Performance Tuning and Problem Determination

Doug Burns - burnsd@uk.ibm.com
IBM UK

08/11/2017
Session JG

Agenda

- What is performance to me?
- Queue manager performance
- Channel initiator performance
- MQ application performance
- Summary

What is performance to me?

What is performance to me?

- Performance can mean different things:
  - Meeting sub-second SLAs on critical transactions
    - In spite of workload fluctuations
  - Meeting batch windows
  - Meeting previously set expectations
  - Performance is not availability
    - Though if resources are not available it can show up as a performance problem
- Performance is a matter of perception
- It can also mean different things to different applications in the same organization!

What is performance to me? – Notes

- Performance can mean different things:
  - Workload fluctuations can be predictable
    - Daily - Market open, ‘lunch time’ spikes
    - Weekly – Monday morning blues, Friday payday
    - Monthly – Pension day payouts
    - Annual – ‘Black Friday’, enrollment periods
  - Some workload fluctuations are not as predictable
    - “The market went nuts today”
  - Batch window are still critical to many businesses
    - SOA evolution has shortened or eliminated some, but the work still has to get done
  - Meeting previously set expectations
    - Yesterday my request was back in less than a second, today it is two seconds. MQ must be broken
  - Just as availability is not performance, performance is not availability
    - A sharp slowdown caused by performance problems may be perceived as an outage, just as a real outage may be reported as a performance problem

Performance issues with MQ

- MQ uses system provided resources
  - Performance issues in MQ may stem from underlying system constraints
  - Tuning for MQ can involve optimizing both MQ and the resources it uses
- Tuning is workload specific
  - Need to understand how MQ performs with your workload to determine how best to tune it
- We cannot cover everything so here are the most important points…

Performance issues with MQ – Notes

- MQ can only perform as well as the underlying resources that are used. There are a number of things that can be done with the queue managers to help control performance issues, some will be highlighted in this presentation, along with what has been exposed via the MQ SMF data that aid in finding tuning opportunities.
- Changes to MQ for z/OS itself that have been made in recent releases are highlighted. Often performance gains can be made by upgrading to the most current release.

Queue Manager Performance

Queue Manager Performance – Buffer pools

- Virtual Storage – Largest user is the buffer pools
  - For private queues, messages are put into buffer pools
  - Buffer pool allocation and tuning are critical for private queue performance
  - MQ Statistics records track use over time
  - Real time messages are indicators that the pool is currently constrained
  - Buffer pool thrashing
    - Caused by messages being put at the same time they must be read into a constrained buffer pool

Queue Manager Performance – Buffer pools – Notes

- Often the biggest ‘bang for the buck’ in tuning MQ on z/OS can be from evaluation of the bufferpool use
  - This is even true for MQ V8 users, where large above the bar bufferpools are supported
- Even though many are aware of the benefits of buffer pool tuning we still see problems in this area
- The next slides illustrate how to find ‘hotspots’ and potential problem areas
- Know your environment
  - As a warning to casual observers of performance data – if you don’t know what looks normal, it can be difficult to figure out what may be a problem
  - As a warning to serious observers of performance data – what looks normal, may not be right.

Buffer Pool Use – General Recommendations

- MQ buffer pool and page set reservations:
  - Reserve Buffer pool 0 and Page set 0 for MQ
  - Buffer pool 1 for ‘SYSTEM’ queues that do not get deep
  - Buffer pool 2 for ‘SYSTEM’ queues that may get deep
  - Reserve a separate buffer pool and page set for the SYSTEM.CLUSTER.TRANSMIT.QUEUE

Buffer Pool Use – General Recommendations – Notes

- Bufferpool and Pageset 0 should always be reserved to MQ to use
- If you have a large and active cluster, especially if you are using pub/sub in a cluster; please use a separate page set and bufferpool for the SYSTEM.CLUSTER.TRANSMIT.QUEUE. This will prevent the chattiness of the cluster, and possible cluster member outages from impacting other workload.

Buffer Pool Use – General Recommendations

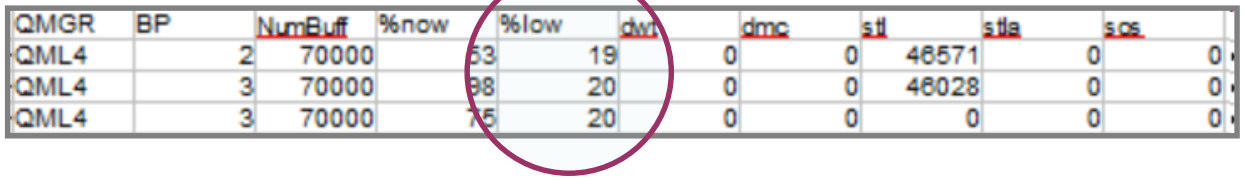
- Application Buffer pool use
  - Buffer pools 3-99 for application data
  - One or more buffer pools for short lived messages
    - Buffer pool should not fill up
    - Short lived may be seconds or minutes
    - Make buffer pool as large as necessary
    - Keep buffer pool < 85% full
      - DWT = 0
    - Allow for unexpected spikes
    - High volume request and reply queues should use different buffer pools where possible
      - And if not, separate page sets are recommended
  - Buffer pool for long lived messages
    - Expect messages to be moved to the page set
- Keep batch processing separate from transactional

Buffer Pool Use – General Recommendations – Notes

- Buffer pools for short-lived messages should have enough pages without ever getting over 85% full.
- Once the buffer pool gets over 85% full then the queue manger starts moving pages out to the page set to free up space in the buffer pool (the deferred write task or DWT in the statistics printing). Applications getting these messages on these pages may have to do disk I/O to retrieve them, which will slow down the applications.
- You should also ensure you have enough capacity to handle peak workloads, for example at busy times, and if there was an outage, so the work is now flooding in to the queue manager.
- If you have long lived messages, they will typically be flushed to disk, either to free up pages or if they have been in the buffer pool for 3 checkpoints. A relatively small buffer pool has an advantage, in that if the DWT is started frequently it is only writing a few pages at a time. If the buffer pool is very large, then the task may write a large number of pages when space is needed. This may have a small, but possibly noticeable, blip on the CPU resources used by the queue manager.

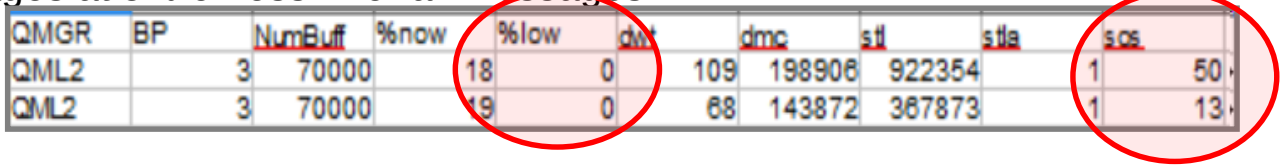
What does buffer pool stress look like?

