IBM MQ for z/OS – Deep Dive on new features

Lyn Elkins – elkinsc@us.ibm.com

Timeline



- End of service dates
- Overview of what is new in MQ V8, both common and z/OS
- 64-bit bufferpools
- Multiple Cluster Transmission queues
- Performance enhancements for IBM Information Replicator
 - And applications that follow this model
- Z Hardware and Software Exploitation
- SMF enhancements

End of Service for old platforms and versions

- MQ V7.0.0 and V7.0.1 for multiplatforms EOM, EOS effective September 2015
 - V7.0 will have had more than 7 years of support
- MQ V7.0.1 for z/OS EOM, EOS effective September 2015
 - V7.0 .0 already out of service



If you are on the 7.0 versions and have not started upgrade planning – it's almost



IBM MQ V8 delivering best in class enterprise messaging

Platforms & 💙 Standards	Security	Scalability	System z exploitation
64-bit for all platforms	Userid authentication via OS & LDAP	Multiplexed client performance	64-bit buffer pools in MQ for z/OS means less paging, more performance
Multiple Cluster Transmit Queue on all platforms	User-based authorisation for Unix	Queue manager vertical scaling	Performance and capacity
Support for JMS 2.0	AMS for IBM i & z/OS	Publish/Subscribe improvements	Performance enhancements for IBM Information Replicator (QRep)
Improved support for .Net and WCF	DNS Hostnames in CHLAUTH records	Routed publish/subscribe	Exploit zEDC compression accelerator
SHA-2 for z, i & NSS	Multiple certificates per queue manager		SMF and shared queue enhancements

- End of service dates
- Overview of what is new in MQ V8, both common and z/OS
- 64-bit bufferpools
- Multiple Cluster Transmission queues
- Message Suppression
- Logging Changes
- Other changes
- Z Hardware and Software Exploitation
- SMF enhancements

MQ for z/OS: 64-bit bufferpools

64-bit buffer pools in MQ for z/OS

- Allows large numbers of messages to be cached before writing to pagesets
- Allows MQ to exploit the vast amount of storage on today's machines
- Improves performance of putting/getting messages by minimizing disk I/O
 - What's the BEST I/O?
- Minimizes administrative overhead of managing buffer pools
 - You can now have bufferpools 0-99
- Buffer pool LOCATION attribute says where it is located relative to the bar
 - BELOW: The default. Buffer pool is located below the bar in 31 bit storage
 - ABOVE: Buffer pool is located above the bar in 64 bit storage
 - This can be altered dynamically
- Storage can be pagefixed based on pageclass attribute

MQ for z/OS: 64-bit bufferpools

- Buffer pools above the bar can (theoretically) use up to 16 EB storage
- Increased maximum size of pool to 999,999,999 buffers
 Was 500,000
- Allows up to 100 buffer pools
 - Was 16
 - Can have a 1-1 page set buffer pool mapping
- The storage available above the bar
 - Set by MEMLIMIT or systems settings



Definition Changes

- To implement expanded bufferpools
 - New attributes added to the BUFFPOOL commands
 - LOCATION
 - BELOW default, the pool is taken from the 2G address space
 - ABOVE the pool is allocated above the 2G bar
 - PAGECLAS
 - 4KB each 4K page is pageable by the operating system
 - This is the only option for pools defined in BELOW the bar storage
 - 4KBFIXED each 4K page is fixed in memory
 - REPLACE/NOREPLACE
 - Should this definition override the what is held in the log of the queue manager?

```
BUFFERS( 20000 )
                                      LOCATION ( BELOW )
DEFINE BUFFPOOL
      PAGECLAS ( 4KB ) NOREPLACE
DEFINE BUFFPOOL (2) BUFFERS (50000)
                                      LOCATION( ABOVE ) +
      PAGECLAS ( 4KB )
                         REPLACE
DEFINE BUFFPOOL(3) BUFFERS(20000)
                                      LOCATION ( BELOW ) +
      PAGECLAS ( 4KB ) NOREPLACE
DEFINE BUFFPOOL( 10 ) BUFFERS( 1000 )
                                      LOCATION ( BELOW ) +
      PAGECLAS ( 4KB ) NOREPLACE
DEFINE BUFFPOOL( 11 ) BUFFERS( 1000 )
                                      LOCATION ( ABOVE ) +
      PAGECLAS (FIXED4KB) REPLACE
                                      LOCATION ( BELOW ) +
DEFINE BUFFPOOL( 12 ) BUFFERS( 1000 )
      PAGECLAS(4KB)
                      NOREPLACE
DEFINE BUFFPOOL(13) BUFFERS(1000)
                                      LOCATION( ABOVE ) +
      PAGECLAS ( 4KB )
                         REPLACE
```

MEMLIMIT

V7.1 MEMLIMIT

//QML1MSTR PROC

//PROCSTEP EXEC PGM=CSQYASCP,REGION=0M,MEMLIMIT=2G

//*

- Allocates 64-bit storage for MQ's use
 - Indexes
 - Security cache
 - SMDS buffers

V8.0 MEMLIMIT

//QML1MSTR PROC //PROCSTEP EXEC PGM=CSQYASCP,REGION=0M,MEMLIMIT=7<u>G</u> //*

Same storage areas and the above the bar bufferpools (sized for example in MQ V8 Redbook)

QREP and similar workload performance



QREP and similar workload performance



The benefit of sufficient buffers

- While QREP was one of the focus areas when testing above the bar bufferpools, any application that overfills a bufferpool will benefit.
- Another example illustrated in the upcoming MP1J, MQ for z/OS performance, is the effect of having sufficient buffers on a normal transmission queue.
- In this example a transmission queue has filled due to an unavailable receiver queue. Please note the following:
 - The below the bar bufferpool had 200,000 pages which is quite large for a buffer pools
 - ► The above the bar bufferpools had a **1,048,576 page (**4GB) buffer pool for the transmission queue.
 - Each test put the same number of messages, batching them at 200 messages per commit.
 - The large bufferpool peaked at 500,000 messages and did not have to write any to the pageset during processing.

