z/TPF IBM MQ Support for 64-Bit

2023 TPF Users Group Conference
April 24-26, Dallas, TX
Communications Subcommittee

—

Cameron Doggett

Disclaimer

Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Problem Statement

IBM MQ messages on z/TPF continue to grow in both size and volume, while the memory used for them is limited to 31-bit memory (2 GB limit)

Pain Points

- Today, messages on memory queues are stored in system work blocks (SWBs), a critical system resource residing below the 2 GB bar
  - With limited 31-bit memory, more frequent I/O might result as the IBM MQ sweeper needs to free up SWB memory
  - Expanding the number of SWBs might not be possible or might mean sacrificing other system resources
  - Limits growth of applications and IBM MQ

Available Enhancements

The following enhancements provide some **relief to several pain points**:

- **z/TPF IBM MQ support for message compression on local queues— APAR PJ46078 (Dec 2020)**
  - Compresses messages on local memory queues by using hardware acceleration
  - Compressed messages consume less SWBs, and in turn, less I/O, CPU, and DASD usage
  - Frees up space below the 2 GB bar for the allocation of other system resources
- **z/TPF IBM MQ support for network compression — APAR PJ46137 (May 2021)**
  - Compresses messages sent across the network with IBM MQ sender and receiver channels
  - Reduces network bandwidth usage and processing time, especially for encrypted channels

z/TPF IBM MQ compression uses the Integrated Accelerator for zEDC (zEnterprise Data Compression) on the IBM®z15 (z15) or later.

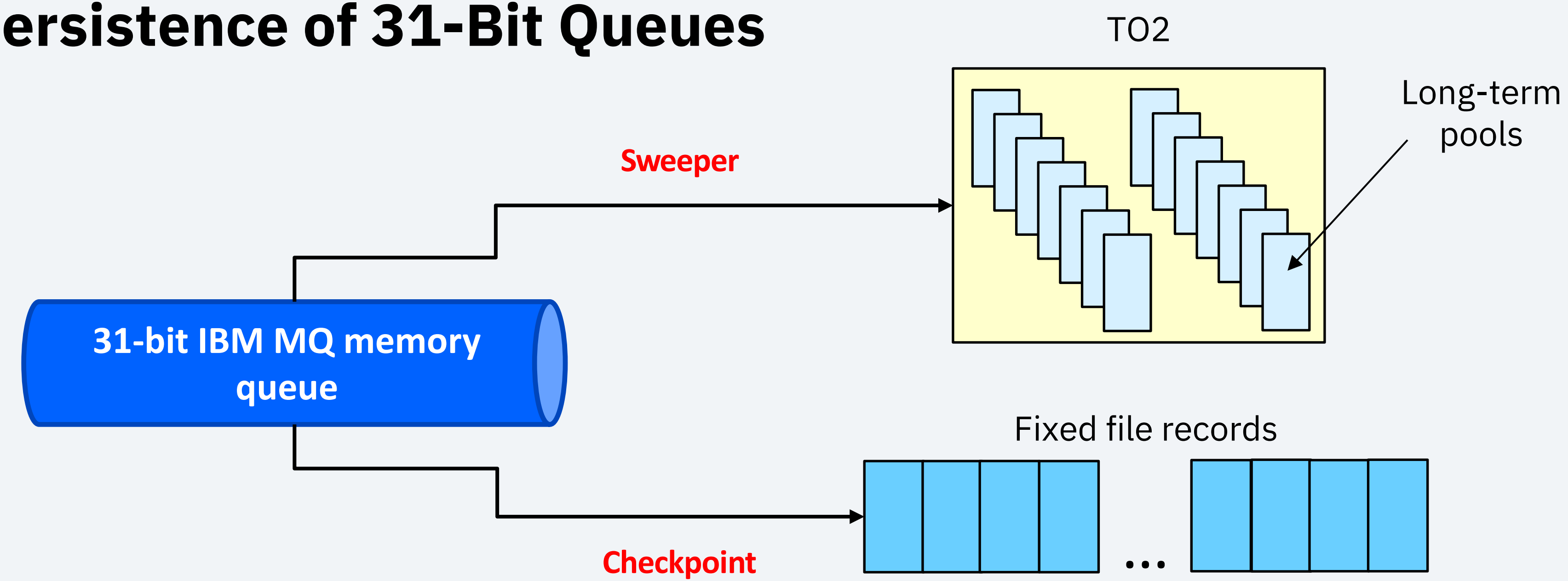
Value Statement

64-bit IBM MQ queues are the next step to improve **scalability**, designed to take full advantage of memory above the 2 GB bar while improving **performance**

