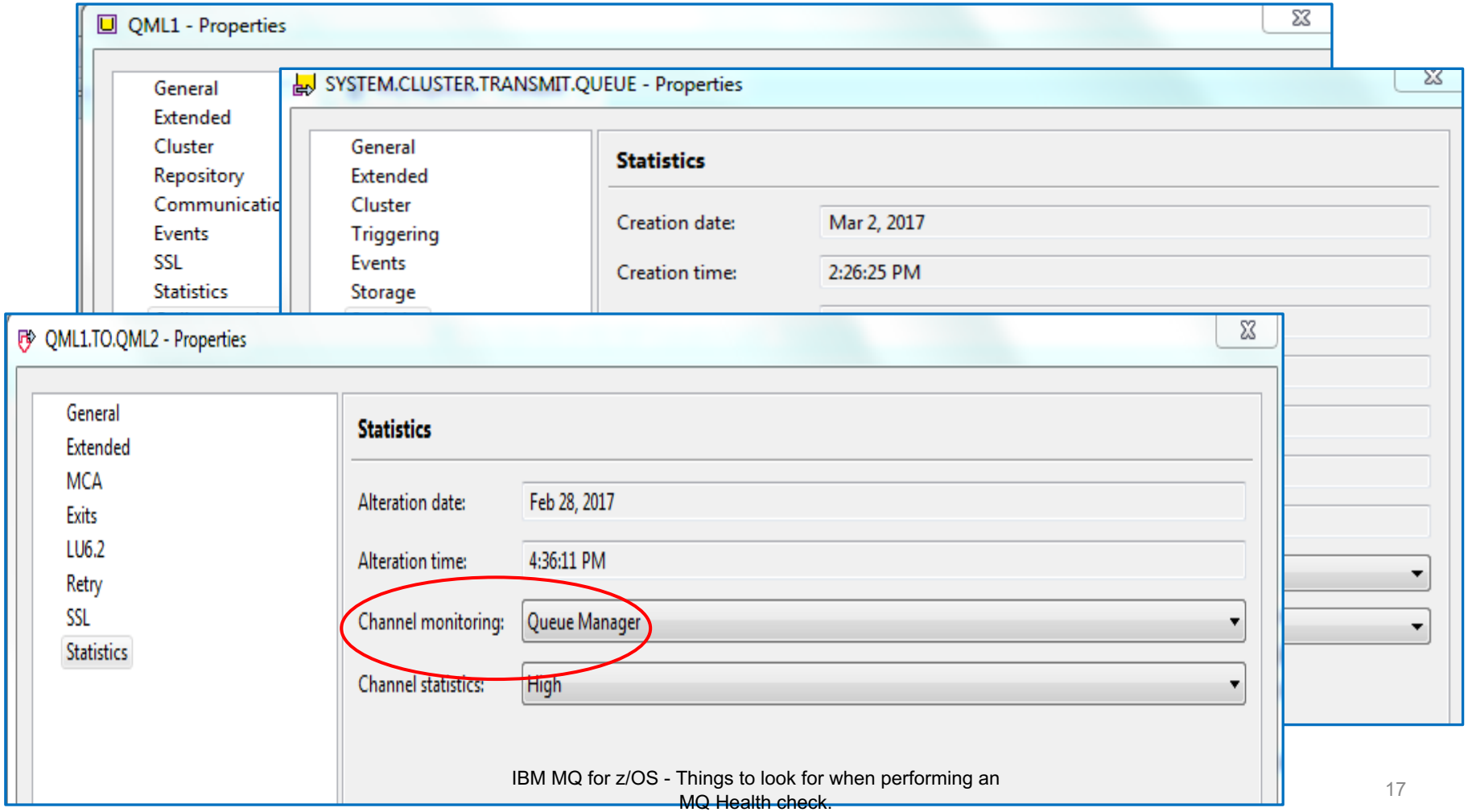
Display status standard values and monitoring values

DISPLAY STATUS command for various MQ object types

- Some values always available
  - QSTATUS
    - CURDEPTH, IPPROCS, OPPROCS
  - CHSTATUS
    - BATCHES, BYTSSENT, BYTSRCVD, MSGS, BUFSSENT, BUFSRCVD
- Additional values require monitoring to be explicitly enabled for the object
  - QSTATUS
    - QTIME, MSGAGE, LPUTDATE, LPUTTIME, LGETDATE, LGETTIME
  - CHSTATUS
    - XQTIME, XQMSGA, NETTIME, EXITTIME, XBATCHSZ, COMPTIME, COMPRATE

