Session : 17895
Using and Abusing New Hardware
and the New V8 Features

Paul Kettley
PLM for Messaging on z/OS
paulk@uk.ibm.com
Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

<table>
<thead>
<tr>
<th>Trademark</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigInsights</td>
<td>DFSMSdss</td>
</tr>
<tr>
<td>BlueMix</td>
<td>DFSMSHsm</td>
</tr>
<tr>
<td>CICS*</td>
<td>DFSORT</td>
</tr>
<tr>
<td>COGNOS*</td>
<td>DS6000*</td>
</tr>
<tr>
<td>DB2*</td>
<td>DS8000*</td>
</tr>
<tr>
<td>DFSMScdf</td>
<td></td>
</tr>
<tr>
<td>IBM*</td>
<td></td>
</tr>
<tr>
<td>IMS</td>
<td></td>
</tr>
<tr>
<td>RACF*</td>
<td></td>
</tr>
<tr>
<td>System z10*</td>
<td></td>
</tr>
<tr>
<td>zEnterprise*</td>
<td></td>
</tr>
<tr>
<td>z/OS*</td>
<td></td>
</tr>
<tr>
<td>zSecure</td>
<td></td>
</tr>
<tr>
<td>z/VM*</td>
<td></td>
</tr>
<tr>
<td>Linux</td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td></td>
</tr>
<tr>
<td>UNIX</td>
<td></td>
</tr>
</tbody>
</table>

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the OpenStack website.

TEALEAF is a registered trademark of Tealeaf, an IBM Company.

Windows Server and the Windows logo are trademarks of the Microsoft group of countries.

Worklight is a trademark or registered trademark of Worklight, an IBM Company.

UNIX is a registered trademark of The Open Group in the United States and other countries.

* Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g. zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the Authorized Use Table for IBM Machines provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT"). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.
Agenda

- 64 Bit Buffer Pools
- 8 Byte Log Relative Byte Address (RBA)
- Channel Initiator (CHINIT) Statistics and Channel Accounting Data
- Storage Class Memory (SCM) (Flash Memory)
- Other Enhancements
64 Bit Buffer Pools
Buffer Pools: What we have today - 1

Q1
Page Set
4KB Page
Page Set
Q2
BufferPool
STGCLASS maps
PSID maps

Page Set
BufferPool
4KB Page
(x 16)
Buffer Pools: What we have today - 2

Queue Manager Address Space

- 16 EB
- 2 GB BAR
- Max 1.6GB for Buffer Pools
Buffer Pools: The Problems

- Not much space below the bar for buffer pools
  - Maximum 1.6GB, depending on common storage usage

- Put/Get to/from:
  - Buffer pool = 'memory' speed (fast)
  - Page set = 'disk' speed (slow)

- With less/small buffer pools, can spend a lot of time:
  - Putting pages from buffer pool into page set (to free buffers)
  - Getting pages from page set into buffer pool (to satisfy get requests)
  - This is detrimental to performance

- A maximum of 16 buffer pools
  - But, up to 100 page sets .. hence page sets must share buffer pools

- System programmers can spend a lot of time tuning:
  - Buffer pool sizes
  - Queue, buffer pool, and page set mappings
64 Bit Buffer Pools: The Solution

- Buffer pools above the bar.
  - Buffer pools can (theoretically) make use of up to 16 EB of storage

- More buffer pools
  - Up to 100 buffer pools
  - Can have 1-1 mapping between page set and buffer pool

- More buffers per pool
  - Above the bar
    - Up to 999,999,999 4K buffers per pool
  - Below the bar
    - Up to 500,000 4K buffers per pool
Buffer Pools: Using 64 bit storage

Queue Manager Address Space

Buffer Pool
Buffer Pool
Buffer Pool

DATA

Move above bar

Max 1.6GB for Buffer Pools

2 GB BAR

16 EB
64 Bit Buffer Pools: What has changed?

```
DEFINE BUFFPOOL(<id>)
  BUFFERS(<integer>)
  PAGECLAS(4KB/FIXED4KB)
  LOCATION(BELOW/ABOVE)
```

- BUFFPOOL id
  - 0 to 99

- BUFFERS integer
  - Up to 500,000 if LOCATION(BELOW)
  - Up to 999,999,999 if LOCATION(ABOVE)

- PAGECLAS can be:
  - 4KB, if LOCATION(BELOW)
  - FIXED4KB, if LOCATION(ABOVE)
    - permanent backing by real storage for life of Queue Manager
    - no need to programmatically page fix/unfix when doing I/O
    - better performance
    - ensure you have enough real storage available