- Free pages at 20% or less – for short lived messages



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

- Free pages at 5% or less – for all messages



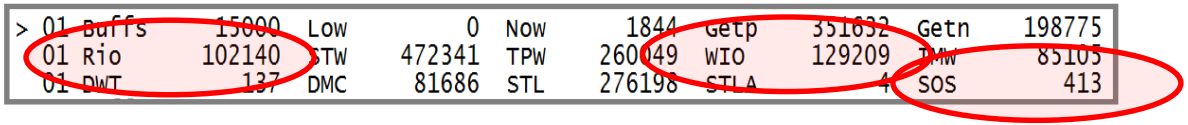
QMGR	BP	NumBuff	%now	%low	dwt	dmc	stl	stla	sos
QML2	3	70000	18	0	109	198906	922354	1	50
QML2	3	70000	19	0	68	143872	367873	1	13

What does buffer pool stress look like? – Notes

- What is shown is from an older Buffer Manager report, imported into a spreadsheet
- Things to watch out for include:
  - A concentration of highly active queues in one bufferpool
    - Avoid the ‘define like’ syndrome
  - A mixture of long lived and short lived messages in the same buffer pool
  - High volume request and reply queue on the same page set and buffer pool
  - Change in usage patterns over time
- The best I/O is no I/O
- In the samples shown
  - The free pages at 15% is the point where the async write task normally kicks off
    - Not a problem for long lived messages, but may be an indicator of problems if the messages are short lived
  - The free page at 5% or less is a real sign of problems, no matter the type of messages
    - The synchronous write process will start, delaying work and consuming additional CPU

What does buffer pool thrashing look like?

- Bufferpool churn example Note the ‘low’ value of ‘0’ and the SOS value of 413



  - The bufferpool went to short on storage 413 times in a 5 minute interval
  - There were 129,209 writes to the page sets, and 102,140 reads from the page sets
  - The async write process threshold was hit 137 times
  - The synchronous write process threshold was hit 81,686 times!

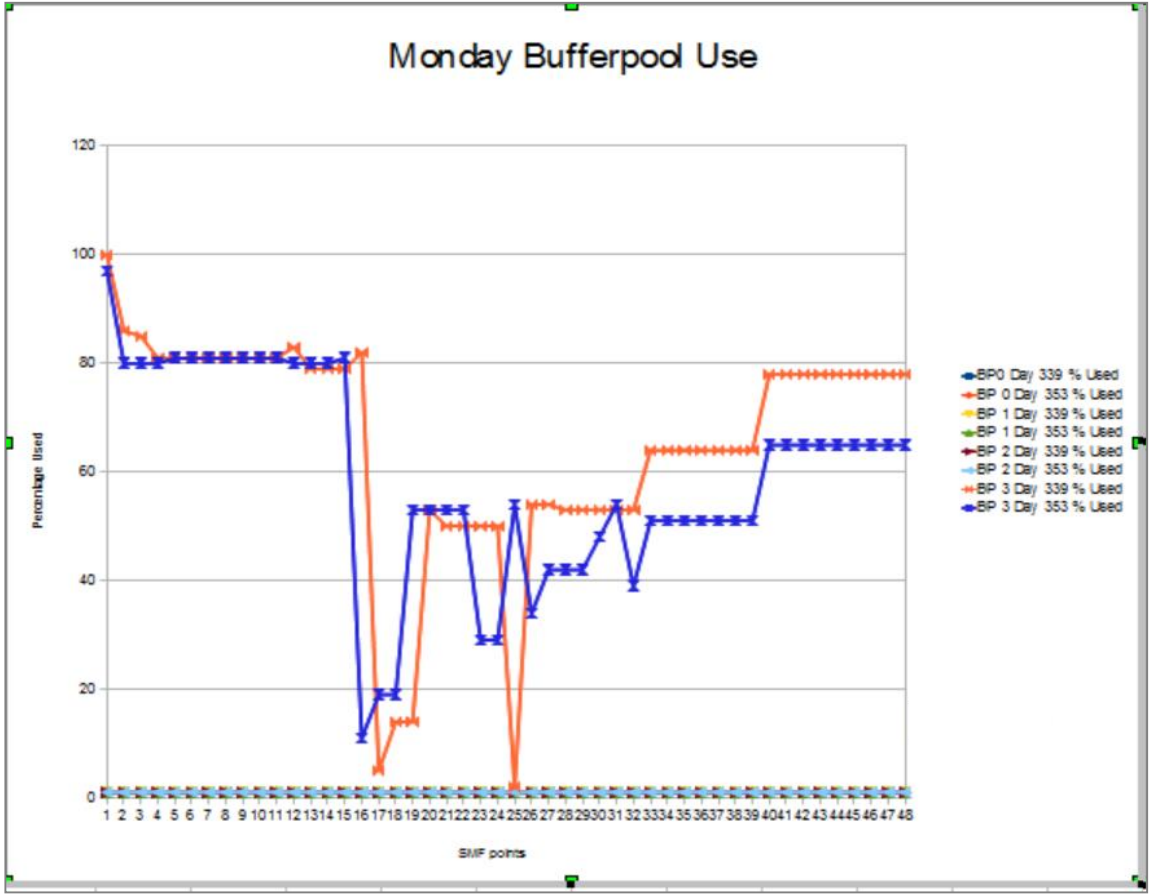
- JES log also had repetitions of the following messages

```
CSQP020E QML1 CSQP1RSW Buffer pool 1 is too small
CSQP020E QML1 CSQP3GET Buffer pool 1 is too small
```

What does buffer pool thrashing look like? – Notes

- This information is taken from the old MQ1150 SMF print program
- In this truly fine ‘bad example’, the volume and overuse of a single bufferpool and pageset
- The SMF interval was really short for this evaluation, 2 minutes
- Messages were being read (RIO) at the same time they were having to be flushed to disk
- This created a huge amount of overhead in the queue manager, and caused a number of slowdowns as real I/O had to take place.
- In this case breaking up the queues in this page set into multiple page sets greatly reduced the contention. Contention reduction made a substantial improvement in performance, and lower CPU utilization for the queue manager as a whole.