Enabling monitoring

- Queue manager and queue/channel settings are important



Display *qstatus* example

```
+MQ09 DISPLAY QSTATUS(QUEUE1) CURDEPTH IPPROCS OPPROCS MONITOR
CSQM293I +MQ09 CSQMDRTC 1 QSTATUS FOUND MATCHING REQUEST CRITERIA
CSQM201I +MQ09 CSQMDRTC  DISPLAY QSTATUS DETAILS
QSTATUS(QUEUE1)
TYPE(QUEUE)
OPPROCS(0)
IPPROCS(0)
CURDEPTH(796)
MONQ(HIGH)
QTIME(20948775,20953090)
MSGAGE(271)
LPUTDATE(2018-05-14)
LPUTTIME(13.05.48)
LGETDATE(2018-05-14)
LGETTIME(13.00.43)
QSGDISP(QMGR)
 END QSTATUS DETAILS
CSQ9022I +MQ09 CSQMDRTC ' DISPLAY QSTATUS' NORMAL COMPLETION
```

Display *chstatus* example

```
+MQ09 DISPLAY CHSTATUS(CLUSB.TO.MQ08) CURRENT BATCHES BYTSSENT
 BYTSRCVD MSGS BUFSSENT BUFSRCVD MONITOR
CSQM293I +MQ09 CSQMDRTC 1 CHSTATUS FOUND MATCHING REQUEST CRITERIA
CSQM201I +MQ09 CSQMDRTC  DISPLAY CHSTATUS DETAILS
CHSTATUS(CLUSB.TO.MQ08)
CHLDISP(PRIVATE)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
CONNAME(9.20.5.158)
CURRENT
CHLTYPE(CLUSSDR)
STATUS(RUNNING)
SUBSTATE(MQGET)
MSGS(5054)
BYTSSENT(2486892)
BYTSRCVD(3152)
BATCHES(103)
BUFSSENT(5057)
BUFSRCVD(104)
MONCHL(HIGH)
XQTIME(17867,602949)
XQMSGSA(0)
NETTIME(709,423)
EXITTIME(0,0)
XBATCHSZ(44,48)
COMPTIME(0,0)
COMPRATE(0,0)
STOPREQ(NO)
RQMNAME(MQ08)
 END CHSTATUS DETAILS
CSQ9022I +MQ09 CSQMDRTC ' DISPLAY CHSTATUS' NORMAL COMPLETION
```

GUIDE SHARE EUROPE UK REGION

SMF statistics and accounting Capture

Setting up for Capture

GUIDE SHARE EUROPE UK REGION

CSQ4ZPRM

- SMFSTAT=NO – Default, (ARRGGGHHH!) should be changed to SMFSTAT=(01,02,03,04) or SMFSTAT=(*)
  - ➢Gathering and producing the statistics is not expensive
  - ➢Most are always gathered, just written when the interval expires

    - *NEW Knowledge*
      - *Using the asterisk does not include the class 4 data.*
      - *AND if you have SMFSTAT=NO, you must turn on all the classes (except 4) then turn on the class 4 data collection independently.*

- SMFACCT=NO – Default, normally controlled via commands
- STATIME – the interval, in minutes, between the creation of the SMF statistical and long running task accounting records
  - ➢30 – default, every 30 minutes
  - ➢0 – Use the system wide SMF interval, usually preferred
  - ➢Any other integer up to 1440
    -Once a day

Setting up for Capture (continued)

Controlling SMF via commands

- *START TRACE(S) CLASS(*)*
  - Starts the statistics production for the queue manager and channel initiator
  - Note that if you have never produced this data, the first record should be ignored. It will have data from when the queue manager started.
- *START TRACE(A) CLASS(*)*
  - Starts the task and channel accounting capture and production
    - Note that tasks that cross interval boundaries will cut a set of accounting records per interval reflecting the activity for that interval.
- *SET SYSTEM STATIME (interval)*
  - The interval is in minutes
  - Change takes effect at the end of the current interval
    - So if you've been silly and set it to a full day (1440), it will be a day before this takes effect
  - Often used to shorten the interval when trying to isolate a performance problem.