- LOCATION
  - BELOW – buffer pool is below the bar (default)
  - ABOVE – buffer pool is above the bar
64 Bit Buffer Pools: Migration

- To use this function `OPMODE(NEWFUNC,800)` must be set
  - Otherwise behaviour is same as in version 7
  - Though, LOCATION(BELOW) is valid regardless of OPMODE
64 Bit Buffer Pools: Configuration

- **CSQINP1**
  - DEFINE BUFFPOOL(22) LOCATION(ABOVE) BUFFERS(1024000) REPLACE
  - DEFINE BUFFPOOL(88) BUFFERS(12000) REPLACE

- **CSQINP1 or dynamically**
  - DEFINE PSID(22) BUFFPOOL(22)

- **CSQINP2 or dynamically**
  - ALTER BUFFPOOL(88) LOC(ABOVE)

CSQP024I !MQ21 Request initiated for buffer pool 88
CSQ9022I !MQ21 CSQPALTB ' ALTER BUFFPOOL' NORMAL COMPLETION
CSQP023I !MQ21 Request completed for buffer pool 88, now has 12000 buffers
CSQP054I !MQ21 Buffer pool 88 is now located above the bar
64 Bit Buffer Pools: Migration

- Some messages have changed regardless of the value of OPMODE
  - Space has been added to allow for a larger number for buffers
  - PAGE CLASS and LOCATION can be seen on DISPLAY USAGE

```sql
CSQI010I !MQ21 Page set usage ...
<REMOVED>
End of page set report
CSQI065I !MQ21 Buffer pool attributes ...

<table>
<thead>
<tr>
<th>Buffer pool</th>
<th>Available buffers</th>
<th>Stealable buffers</th>
<th>Stealable percentage</th>
<th>Page class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1024</td>
<td>1000</td>
<td>99</td>
<td>4KB</td>
</tr>
<tr>
<td>22</td>
<td>1024000</td>
<td>234561</td>
<td>23</td>
<td>FIXED4KB</td>
</tr>
<tr>
<td>88</td>
<td>12000</td>
<td>1200</td>
<td>10</td>
<td>4KB</td>
</tr>
</tbody>
</table>

End of buffer pool attributes
```
**Single Requester per Queue:**

<table>
<thead>
<tr>
<th>Test</th>
<th>Transaction Rate (per second)</th>
<th>Transaction Cost (cpu microseconds)</th>
<th>LPAR %Busy</th>
<th>Channel Path %Busy</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-bit</td>
<td>232762</td>
<td>35.92</td>
<td>54%</td>
<td>56%</td>
</tr>
<tr>
<td>64-bit</td>
<td>235217</td>
<td>37.48</td>
<td>57%</td>
<td>57.4%</td>
</tr>
<tr>
<td>64-bit (enough buffers)</td>
<td>324213</td>
<td>38.12</td>
<td>83%</td>
<td>0.07%</td>
</tr>
<tr>
<td>64-bit (4GB per buffer pool)</td>
<td>341412</td>
<td>38.23</td>
<td>83%</td>
<td>0.08%</td>
</tr>
</tbody>
</table>

**2 Requesters per Queue:**

<table>
<thead>
<tr>
<th>Test</th>
<th>Transaction Rate (per second)</th>
<th>Transaction Cost (cpu microseconds)</th>
<th>LPAR %Busy</th>
<th>Channel Path %Busy</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-bit</td>
<td>149140</td>
<td>42.3</td>
<td>42%</td>
<td>75.4%</td>
</tr>
<tr>
<td>64-bit</td>
<td>145623</td>
<td>44.84</td>
<td>43.5%</td>
<td>75.9%</td>
</tr>
<tr>
<td>64-bit (enough buffers)</td>
<td>384062</td>
<td>40.85</td>
<td>99.59%</td>
<td>0.08%</td>
</tr>
<tr>
<td>64-bit (4GB per buffer pool)</td>
<td>370546</td>
<td>52.15</td>
<td>99.69%</td>
<td>0.07%</td>
</tr>
</tbody>
</table>

- 16 Central Processor LPAR
- Each transaction puts and gets a random message from a pre loaded queue.
- Second test requires a doubling in buffer pool size
8 Byte Log RBA
6 byte log RBA: The Problem

- MQ for z/OS V7.1 (or earlier):
  - Implements 6 byte Log RBA (Relative Byte Address)
  - Gives an RBA range of 0 to x'FFFFFFFFFFFF' (= 255TB)
  - Some customers reach this limit in 12 to 18 months
  - At 100MB/sec, log would be full in 1 month
Warning Messages and abend