64-Bit Queue Memory

- 64-bit queues will store their messages in a **new memory type dedicated to IBM MQ** known as **MQM** which resides above the 2 GB bar
  - Not possible for other activities on the system to deprive 64-bit queues of memory and vice versa
- MQM memory will be carved on IPL
  - Initialization occurs on demand as 4 KB MQM entries get used for the first time
- A new CTKA variable created to define the amount of MQM that z/TPF should allocate
  - The amount can be on the order of **terabytes**
- Decreased usage of SWBs means additional ECBs or IOBs can be allocated in their place

Persistence of 31-Bit Queues



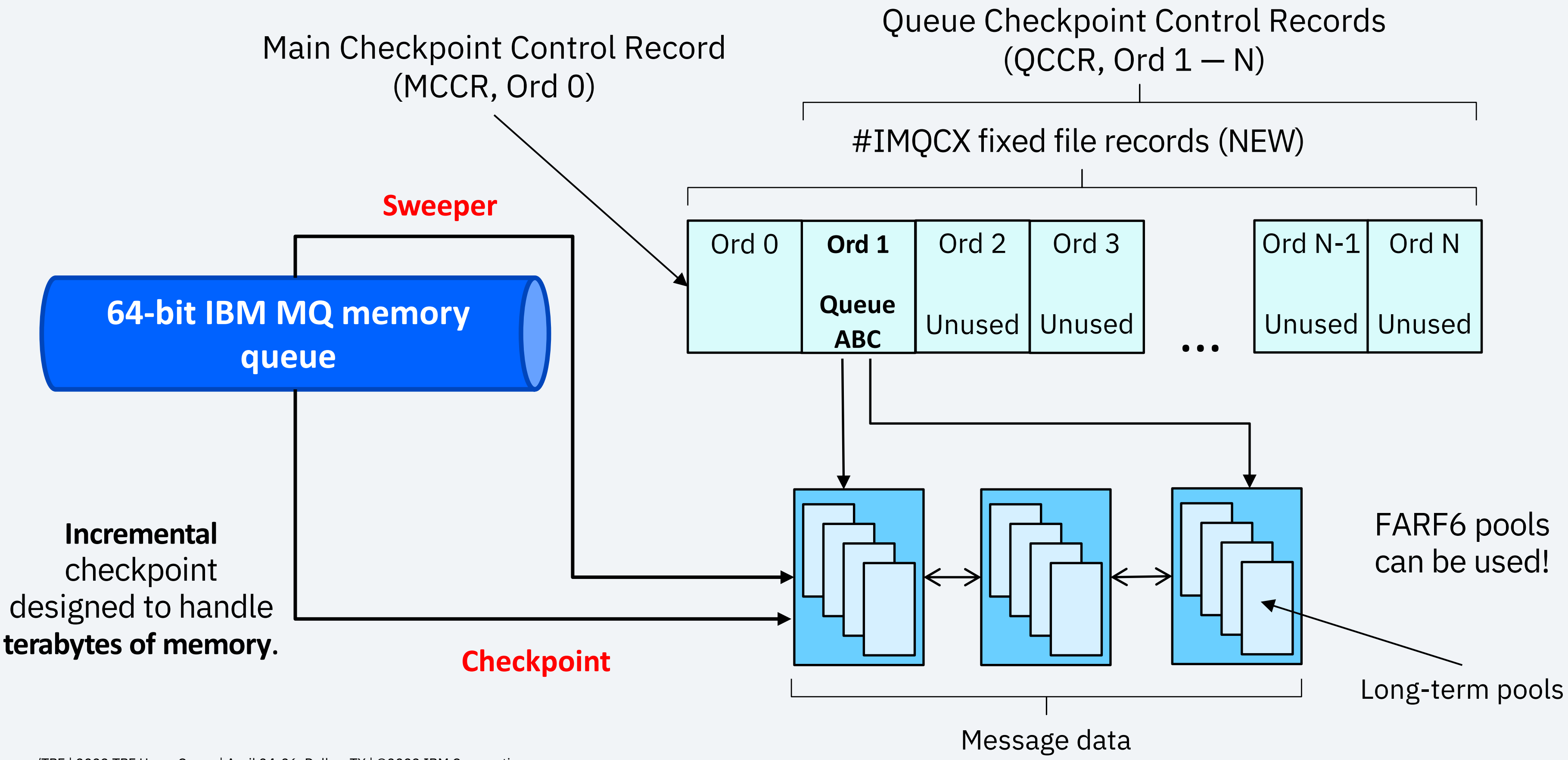
Full replace of queue content on each checkpoint.
Based on 2 GB of memory!