SMF statistics and accounting formatting

Processing MQ SMF data – CSQ4SMFD

CSQ4SMFD

- Provided with MQ
- Dump format of the data
- Largely useless for analysis

```
                                                                      CSR
********************************** TOP OF DATA ***********************************
 message manager statistics data
--Q-M-S-T---H-E-X---P-R-I-N-T----
Address  = 2072AC08
00000000 : D40F0048 D8D4E2E3 000024FE 00002402 <M...QMST........>
00000010 : 0000EB1A 0000B480 00000000 00000C48 <................>
00000020 : 00000000 00000000 00000000 00000000 <................>
00000030 : 00000000 00000000 000000B4 00000438 <................>
00000040 : 00000000 00000000                   <........        >
--Q-M-S-T---F-O-R-M-A-T-T-E-D----
qmstid   = d40f
qmstll   = 0072
qmsteyec = QMST
qmstopen = 00009470
qmstclos = 00009218
qmstget  = 00060186
qmstput  = 00046208
```

Processing MQ SMF data – MP1B

MP1B – MQ SMF report formatter

https://www-01.ibm.com/support/docview.wss?uid=swg24005907

- Message Manager, MSGM output file, report sample:

```
Message Manager
MPX2,QML2,2017/01/12,08:50:54,VRM:800,
  From 2017/01/12,08:21:04.455699 to 2017/01/12,08:50:54.787219, duration  1790 seconds.
   MQOPENs       9470,  MQCLOSEs       9218,  MQGETs      60186,  MQPUT        46208
   QMLUT1s          0,  MQINQs         3144,  MQSETs          0,  C ALL H          0
   MQSUBs           0,  MQSUBRQs          0,  MQCBs         180
   MQCTLs        1080,  MQSTATs           0,  Publish         0
   MQGet rate 33.000000/sec  QMLut rate 25.000000/sec
MPX1,QML1,2017/01/12,08:53:53,VRM:800,
  From 2017/01/12,08:24:00.111232 to 2017/01/12,08:53:53.553654, duration  1793 seconds.
   MQOPENs      27170,  MQCLOSEs      24670,  MQGETs     325273,  MQPUT       291386
   QMLUT1s          0,  MQINQs         1043,  MQSETs          0,  C ALL H          0
   MQSUBs           0,  MQSUBRQs          0,  MQCBs        2958
   MQCTLs        8933,  MQSTATs           0,  Publish         0
   MQGet rate 181.000000/sec  QMLut rate 162.000000/sec
```

Processing MQ SMF data – MP1B (continued)

MP1B – MQ SMF report formatter

- Message Manager CSV, **MSGMCSV output file**, sample:

```
MVS,QM,Date,Time,Puts,Put1s,Gets,Open,Close,Inquire,Set,"Close all H",Sub,SubR,"Reg CB",Control,Stat,Publish,
MPX1,QML2,2017/01/12,08:50:54,     46208,         0,     60186,      9470,      9218,      3144,         0,         0,         0,
      0,       180,      1080,         0,         0,
MPX1,QML1,2017/01/12,08:53:53,    291386,         0,    325273,     27170,     24670,      1043,         0,         0,         0,
      0,      2958,      8933,         0,         0,
MPX1,QML3,2017/01/12,08:54:25,     40824,         0,     44904,      1735,      1700,       232,         0,         0,         0,
      0,        57,       440,         0,         0,
MPX1,QML4,2017/01/12,08:55:10,     44112,         0,     48907,      2008,      1910,       337,         0,         0,         0,
      0,        57,       427,         0,         0,
```

Processing MQ SMF data – MP1B (continued)

MP1B produces messages to warn of potential problems

MP1B Message	LPAR	QMGR	Date	Time	MQ Versio	Message
MQQJST11W	MPX1	QML1	2/1/2018	13:30:00	VRM:900	logging rate is low 0 < 50 MB/Sec
MQQJST01W	MPX1	QML2	1/29/2018	0:30:00	VRM:900	Log read log buffers from active logs 157 > 0
MQQPST02S	MPX2	QML2	2/1/2018	16:15:00	VRM:900	BP 2 Filled many(275) times. This is typical of long lived messages. Buf
MQQJST13S	MPX1	QML1	2/1/2018	13:45:00	VRM:900	Log Long delay after I/O completed 2:=10132 uSec
MQQJST11W	MPX1	QML1	2/1/2018	14:00:00	VRM:900	logging rate is low 0 < 50 MB/Sec
MQQJST11W	MPX1	QML1	2/1/2018	14:15:00	VRM:900	logging rate is low 0 < 50 MB/Sec