- V7.1 Queue Managers do issue warning messages as log RBA gets high:

<table>
<thead>
<tr>
<th>Message Code</th>
<th>Log RBA Ranges</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSQI045I</td>
<td>x'700000000000', x'7100..', x'7200..' and x'7300..'</td>
</tr>
<tr>
<td>CSQI046E</td>
<td>x'740000000000', x'7500..', x'7600..' and x'7700..'</td>
</tr>
<tr>
<td>CSQI047E</td>
<td>x'780000000000', x'7900..', x'nn00..' and x'FF00..'</td>
</tr>
</tbody>
</table>

- APAR PM48299 (WebSphere MQ V7.0.1 and above) added messages:

<table>
<thead>
<tr>
<th>Message Code</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSQJ032E</td>
<td>Log RBA higher than x'F800000000000'</td>
</tr>
<tr>
<td>CSQJ031D</td>
<td>Confirm restart even though log RBA has passed x'FF80000000000'</td>
</tr>
</tbody>
</table>

- To prevent loss of data, Queue Managers with APAR PM48299 applied:

  Terminate with abend 00D10257 when log RBA reaches x'FFF8000000000'
6 byte log RBA: The Problem

- If end of the Log RBA range is reached:
  - You get an unplanned outage
    - Queue Manager terminates
    - Requires a “cold” start – a disruptive outage!
    - Potential for loss of persistent data

- To avoid an unplanned outage, at regular planned intervals:
  - Quiesce the Queue Manager
  - Recreate all QMGR logs and BSDS
  - Run CSQUTIL RESETPAGE
    - RESETs the LOG RBA in header of each page
  - Restart the Queue Manager
  - Some customers are happy to do this, but others are not!
8 byte log RBA: The Solution

- **Implement an 8 byte (64-bit) log RBA**
  - Gives an RBA range of 0 to \( \text{x'FFFFFFFFFFFFFFFFFFFFF} \)
    - Upper limit on logical log is now 64K times bigger
    - At 100MB/sec, log would be full in 5000+ years!!

- Format of BSDS and log records has changed to accommodate 8 byte RBAs
- URIDs are now 8 bytes long
- Utilities or applications that read the BSDS and Logs have been updated
- Console messages that contain the log RBA or URID have been updated
- Queue Manager uses 6 byte log RBAs until 8 byte log RBAs are enabled
Enabling 8 byte log RBAs

- **Procedure to enable 8 byte log RBAs:**
  - Stop the QMgr cleanly
  - Enable OPMODE(NEWFUNC,800)
    - In a QSG, new function mode is entered once all QMgrs have been started at NEWFUNC
  - Define new BSDSs in V1 format (these will be used to create the V2 format BSDSs)
    - V2 format BSDS contains more data than V1 format BSDS
    - Recommended space allocation is now RECORDS(850 60)
    - CSQ4BSDS sample job has been updated with this value
  - Run BSDS conversion utility (CSQJUCNV) to convert the V1 BSDS to V2
    - Creates a copy of a V1 format BSDS in V2 format
    - Checks all QSG QMgrs are running OPMODE(NEWFUNC,800)
  - Rename BSDSs so that V2 BSDSs are used during next restart of QMgr
  - Restart the QMgr
BSDS conversion utility (CSQJUCNV)

- **Parameters**
  - **NOQSG** (specify for a stand alone queue manager)
    - No OPMODE checks performed
  - **INQSG, qsgname, dsgname, db2ssid** (specify for a queue manager in a QSG)
    - Utility checks that all QMgrs in the QSG have been started at OPMODE(NEWFUNC,800) before allowing conversion to proceed

- **Example JCL:**

```bash
//CSQ4BCNV JOB
//CONVERT EXEC PGM=CSQJUCNV,REGION=32M,PARM=('INQSG,SQ13,db2,DB4A')
//STEPLIB DD DSN=ANTZ.MQ.V000.CUR.SCSQAUTH,DISP=SHR
// DD DSN=ANTZ.MQ.V000.CUR.SCSQANLE,DISP=SHR
// DD DSN=SYS2.DB2.V10.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=* 
//SYSUT1 DD DSN=VICY.MQ10.BSDS01,DISP=SHR
//SYSUT2 DD DSN=VICY.MQ10.BSDS02,DISP=SHR 
//SYSUT3 DD DSN=VICY.MQ10.NEW.BSDS01,DISP=OLD
//SYSUT4 DD DSN=VICY.MQ10.NEW.BSDS02,DISP=OLD
```
Externals – BSDS conversion utility (CSQJUCNV)