Comparing time to drain a transmission queue



Comparing time - Notes

- With the 1,048,576 page bufferpool the 819,200 2Kb messages were transmitted in about 32 seconds
- The same volume took almost 165 seconds when there were only 200,000 pages available
 - Writing buffer pages to disk, due to the bufferpool filling is the difference.

- End of service dates
- Overview of what is new in MQ V8, both common and z/OS
- 64-bit bufferpools
- Multiple Cluster Transmission queues
- Message Suppression
- Logging Changes
- Other changes
- Z Hardware and Software Exploitation
- SMF enhancements

Multiple Cluster Transmission Queues

Harmonization with WMQ V7.5 Distributed

Advantages:

- Isolation Isolated transmission queues can be specified for particular cluster sender channels
- Providing separate service classes
 - A high priority transmission queue can be located on an above the bar bufferpool (large) and get better service

Disadvantage:

- Slightly increases the cost of the MQPUTs to a cluster queue
 - The overhead of supporting multiple cluster transmit queues is between 1 and 2 CPU microseconds. This does not change with message size or persistence.

Cost of MQPUT to clustered queue



- End of service dates
- Overview of what is new in MQ V8, both common and z/OS
- 64-bit bufferpools
- Multiple Cluster Transmission queues
- Message Suppression
- Logging Changes
- Other changes
- Z Hardware and Software Exploitation
- SMF enhancements

Message Suppression

This includes separate messages for client channels

- CSQX511I and CSQX512I for client connection to a SVRCONN
- CSQX500I and CSQX501I report other channel start/stop events

Message suppression

- Messages that are suppressed are not written to the JES log, so can reduce CPI
- Implemented via a new ZPARM, EXCLMSG
- Can be altered via a SET SYSTEM command
- Up to 16 messages can be suppressed
- Significant performance improvements have been observed noted when the SVRCONN messages are suppressed
 - Up to a 20% CPU reduction seen for poorly behaved clients

- End of service dates
- Overview of what is new in MQ V8, both common and z/OS
- 64-bit bufferpools
- Multiple Cluster Transmission queues
- Message Suppression
- Logging Changes
- Other changes
- Z Hardware and Software Exploitation
- SMF enhancements

Logging Changes

Log RBA constraint relief

- Already improved messages to warn of approaching RBA
- Now widening RBA field from 6 to 8 bytes
- At 100MB/sec this will now take about 5578 years to fill



Log capacity improvement

Log access changed to BSAM

- Archive logs can now hold 4G of data when written to DASD, active logs can be defined as 4G
- Note recommendations from KnowledgeCenter:
 - The maximum log size is 4 GB. You need to take care when defining a log of 4 GB, as the system might round up the number of records specified to a value that results in a log being greater than 4 GB.

When the queue manager reads such a log, the log is seen as being much smaller than it actually is, giving undesirable results; for example, very frequent log switches.

When defining a large log, you should check the HI-A-RBA value of the log using IDCAMS LISTCAT, to ensure that the log is strictly less than 4 GB (4 294 967 296 bytes).

- End of service dates
- Overview of what is new in MQ V8, both common and z/OS
- 64-bit bufferpools
- Multiple Cluster Transmission queues
- Message Suppression
- Logging Changes
- Other changes
- Z Hardware and Software Exploitation
- SMF enhancements

MQ for z/OS: Performance and Capacity

- Support for LP64 batch/RRS C applications
- Performance enhancements for IBM Information Replicator (QRep) and similar application patterns
 - Read-ahead and changes to deferred write processing allows MQ to increase sustainable data rates

General improvements to channel performance



- End of service dates
- Overview of what is new in MQ V8, both common and z/OS
- 64-bit bufferpools
- Multiple Cluster Transmission queues
- Message Suppression
- Logging Changes
- Other changes
- Z Hardware and Software Exploitation
- SMF enhancements

MQ for z/OS: Other features

Exploitation of zEDC compression accelerator

- SMF
 - We are awaiting performance numbers on this, but initial results looks promising!
- Channel compression (zlibfast option) can be useful when using SSL
 - In benchmark tests zEDC hardware compression can reduce the message costs by 80% vs software compression
 - Note that this can impact the dispatcher tasks, please see MP1J for additional information

MQ for z/OS: Other features

Support for Flash Express feature on zEC12 and zBC12 servers

- Improves resiliency of Coupling Facility with cost-effective standby capacity to handle overflow of shared queues
- Messages <63K fully held in Flash</p>
- Messages >63K have pointers in Flash, body in SMDS just as for traditional CF structures
- More information to come on this!

- End of service dates
- Overview of what is new in MQ V8, both common and z/OS
- 64-bit bufferpools
- Multiple Cluster Transmission queues
- Message Suppression
- Logging Changes
- Other changes
- Z Hardware and Software Exploitation
- SMF enhancements

SMF enhancements

SMF enhancements

- Channel initiator statistics and channel accounting allow better monitoring and tuning of channel behaviour these are discussed in detail
 - Tuesday at 10 am Sagewood
 - Wednesday at 1 pm Sagewood
- Correlation of MQ and CICS statistics

For more info ... Already available (draft)

















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

Capitalware's MQ Technical Conference v2.0.1.4

Questions???