Processing MQ SMF data – mq-smf-csv

mq-smf-csv is an open source tool for converting MQ SMF records into CSV format

- Available as a GitHub repository
  - **http://github.com/ibm-messaging/mq-smf-csv**
- Pre-built executables provided for Windows, Linux and AIX
- Produces CSV files corresponding to all SMF 115 and 116 data components
  - Raw values for all fields (no analysis calculations)
  - Fields converted into sensible data types
- Output CSV files can be imported into other applications for further analysis
  - Spreadsheets
  - DB2 (mq-smf-csv can provide DDL files to assist with this)
- New: output can now be produced in JSON format

SMF statistics – SMF 115

MQ Statistics – The basic health of the QMGR

The SMF 115 data is the statistical information produced by a IBM MQ 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 resource ‘managers’ within IBM MQ

QMGR Health – Bufferpool Constraints

Red Flags for Bufferpools

- SOS

QMGR	BP	NumBuff	%now	%low	dwt	dmc	stl	stla	sos
QML2	3	70000	18	0	109	198908	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
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

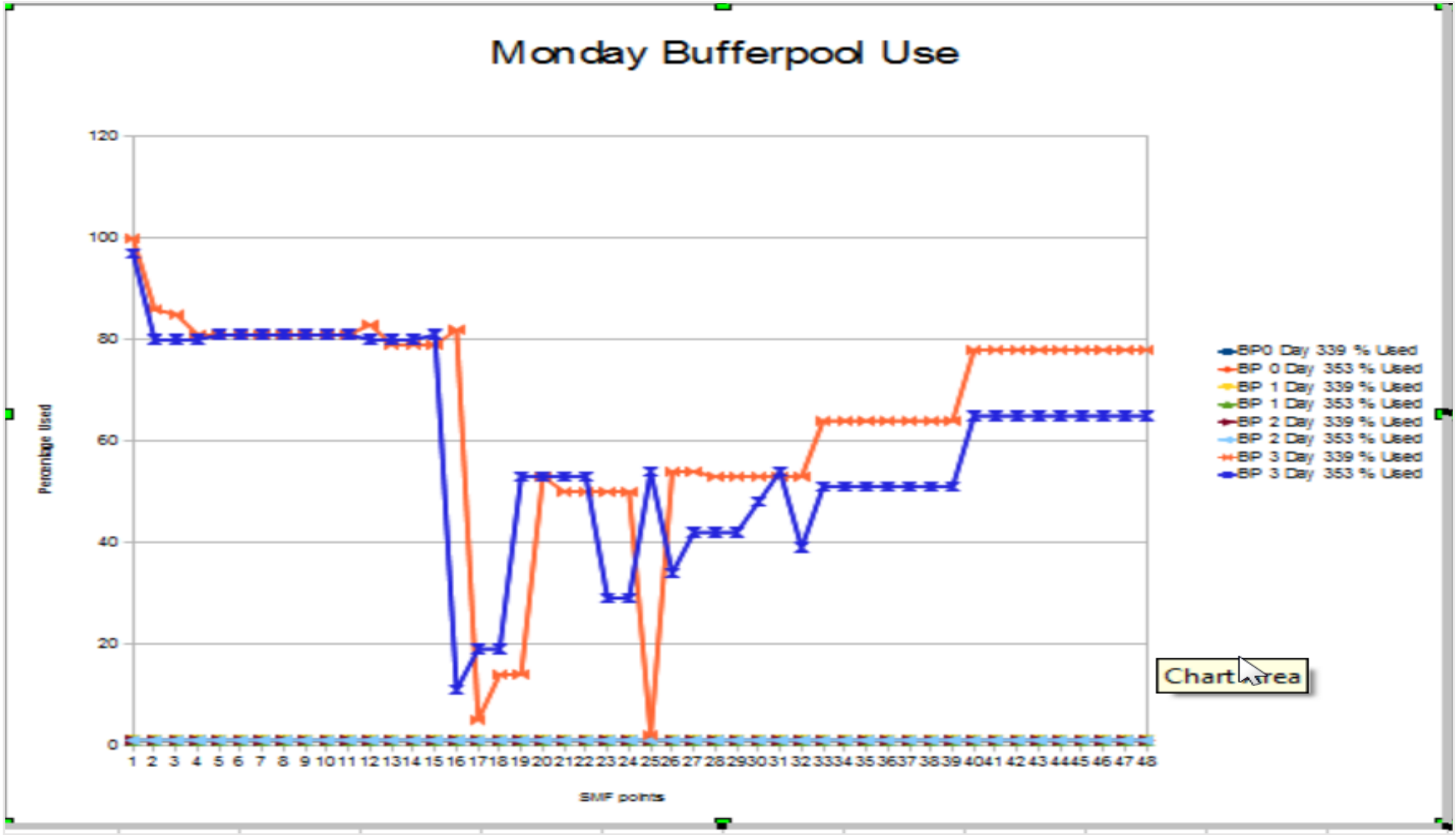
QMGR Health – Bufferpool Constraints

Red Flags for Bufferpools - Continued

- DMC – synchronous write process kicks off

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	132	36526	1232774	2	0

QMGR Health – spotting trends



QMGR Health – anticipating problems

Getting into the danger zone

- Consistently Approaching/Achieving 20 % Free pages

QMGR	BP	NumBuff	%now	%low	dwt	dmc	stl	stla	scs
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