Buffer pool Trends and Analysis



SMF 115 – Buffer pool Trends and Analysis – Notes

- In the chart shown two high volume days were compared to see if there was a pattern to the BP use.
  - The ‘Y’ axis shows the percent of the bufferpool that is used
  - The ‘X’ axis shows the SMF point during the 24-hour periods (SMF is set to 30 minutes)
  - BP 0, 1 an 2 showed almost no utilization.
  - BP 3 was in very heavy use, some of the time.
  - BP 3 is under some stress.
- Having multiple days worth of data is vital, had there just been one heavy day the buffer pool use shown may have been an anomaly. Data from longer periods of time, when compared like this can be very useful in tracking usage, and predicting when there may be issues.
- In this case there was a clear pattern of overuse of buffer pool 3, in further evaluation the SMF116 data showed that all the queues that were being used for this queue manager were defined on the same page set/buffer pool. By moving some of the queues to another resource pool, the stress was reduced, work flowed faster and the CPU usage was reduced.

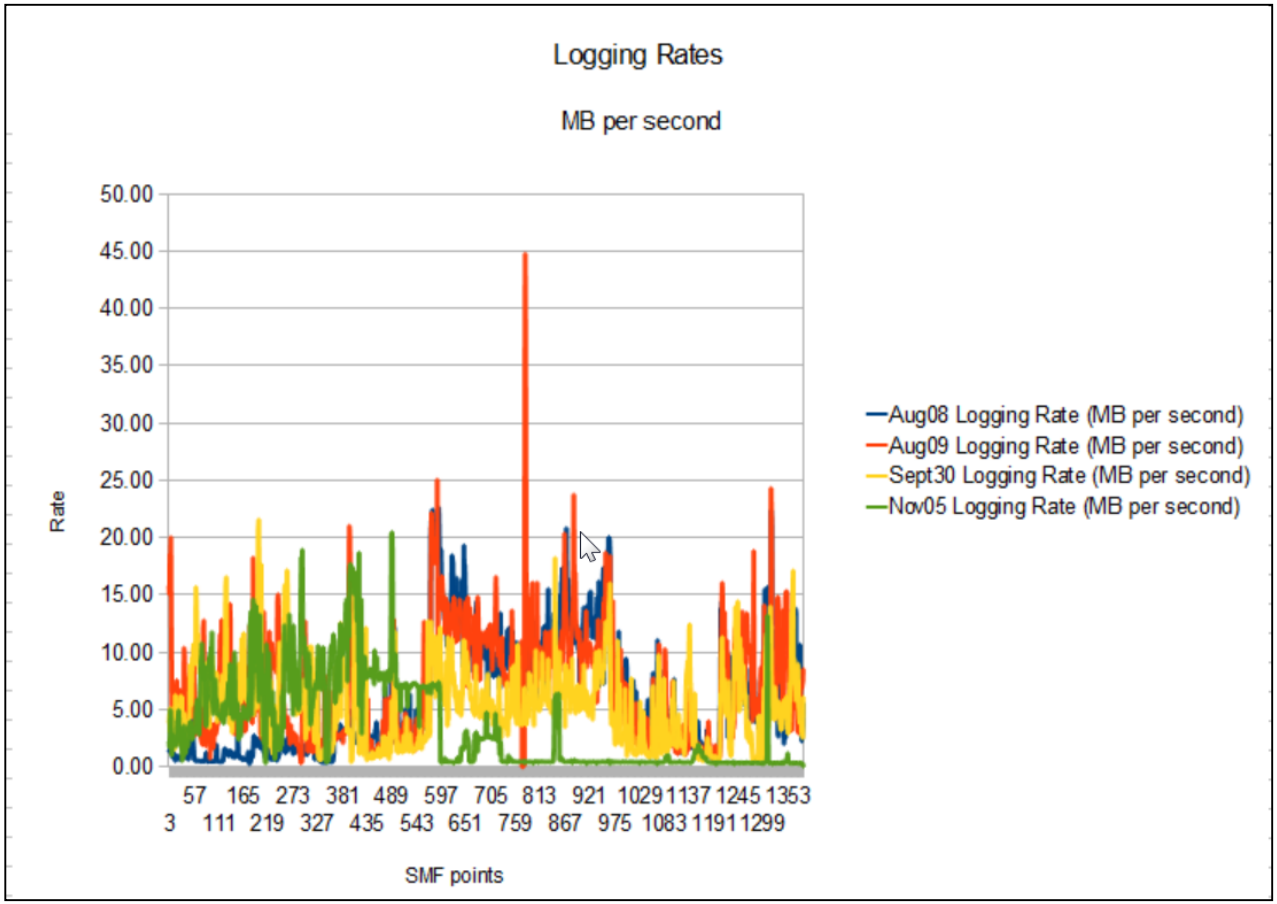
Queue Manager Performance – Logging

- All Persistent messages are logged to disk
- MQ log data sets are limited to 4GB
  - This limit is made obvious in V7.1
    - Message - CSQJ499I: Log data set is larger than 4GB
- Logs are switched when full
- Checkpoints are issued at log switches and at LOGLOAD points
- Proper positioning of the log files is a critical factor
- When examining real I/O
  - Look not only for averages, but for outliers

Queue Manager Performance – Logging – Notes

- Logs can be defined at larger than 4GB, but MQ only uses 4GB of them. A log switch will occur when it reaches that limit.
- Checkpoints counted in the SMF115 data are only those that are LOGLOAD driven, checkpoints at log switches are not included in that count.
  - This changed, and is undocumented (to date) in V9. The checkpoints driven by log switches are now included in the count.
- I/O rates can vary dramatically, there have been a number of performance issues related to work backing up behind a single or a very few slow responses from the I/O subsystem.
- Many documents would have people believe that with more current hardware and software the positioning on devices is less critical. In the past few years a number of critical performance issues have been traced back to things like active and archive logs on the same physical device causing periodic slowdowns.
- Average I/O rates, like any averages, can be misleading. While my average may be quite good, included in that average may be the one or two long response times that is causing serious back-up.