If the queue is stalled, persistent messages are repeatedly written to the checkpoint, and if the sweeper runs, those same messages are written again to a separate area.

Persistence of 64-Bit Queues

- Checkpoint **incrementally builds on the last checkpoint**, only additions and removals to persistent messages need to be recorded
  - Messages written to long-term pools anchored off a fixed file record assigned to the queue
  - This means a **message will be written to pools one time at most**
- Sweeper **leverages the checkpoint to persist messages to the same location**, increasing efficiency
  - Only writes messages **which had not already been persisted** before taking them out of memory
- The **sweeper is far less likely to run at all** since the amount of MQM can be very large — hundreds of gigabytes to even terabytes in size

Persistence of 64-Bit Queues (cont.)



Main Checkpoint Control Record
(MCCR, Ord 0)

Queue Checkpoint Control Records
(QCCR, Ord 1 — N)

#IMQCX fixed file records (NEW)

Ord 0	**Ord 1**	Ord 2	Ord 3		Ord N-1	Ord N
	Queue ABC	Unused	Unused	...	Unused	Unused

Sweeper

64-bit IBM MQ memory queue

Incremental checkpoint designed to handle **terabytes of memory.**

Checkpoint

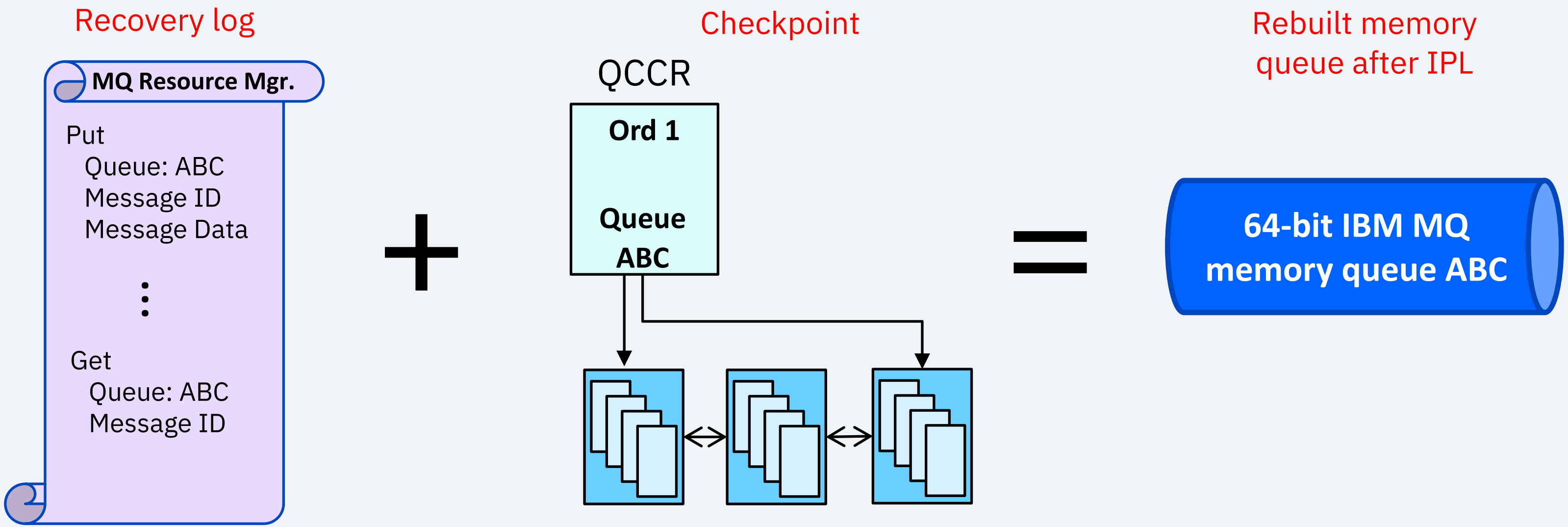
FARF6 pools can be used!