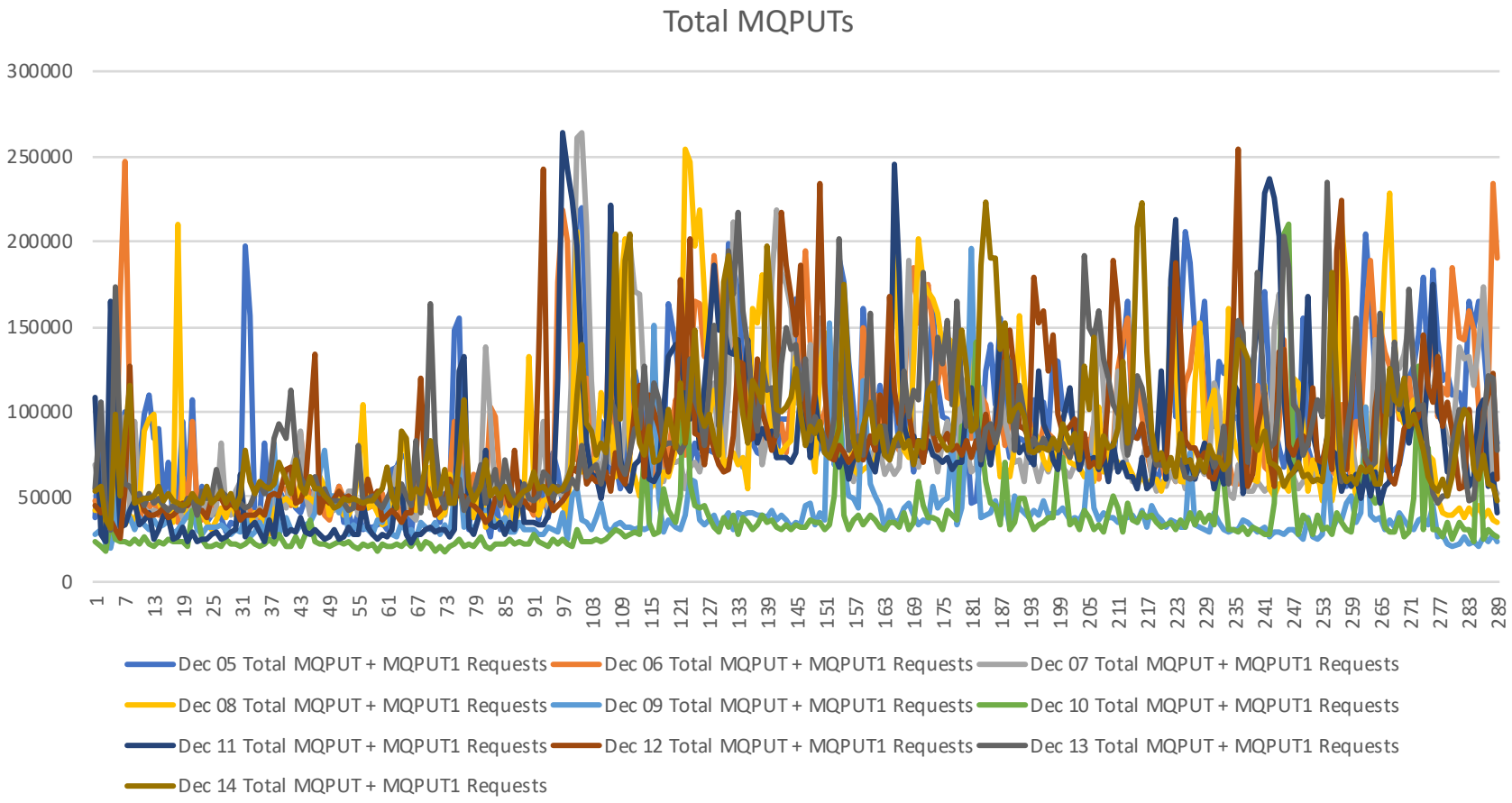
QMGR Health – Use trends

Message Manager Information

- Good indication of queue manager usage
  - This is only a count of API calls, not one of successful calls
  - MQGETs may or may not have data returned

Interval Duration (seconds)	Opens	Closes	Gets	Average Number of gets per second	Puts	Put1s	Total Put and Put1	Average Number of puts per second	Inq	Total Inq & Inq1	Total API Requests	Average Number of API calls per second
1800	122757	119906	516266	286.81	466991	0	466991	259.44	1035	1035	1293656	718.70
1801	124063	121691	509287	282.78	442368	0	442368	245.62	732	732	1198441	665.43
1796	123061	120178	538814	300.01	439458	0	439458	244.69	929	929	1278776	712.01
1793	104548	101209	505250	281.79	438985	0	438985	244.83	947	947	1204861	671.98
1803	100303	97322	514515	285.37	437573	0	437573	242.69	832	832	1199469	665.26
1788	133418	130388	531761	297.41	436343	0	436343	244.04	801	801	1301497	727.91
1800	109897	107752	498573	276.99	434699	0	434699	241.50	494	494	1151663	639.81
1794	99830	96833	501172	279.36	432264	0	432264	240.95	1133	1133	1180186	657.85
1787	126553	123691	515540	288.49	425578	0	425578	238.15	943	943	1246676	697.64
1785	102356	99829	471169	263.96	425410	0	425410	238.32	555	555	1099563	616.00
1786	134208	131756	513848	287.71	423742	0	423742	237.26	1058	1058	1263949	707.70
1787	83232	81100	470372	263.22	422571	0	422571	236.47	815	815	1087112	608.34

QMGR Health – Message Manager Trends



QMGR Health – Log use analysis

Log Manager Information

- Only indication of persistent messaging use

```
MPX1,QML1,2017/03/02,14:08:22,VRM:900,
  From 2017/03/02,14:06:55.422312 to 2017/03/02,14:08:22.217087, duration    87 seconds.
   Wait for buffers(should be 0):              0 out of         83, 0%
   Total Number of pages written:            219