Logging Rate Trends and Analysis

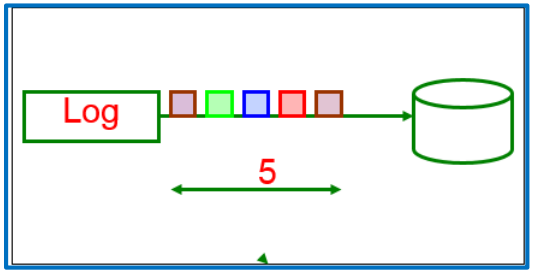


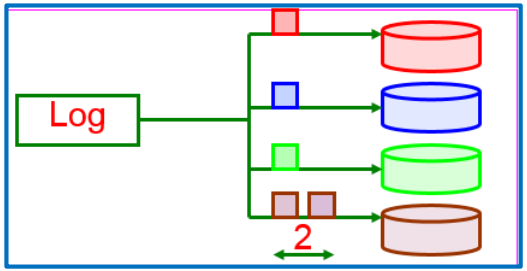
Logging Rate Trends and Analysis – Notes

- The chart is taken from real data. The spike, which had to be investigated was not a volume spike, but actually from a change in the SMF interval.
- In this case the queue manager was being almost constantly throttled by logging I/O limit (old hardware, max. I/O rate was 25 MB/second), in this environment.
  - They moved to newer hardware and were able to achieve >50 MB/second

Queue Manager Performance – VSAM striping

- What is VSAM Striping
  - Simply it is using more than one physical location for the data
  - With VSAM striping writing 4 consecutive pages, each page will be mapped to one of 4 volumes, frequently reducing contention. Data transfer time is typically less.
- Significant performance benefits for large messages
  - But archive log cannot exploit this, so may be a bottleneck
  - Consider using VSAM striping if you have a peak rather than sustained high volume





VSAM Striping – Notes

- VSAM striping is part of DFP (VSAM). When you define your data sets you can define the data set as being striped.
- Without striping if you wanted to write 5 4K pages to a track, then one requests would be issued, and 5 pages of data sent to the device, for example page 1,2,3,4,5 on track 7. The time for the I/O requests is essentially the time to send 5 pages down the channel – perhaps 5 ms.
- With VSAM striping, the pages are spread across different volumes, so pages 1&5 would be written to volume 1, page 2 to volume 2, page 3 to volume 3, and page 4 to volume 4. So we now have 4 I/Os in parallel, with at most 2 pages per I/O. This may take 2ms – which is less than the 5ms above.
- You can use VSAM striping for active logs, but not for archive logs. So if you have a sustained high log throughput you may run out of active logs, and have to wait for logs to be archived.
- One customer, who resisted the advice to stripe the logs for years (and the advice came from many sources), saw a 35% increase in throughput when they moved to striped logs.
  - They had tried using striped logs some years prior and found there was no benefit at the time
  - What they had not considered is that in the intervening years, their message sizes and volume had grown substantially.

Queue Manager Performance – Shared Queues

- Coupling Facility
  - CF storage constraints and large message performance problems can be mitigated by Shared Message Data Sets
    - New with V7.1
    - Reduces (but does not eliminate) the dependence on DB2 – large (over 63K) messages can be stored on SMDS
  - Links to the CF can become saturated, causing delays and performance problems

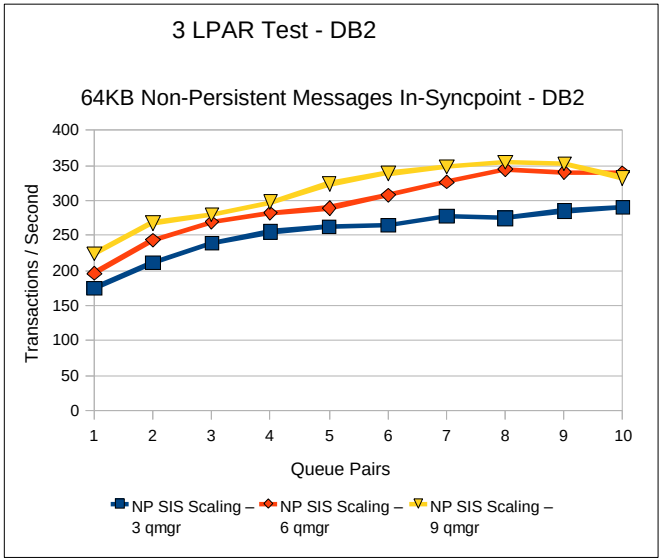
Shared Queues – Notes

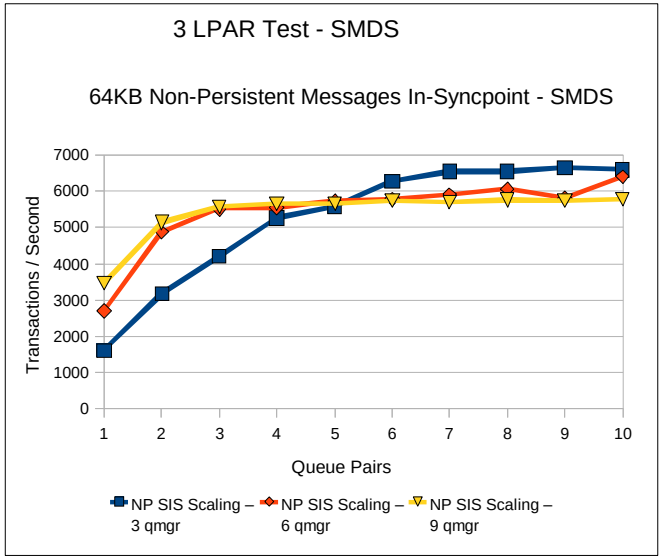
- The Coupling Facility can be a performance bottleneck when links to the CF from the LPAR where MQ resides become saturated. The CF activity reports, based on the RMF data show issues like this. The systems programmers or capacity group should be monitoring the CF activity reports.

- We do recommend that you become familiar with the CF activity reports.

Queue Manager Performance – SMDS

- Test Results on z196
- Tests show comparable CPU savings making SMDS a more usable feature for managing your CF storage
- SMDS per CF structure provides better scaling than DB2 BLOB storage



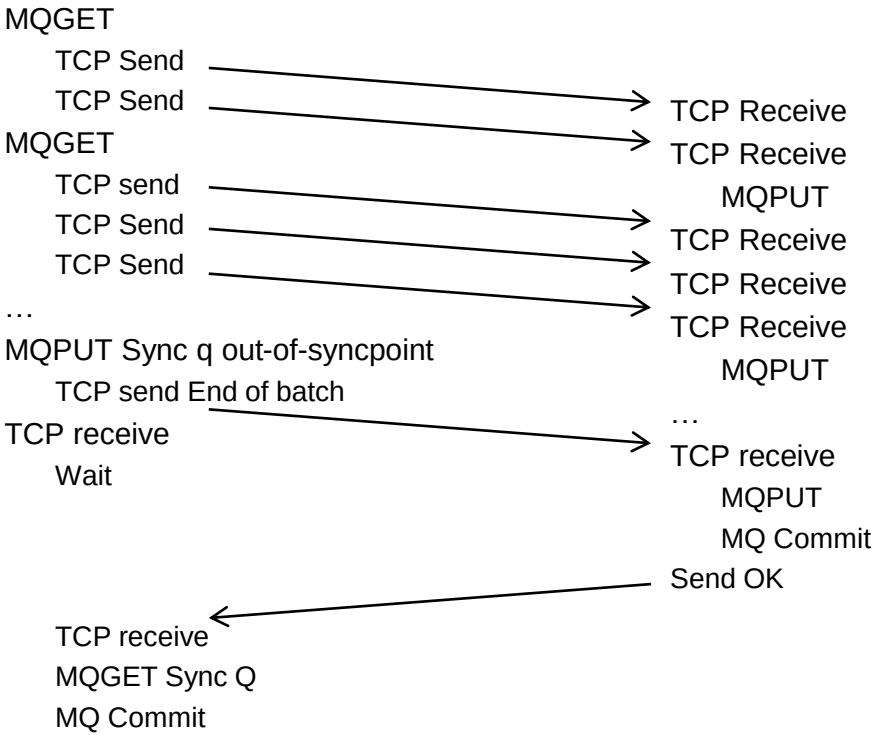


Queue Manager Performance – SMDS – Notes

- Using VSAM datasets for large message storage is a remarkable improvement over DB2

Channel Initiator Performance

How does a channel work?