Long-term pools

Message data

Recovering a 64-Bit Queue

- The recovery log will be used by 64-bit queues similarly to 31-bit queues
- On IPL, the recovery log is merged with the checkpoint to reconstruct the latest view of the 64-bit queue



Creating a 64-Bit Queue

- Define a new queue
  - `ZMQSC DEFINE QL-’<Queue name>’ 64BIT-YES`
- Migrate an existing 31-bit queue
  - `ZMQSC ALTER QL-’<Queue name>’ 64BIT-YES`
  - Seamless migration to and from 64-bit without application changes or stopping the flow of messages

64-Bit Queue Considerations

The primary use case for 64-bit queues is for high volume, FIFO queues.

MQGETs must be sequential (FIFO)

- Browsing is strictly limited to reading messages in memory (not swept)
- Searching by message ID is not supported
- Searching by correlation ID is not supported

A 31-bit queue being used for any of the above **might not be a candidate** for migration to 64-bit.

Identifying Candidate Queues for Migration to 64-Bit

APAR PJ46881 (Nov 2022) adds a new display option to help identify queues using functionality restricted on 64-bit

C = Search by correlation ID
M = Search by message ID
B = Browsing

```
User:      ZMQSC DISPLAY QL-* GETDIAG

System:    CSMP0097I 14.39.34 CPU-B SS-BSS  SSU-HPN  IS-01
           MQSC0278I 14.39.34 LOCAL QUEUE MQGET DIAGNOSTICS DISPLAY
                                                              SEARCH/    XMITQ  LAST
           Queue Name                                         BROWSE     GET    PROG
           -------------------------------------------------- ---------- ------ -----
           CalculatorQueue                                    NO         N/A
           CalculatorSyncReplyQueue                           NO         N/A
           AsyncCalculatorQueue                               NO         N/A
           MY.MEMQ.1                                          NO         N/A
           MY.MEMQ.2                                          YES  (CMB) N/A    ABCD
           MY.XMITQ.1                                         NO         NO
END OF DISPLAY
```

Additional Value of 64-Bit IBM MQ — Maximum Message Size

- 31-bit queues on z/TPF limit IBM MQ messages to a maximum size of 4 MB
- **64-bit queues support message sizes up to 100 MB** (the architected IBM MQ limit)
  - When using large messages, compression is recommended
  - As today, larger persistent messages will likely necessitate a larger recovery log allocation

Summary

31-Bit Queues

- Messages below the 2 GB bar in SWBs
- Full replace checkpoint
- Sweeper writes messages to TPFCS (distinct area from checkpoint)
- Message size limited to 4 MB

64-Bit Queues

- Messages **above the 2 GB bar in MQM**
- **Incremental checkpoint**
- Efficient **sweeper leverages new checkpoint**
- Message size **increased to 100 MB**

We want sponsor users!

Our development cycle is driven by your feedback.

We are looking for sponsor users to assist in design and implementation, targeting the following personas:

- System Administrator
- IBM MQ Admin

Sponsor user calls are ongoing. The next call will be in May 2023. We are looking for beta users!

If you are interested in participating as a sponsor user, please contact:

John Muller (jfmuller@us.ibm.com)

Jamie Farmer (jvfarmer@us.ibm.com)

Cameron Doggett (camd.99@ibm.com)

Thank you