   Number of pages written/sec:                2
   Amount of data written/sec:                 0 MB/Sec
   Total Number of write requests:           209
   Number of write requests/sec:               2
   Pages written per I/O:                      1
   Total number of read requests:              7
   Write_Wait          0, Write_Nowait      1043, Write_Force          16, WTB           0
   Read_Stor           7, Read_Active          0, Read_Archive         0, TVC           0
   BSDS_Reqs           2, CIs_Created         83, BFWR                141, ALR           0
   ALW                 0, CIs_Offload          0, LLCheckpoints        0
   Read delayed        0, Tape Lookahead       0, Lookahead Mount      0
   Write_Susp        142, Write_Reqs         209, CI_Writes          219
   Write_Serl          0, Write_Thrsh          0, Buff_Pagein          0
   _____,__ write requests,          CIs, Average I/O , After I/O    , pages/IO
                                          time in uSec, time in uSec,
   Log 1, 1 page         204,          204,          270,              5,          1
   Log 1,>1 page           6,           16,          320,              1,        2.7
   Standard deviation of first log, 1 page per I/O, response time +-          19
   Log 1, 1 page Longest I/O          1247 at 2017/03/02,18:06:56.160934 UTC
   Log 1, 1 page Longest Request      1265 at 2017/03/02,18:06:56.160934 UTC
   Log 1,>1 page Longest I/O           386 at 2017/03/02,18:06:56.160545 UTC
   Log 1,>1 page Longest Request       387 at 2017/03/02,18:06:56.160545 UTC
  Log write rate         0MB/s per copy
  Logger I/O busy : 0.07%
  Logger task busy: 0.30%
```

QMGR Health – Log Manager

This view of the log manager data is emphasizing the number of READs, an indication of applications backout out an inflight transaction