Where is the time spent

- Receive end
  - TCP receive + MQPUT + TCP send typically 10 microseconds
  - MQCommit 1000 microseconds
- At end of Batch if channel keeps up
  - MQPUT of last message
  - MQCommit
  - Typically 1 millisecond

Where is the time spent

- Sender end
  - MQGET + TCP send typically 10 microseconds
  - MQPut out of syncpoint 1000 microseconds
  - MQCommit 1000 microseconds
  - Wait for remote end to reply OK typically 1000 microseconds
- To send 1000 messages
  - Batch size of 1 = 1000 * (10 + 1000 + 1000+ 1000) = 3.01 seconds
  - Batch size of 100 = 10 * (100 * 10 + 1000 + 1000+ 1000) = 40 ms
- Large batch sizes are good – 1000 is OK

What can cause problems

- Queue fills up at receiver end
  - Messages in pipe for queues Q1,Q2,Q3
  - MQPUT Q1 - OK
  - MQPUT Q2 – gets queue full
    - MRTMR set – so channels waits for this time and retries MQPUT. Does this MRRTY times. MQPUT either works, or message put on the DLQ
  - MQPUT Q3 – but this was delayed because of Q2 processing!

What can cause problems

- Unreliable networks
  - Channel keeps stopping
  - Batch needs to be resent
  - Having smaller batch size may help
- Large messages
  - Processing 1000 large messages (>1MB) in one batch may cause buffer pool to fill up
  - Use BATCHLIM to limit amount of data per batch e.g. 10 MB

How can I monitor channels

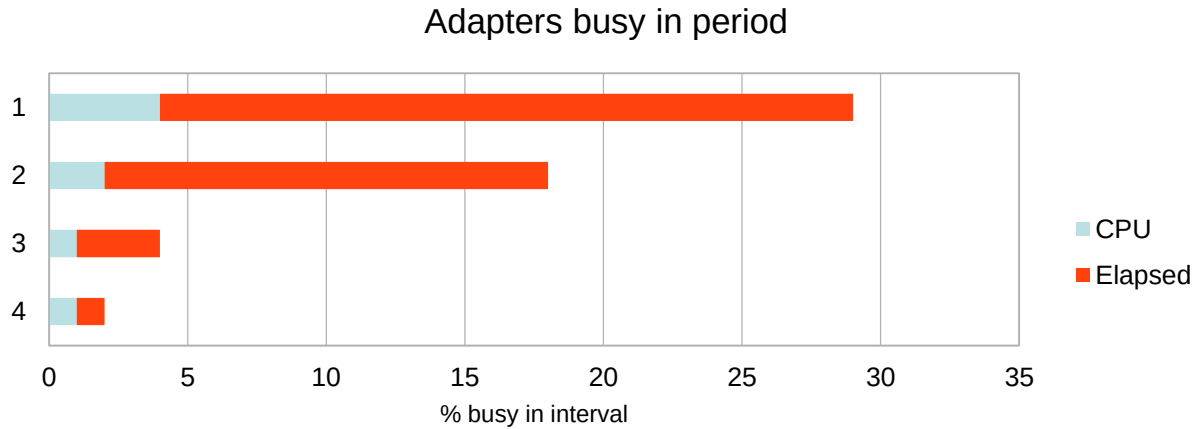
- SMF data for chinit
- DISPLAY CHSTATUS command
  - BATCHSZ – limit how many messages per batch
  - XBATCHSZ – what you actually achieving.
    - If XBATCHSZ close to BATCHSZ then consider increasing BATCHSZ
  - MSGS – messages processed since channel started
  - BYTSSENT/BYTSRCVD bytes send received → Rate

What can you do with the CHINIT SMF data?

- New in V8
- Statistics
  - Maximum number of channels used in interval
  - Adapter, Dispatcher, SSL, DNS server; TCB usage
- Accounting
  - Like Display Channel Status + more fields
- Low cost
  - Records produced when stats collected
  - Or if channel ends in the interval

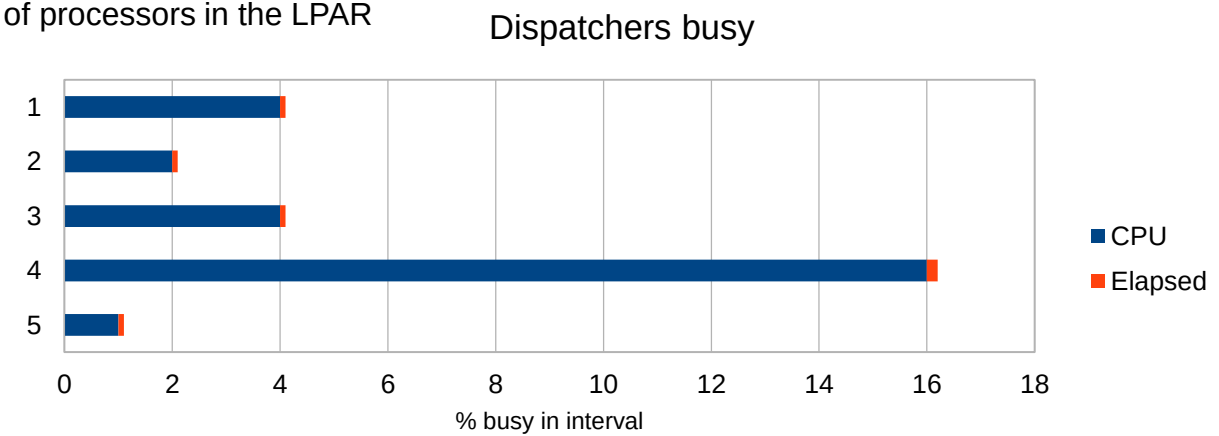
Chinit Statistics – Adapters

- Executes MQ requests, so can have large elapsed time and little CPU
- Pool of TCBs, first free used
- If all used then need more!
- Check you have some unused adapters



Chinit Statistics – Dispatchers

- There is Channel to Dispatcher affinity
  - Uses the same dispatcher for duration of the connection
  - Stop and restart channel may use a different dispatcher
- Do you have a hot dispatcher (> 80% busy?)
  - Few channels
  - Add more dispatchers
  - Stop and restart channels
  - Can vary from day to day – depends on start order
- Number dispatchers <= number of processors in the LPAR



Channel accounting data – sender channel

Records prefixed with

`Channel Address` e.g. `TO_MVSCA  winvsca.hursley.ibm.com`

Data records

```
..Connection name                winmvsca.hursley.ibm.com
..Batch size                                   50
..Messages/batch                             48.8
..Number of messages                       33,896
..Number of persistent messages            33,896
..Message data                        120,263,008   114 MB
..Persistent message data             120,263,008   114 MB
..Total bytes sent                    120,266,176   114 MB
..Total bytes received                     19,996    19 KB

..Bytes sent/Batch                        173,044   168 KB

..Bytes received/second                       137   137  B/sec
..Bytes sent/second                       829,421   809 KB/sec

..Net time average                            116 uSec
..Net time min                                 71 uSec
..Net time max                             14,481 uSec
```

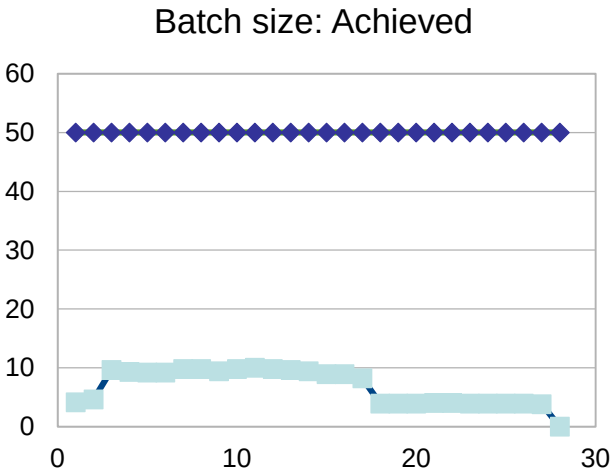
What can you do with the channel accounting?

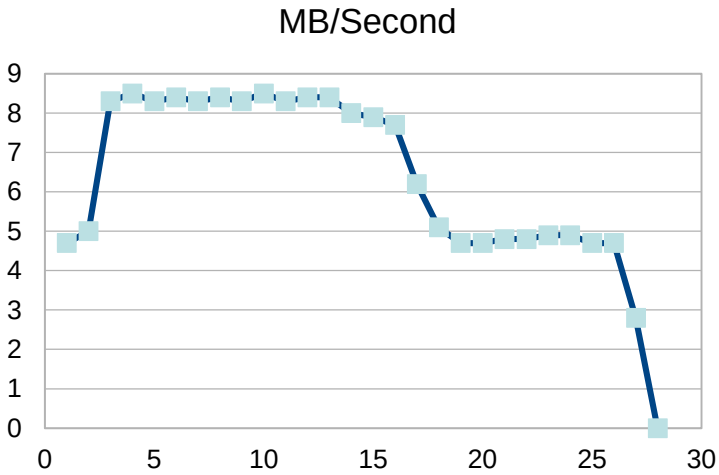
- What are my hot channels?
  - Bytes received/second, Bytes sent/second
- What is the net time
    - Net time max, average, min
    - When was maximum? (search for "*Net time max date&time*")
- Does my channel have more capacity?
  - Number of full batches
  - Xmitq empty count – queue was empty
  - XBATCHSZ & batch size
  - Bytes sent/Batch (Batchint)

Notes

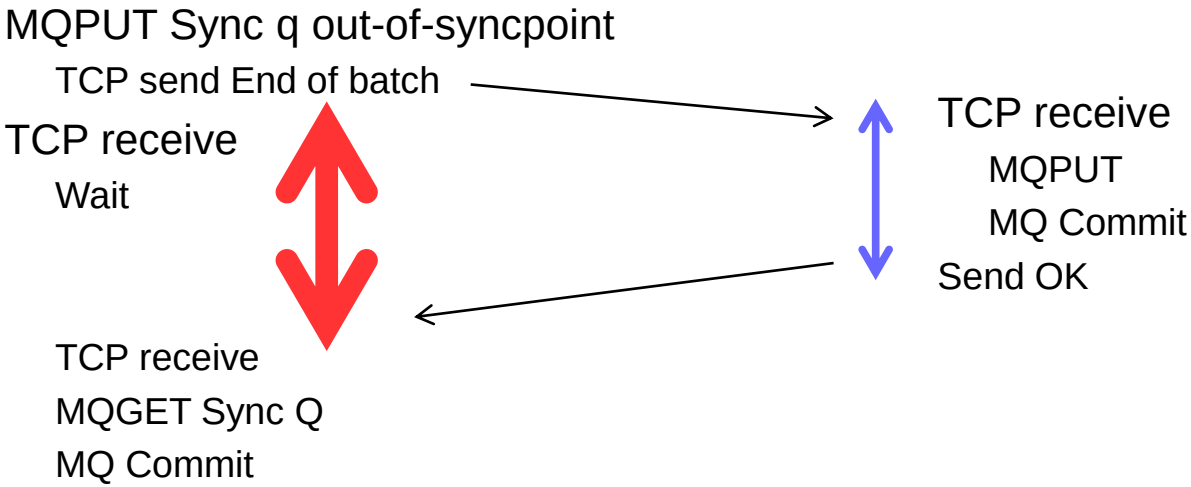
- You can edit the output and include only the total bytes row, then sort on the number column to see channel with highest throughput
  - May want to do it on rate – in case channel started and stopped in the interval
- Max nettime
  - Can your correlate this to when you have a slow down?
- Messages per batch (the XBATCHSZ), tells you if you have spare capacity. If equal to batch size it was always busy.
  - May have hit batchlim – so xbatchsz will be smaller than batch size

Channel Accounting Data – Examples





DIS CHS Nettime



- Red time – Blue time = Nettime
- Nettime should be similar to TSO PING time

What's happening to my channel network?