- In this sample, there were both buffer reads and active log reads
  - Need to look into applications to see why this is being done so often
- Also examine high number of checkpoints

INTERVAL_DURATION	UNAVAILABLE_BUFFER_COUNT	LOG_READ_OUTPUT_BUFFER	LOG_READ_ACTIVE_LOG	LOG_READ_ARCHIVE_LOG	TOTAL_LOG_READS	TAPE_CONTENTION_DELAYS	CHECKPOINTS	LOG_CI	MB_PER_SECOND
1795	0	623	4461	0	5084	0	10	2821634	6.14
1789	0	417	3337	0	3754	0	9	2825604	6.17
1796	0	540	2638	0	3178	0	12	3453542	7.51
1792	0	511	2307	0	2818	0	10	2972254	6.48
1789	0	449	2082	0	2531	0	10	2818718	6.15
1773	0	392	1952	0	2344	0	12	3445866	7.59
1798	0	424	1835	0	2259	0	10	3061346	6.65
1787	0	518	1725	0	2243	0	8	2460906	5.38
1797	0	381	1824	0	2205	0	14	4037442	8.78
1797	0	581	1597	0	2178	0	9	2778470	6.04
1791	0	306	1841	0	2147	0	11	3259292	7.11

SMF accounting – SMF 116

MQ Accounting – The gory details

The SMF 116 data is the accounting information produced by a IBM MQ for z/OS queue manager.

- Primarily used to determine what is going on within IBM MQ workload
- Heavyweight
  - Large volume of data
  - Some processing overhead
- Individual tasks get multiple large records produced
  - Each task gets records produced at the end of the task
  - Long running tasks (like channels, batch jobs, long CICS reader transactions) will get multiple sets of task records at each SMF interval
- Channel accounting records are accumulated and produced at SMF intervals (not when the channel stops)

Task and channel accounting

When is this data critical:

- 'We are missing our SLAs on some of our transactions'
- The statistics data shows bottlenecks, but not specifics
- Trying to identify what queues are actually in use

Can be a daunting task

- You have over 3M SMF116 class 3 records from one SMF interval to see if you can find the problem
  - That produces a TASK report of over 165M lines!
- And, of course, 'MQ is the problem'

What queues are being used and how?

When investigating buffer problems knowing which queues defined using the constrained bufferpool and/or pageset

- Especially when you have taken over admin, capacity planning, or an application from someone else
- Might need to identify the queues using the resource pool (buffers or CF structures)
- Might need to know how busy the queues actually are rather than a vague notion based on ‘RESET QSTATS’ or other less scientific methods

Which queues are in BP2?

```
SELECT DISTINCT Base_Name,
       Pageset_ID, BufferPool_ID
FROM (SELECT * FROM MQSMF.WQ WHERE BufferPool_ID = 2);
```

Results

BASE_NAME	PAGESET_	BUFFERPOOL_ID
XMITQ1	2	2
SYSTEM.ADMIN.CHANNEL.EVENT	2	2
REPLY_Q_1	2	2

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
```

What queues are being used and how?

Overuse of Temporary dynamic queues

- Often used for responses on 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 will have a fixed component (often starting 'CSQ' or 'AMQ'), the name of the application using it, and a random generated name

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
```

Summary

1. Introduced health-checking MQ
2. MQ JES Message log
3. Display commands
4. SMF statistics and accounting
   - Capture
   - Formatting
   - Statistics (SMF 115)
   - Accounting (SMF 116)

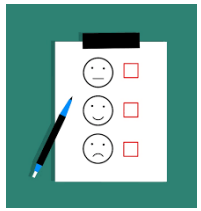
Additional Resources

- **MP16 –**
  - https://www-01.ibm.com/support/docview.wss?uid=swg24007421
- **MP1B -**
  - http://www-01.ibm.com/support/docview.wss?uid=swg24005907
- **MQSMFCSV**
  - https://github.com/ibm-messaging/mq-smf-csv

- **MQ Performance Report, available on Github**
  - https://github.com/ibm-messaging/mqperf

We want your feedback!

- Please submit your feedback online at ….
  - ➢http://conferences.gse.org.uk/2018/feedback/JL
- Paper feedback forms are also available from the Chair person
- This session is JL



Thanks for listening

Questions?

John Waldron IBM UK, IBM MQ L3 Service
John.Diarmuid.Waldron@.ibm.com