- Ping

```
//RUN1GB  EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
 ping winmvsac (count 1000 verbose length 32768
/*
```

- Across the Hursley site
  - Packets: Sent=1000, Received=1000, Lost=0 (0% loss)
  - Approximate round trip times in milliseconds:
  - Minimum=1.03 ms, Maximum=10.19 ms, Average=1.13 ms, StdDev=0.50 ms
- Hursley to Mainz in Germany
  - 64 bytes Minimum=22.37 ms, Maximum=32.30, Average=24.08
- UK to the US
  - 64 bytes Minimum=94.39 ms, Maximum=95.54, Average=94.71

What's happening to my Channel Network

- NETSTAT
  - NETSTAT ALL (CLIENT MQ02CHIN
  - NETSTAT ALL (IPPORT 1.2.3.4+5678
- Output – from start of connection

```
CongestionWindow:  0000065536    >= 64K is good
SndWnd:            0000065536    >= 64K is good
MaxSndWnd:         0000065536    >= 64K is good
Round-trip information: Smooth trip time: 0.000
SmoothTripVariance: 1.000
ReceiveBufferSize: 0000065536    > 64K is good
SendBufferSize:    0000065536    > 64K is good
ReceiveDataQueued: 0000000000    ? Problem
SendDataQueued:    0000000000    May be transient
ReXmt:             0000000000    = 0 is good
ReXmtCount:        0000000000    = 0 is good
DupACKs:           0000000000    = 0 is good
```

Current remote end buffer available

0.00 0 ms is good

Lost packets

Notes

- Use NETSTAT ALL (CLIENT MQ02CHIN to list all the connections for the chinit. Good idea to do it in batch because of potential large number of connections
- Then use NETSTAT ALL (IPPORT 1.2.3.4+5678 for IP address+port
- Netstat data is from start of connection
- Send window is the buffer size at the remote end. 64KB is good.
- If there is congestion this can go down, so compare with max send size
- Round trip time is as seen by TCPIP
  - Variance tells you how much variation there is (e.g. range)
- ReXMT is the number of packets dropped and retransmitted. If this is 0 it is good. If > 0 then this indicates there was a problem at some time. Wait for a period and try again. If numbers go up, this indicates a problem.

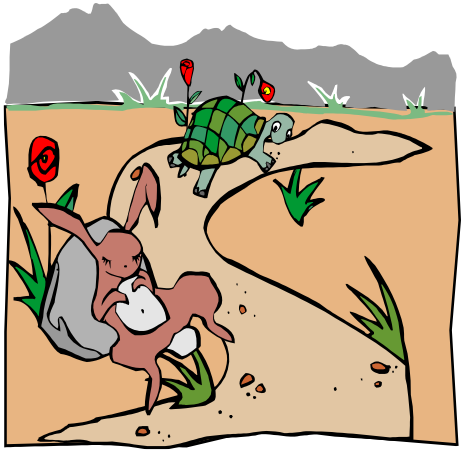
Tuning TCP parameters

- If large distance between sites – or ping is large (5 ms)
  - UK to US ping takes 80 ms
- May need to configure MQ to tell TCP to use very big buffers
  - If 'Receive Window Size' < 32KB, TCP sends 32KB and waits for Ack
    - So send 32kB and wait 80 ms
  - If 'Receive Window Size' >= 64KB
    - TCP gradually increases amount of data – x MB
    - “Dynamic Right Sizing”
    - Search “Getting the best throughput with MQ TCPIP channels COLINPAICE”
    - From MQ V8 you can use the commands
      - RECOVER QMGR(TUNE CHINTCPRBDYNSZ nnnnn)
      - RECOVER QMGR(TUNE CHINTCPSBDYNSZ nnnnn)

MQ Application Performance

MQ Application Performance

- Choose the right queues
- Choose the right messaging styles
- Choose the right verbs



Choosing the right queues – Temporary Dynamic Queues

This information was taken from the SMF116 class 3 records

```
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
```

Choosing the right queues – Permanent Queues

This information was taken from the SMF116 class 3 records

```
== 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
```

Choosing the right queues

- Choose the right queue
  - On z/OS Temporary Dynamic queues should be avoided
    - Higher CPU costs
    - Elapsed time can be significantly longer
- The CPU cost comparison

Verb	TDQ	Permanent	Difference
Open	125	38	238%
Close	111	26	327%
Put	104	113	-8%
Inquire	17	18	-5%

- The Elapsed Time comparison

Verb	TDQ	Permanent	Difference
Open	850	39	2079%
Close	113	26	3347%
Put	106	115	-8%
Inquire	17	18	-5%

Choosing the right queues – TDQs – Notes

- The data shown is taken from one of the SMF print programs from an older version of MP1B (MQ116S)
- The information presented here is a mixture of counts and averages
  - The MQ calls made is a count of the calls made as part of this unit of work, or during the interval for long running tasks.
  - The ‘ET’ (elapsed time) and ‘CT’ (CPU time) are averages for the unit of work.
  - The remaining fields are counts.
- Not only is the CPU noticeably higher, note the suspend count. If the application has very strict SLAs avoiding the opportunity for suspensions can be critical in a heavily loaded system.
- TDQs are often used as reply queues for online monitors, which seems like such an oxymoron. Most monitors have optional permanent queues.

MQ Application Performance – Queue Index

- Queue index specification is unique to MQ on z/OS
  - 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 use of a proper index can substantially improve performance an CPU consumption.

MQ Application Performance – Queue Index – Notes

- The information that follows illustrates the need for proper queue definition based on application use. It also shows where the MQ Admins and application programmer can find out what is going on within an application.
- Many of the slides are from SMF 116 data that has been printed using MQ116S, from an older version of SupportPac MP1B.

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
Open First opened 12-03-2012 15:12:58.55
Base Last closed  **-**-**** **:**:**.**
Queue Page set ID            4,  Buffer pool         3
First Current opens          1,  Total requests        61
Last Generated messages :         0
Page Persistent messages: GETs         0,  PUTs          0,  PUT1s         0
Curre Put to waiting getter: PUT         0,  PUT1         0
Gener GETs: Valid       20,  Max size       80,  Min size       80,  Total bytes    2240
Persi GETs: Dest-S      28,  Dest-G         0,  Brow-S        0,  Brow-G        0, Su cessful destructive        28
Put t Time on queue : Max 4583.730054,  Min  257.434901, Avg 3956.326341
GETs:
GETs:
Time -MQ call-          N        ET        CT       Susp       LOGW       PSET Epages   skip expire
-MQ c  Get    :         28       384       369         0          0          0      0   3505      0
 Get   Inquire:         28        22        21
 Inqu  Maximum depth encountered       258
Maxim
```

Non-Indexed Queue Retrieval – Notes

- How can you tell if a queue is being read for a specific message?
  - In V8 and above there is a JES log message that shows when a queue should be indexed, CSQI004I .
  - Some admins have already suppressed this message because of the frequency. The SMF116 class 3 data will also indicate this is an area of opportunity.
- In the SMF 116 class 3 data record, the fields of interest are:
  - The Queue Indexing
  - The Type of GET request being made. Those with a '-S' are for specific messages (Get by correlid, get by message id, etc.). Those with a –G are generic, get the next message on the queue.
  - The average CPU expenditure for the successful gets – the 'CT' column highlighted
  - The number of pages skipped while finding matching messages

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 Queue Retrieval – Notes

- Note the differences between the non-indexed and indexed retrieval. In particular, no pages had to be skipped during the MQGET process. That saves both CPU and elapsed time.
- In practice, differences were seen with queue depths as low as 5-10 messages.

Indexed vs Non-indexed – Comparison

- Comparing the CPU time, both queues with the same max message depth
  - Indexed - 27 messages at an average of 99 CPU microseconds
    - 2673 µs for 27 messages retrieved
  - Non-indexed 28 messages at an average 369 CPU microseconds
    - 9963 µs for 27 messages retrieved
  - Difference 272%
- Comparing the elapsed time
  - Indexed - 27 messages at an average 105 microseconds
    - 2835 µs elapsed time for the messages
  - Non-Indexed 28 messages at an average 384 microseconds
    - 10368 µs elapsed time for 27 messages
  - Difference 252%

MQ Application Performance – Queues

- Choose the right messaging style
  - Persistent messages are more costly than non-persistent
  - Use non-persistent messaging
    - When the message is a simple query
    - Easy to discover and recover
  - Use Persistent messaging
    - When the message drives an update transaction that must be coordinated
      - When designing/writing/testing the application recovery code is too challenging
    - Difficult to recreate the request
    - When required to by a business

MQ Application Performance – Choose the right verbs

- Like any other subsystem, the choice of verbs can improve performance and scalability
  - Recycling code is a positive
    - Reduces development time and effort
    - Often enforces best practices
    - Can reduce testing time
  - Recycling code is a negative
    - Can introduce performance problems if code not well understood
  - Increased use of a transaction can expose underlying issues

Choose the right verbs

- Misuse of MQPUT1
  - MQPUT1 combines an MQOPEN, MQPUT and MQCLOSE into one verb
  - Typically used for the reply messages on request/reply processing
  - More efficient if just putting one message
  - Substantial performance impact if putting multiple messages to the same queue

Effect of MQPUT1

- Each MQPUT1:
  - 117 µs CPU, for a grand total **351,000** µs
  - 121 µs Elapsed time, for a grand total of 363,000 µs
- Each MQPUT:
  - 72 µs CPU, for a grand total of **216,000** µs
  - 74 µs Elapsed time, for a grand total of 222,000 µs

```
PUTs: Valid      3000,  Max size        80,  Min size        80,  Total bytes 240000
-MQ call-          N         ET         CT        Susp       LOGW       PSET Epages   skip expire
 Put1   :       3000        121        117           0          0
```

```
PUTs: Valid      3000,  Max size        80,  Min size        80,  Total bytes 240000
-MQ call-          N         ET         CT        Susp       LOGW       PSET Epages   skip expire
 Open   :          1         84         81           0
 Close  :          1         18         18           0
 Put    :       3000         74         72           0          0
Maximum depth encountered       6000
```

Effect of MQPUT1 – Notes

- Remember that both elapsed time and CPU time reported in this section of the old MQ116S report is the average time, not the total time.
- The TASK output of the new MQSMF program also reports average CPU time.

Effect of MQPUT1

- For one PUT it is less expensive to use an MQPUT1
  - MQPUT1 117 total µs
  - MQPUT 171 total µs
- For two PUTs it is less expensive to use an MQOPEN, MQPUT and MQCLOSE
  - MQPUT1 234 total µs
  - MQPUT 213 total µs
- Draw your own conclusions

MQPUT vs MQPUT1 Comparison – Notes

- In one particularly good example of this, the WSC MQ people were reviewing CPU use for a very high volume queue manager. A single CICS transaction was issuing 7,000+ MQPUT1s to the same queue for each execution. The transaction, once executed a few hundred times a day had become a service. It was now being executed thousands of times a minute.
- Like the Inquisition, no one expected the dramatic jump in CPU.

Summary

What to do when you get back to the office...

- Understand your queue manager performance profile
  - Collect SMF statistics regularly
  - Collect accounting for short period
- Use SMF data
  - When you think you have a problem
  - Is my workload growing?
  - Health check to see if applications are well behaved
- Look at log stats
- Are buffer pools filling up?
- CF response time
- What are the high use channels?
- Are there enough adapter tasks?
- Look at the 'problem' messages produced by MP1B

More information

- Performance is a huge topic, we have only scratched the surface. There is a lot more investigation that can be done, and more information being published regularly.

- There are a number of SupportPacs available:
  - MP16 – Capacity Planning and Tuning for WebSphere MQ for z/OS
  - MP1K – Performance Report – IBM MQ for z/OS V9.0
  - MP1J – Performance Report – IBM MQ for z/OS V8.0
  - MP1H – Performance Report – WebSphere MQ for z/OS V7.1
  - MP1G – Performance Report – WebSphere MQ for z/OS V7.0.1
  - MP1F – Performance Report – WebSphere MQ for z/OS V7.0.0
  - MP1B – Interpreting accounting and statistics data IBM MQ for z/OS

More information

- There are a number of SupportPacs available:
  - MP16 - Capacity Planning and Tuning for WebSphere MQ for z/OS
    - http://www-01.ibm.com/support/docview.wss?uid=swg24007421
  - MP1K - IBM MQ Version 9.0 for z/OS
    - http://www-01.ibm.com/support/docview.wss?uid=swg24042470
  - MP1J - IBM MQ Version 8.0 for z/OS
    - http://www-01.ibm.com/support/docview.wss?uid=swg24038347
  - MP1H - Performance Report - WebSphere MQ for z/OS V7.1
    - http://www-01.ibm.com/support/docview.wss?uid=swg24031663
  - MP1G - Performance Report - WebSphere MQ for z/OS V7.0.1
    - http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24024589&loc=en_US&cs=utf-8&lang=en
  - MP1F – Performance Report - Performance Report - WebSphere MQ for z/OS V7.0.0
    - http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24020142&loc=en_US&cs=utf-8&lang=en
  - MP1B - Interpreting accounting and statistics data WebSphere MQ for z/OS
    - http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24007421&loc=en_US&cs=utf-8&lang=en

Thank you and any questions?

Notices and Disclaimers

- Copyright © 2017 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.
- **U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.
- **Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law

Notices and Disclaimers Con’t.

- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM’s products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

- The provision of the information contained h erein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

- IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services ®, Global Technology Services ®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli®, Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Session feedback

- Please submit your feedback online at:

  http://conferences.gse.org.uk/2017/feedback/JG

- Paper feedback forms also available from the Chair person

- This session is JG