- Typical output

```plaintext
CSQJ445I CSQJUCNV BSDS CONVERSION UTILITY - 2014-06-04 15:02:48
CSQU526I CSQJUCNV Connected to DB2 DB4A
CSQU528I CSQJUCNVDisconnected from DB2 DB4A
CSQJ200I CSQJUCNV UTILITY PROCESSING COMPLETED SUCCESSFULLY
```
New message CSQJ034I at QMgr start

- **Issued during QMgr startup**
- **Indicates whether QMgr is running in 6 or 8 byte RBA mode**
  
  - `0000FFFFFFFFFFFFF` — 6 byte RBA mode
  
  - `FFFFFFFFFFFFFFFF` — 8 byte RBA mode

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Log ID</th>
<th>Message ID</th>
<th>Log Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.25.05</td>
<td>STC05120</td>
<td>CSQJ127I</td>
<td>!MQ4E</td>
<td>SYSTEM TIME STAMP FOR BSDS=2014-04-02 11:19:18.70</td>
</tr>
<tr>
<td>11.25.05</td>
<td>STC05120</td>
<td>CSQJ001I</td>
<td>!MQ4E</td>
<td>CURRENT COPY 1 ACTIVE LOG DATA SET IS 280</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DSNAME=VICY.MQ4E.LOGCOPY1.DS04, STARTRBA=000000000038F4000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ENDRBA=0000000003B0FFFF</td>
</tr>
<tr>
<td>11.25.05</td>
<td>STC05120</td>
<td>CSQJ099I</td>
<td>!MQ4E</td>
<td>LOG RECORDING TO COMMENCE WITH 281</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>STARTTRBA=00000000039AF000</td>
</tr>
<tr>
<td>11.25.05</td>
<td>STC05120</td>
<td>CSQJ034I</td>
<td>!MQ4E</td>
<td>CSQJW007 END OF LOG RBA RANGE IS 0000FFFFFFFFFFFFF</td>
</tr>
<tr>
<td>22.57.53</td>
<td>STC13100</td>
<td>CSQJ001I</td>
<td>!MQ08</td>
<td>CURRENT COPY 2 ACTIVE LOG DATA SET IS 810</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DSNAME=VICY.MQ08.LOGCOPY2.DS01, STARTRBA=000000002760000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ENDRBA=000000003B0FFFF</td>
</tr>
<tr>
<td>22.57.53</td>
<td>STC13100</td>
<td>CSQJ099I</td>
<td>!MQ08</td>
<td>LOG RECORDING TO COMMENCE WITH 811</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>STARTTRBA=00000002A8000</td>
</tr>
<tr>
<td>22.57.53</td>
<td>STC13100</td>
<td>CSQJ034I</td>
<td>!MQ08</td>
<td>CSQJW007 END OF LOG RBA RANGE IS FFFFFFFFFFFFFFFFFF</td>
</tr>
</tbody>
</table>
Warning message thresholds in V8

- The messages issued remain the same:

CSQI045I, CSQI046E, CSQI047E, CSQJ031D and CSQJ032E

- The thresholds at which the qmgr starts to issue warning messages are:
  - With 8 byte RBAs disabled:
    - HIGH RBA ADVICE: '0000F00000000000'x
    - HIGH RBA WARNING: '0000F80000000000'x
    - HIGH RBA CRITICAL: '0000FF8000000000'x
    - HIGH RBA ABEND: '0000FFF800000000'x

- The thresholds for issuing the CSQI messages have been increased to match those for the CSQJ messages:
  - With 8 byte RBAs enabled:
    - HIGH RBA8 ADVICE: 'FFFF800000000000'x
    - HIGH RBA8 WARNING: 'FFFFC00000000000'x
    - HIGH RBA8 CRITICAL: 'FFFFFC00000000000'x
    - HIGH RBA8 ABEND: 'FFFFFFC0000000000'x

- Once the threshold is exceeded, the frequency at which messages are issued remain the same
Updates to existing messages

- URIDs and RBAs in command outputs and console messages are 8 bytes:
  - Output of the DISPLAY CONN command looks like:

```plaintext
CSQM201I !MQ1P CSQMDRTC DIS CONN DETAILS
   CONN(CC15CF64B98D0001) EXTCONN(C3E2D8C3D4D8F1D74040404040404040)
   TYPE(CONN)
   QMURID(0000000000078599)
END CONN DETAILS
```

- Console Message **CSQE130I** (for CF structure recovery) looks like:

```plaintext
CSQE130I !MQ1P CSQERCF2 Recovery of structure
   APPLICATION1 started, using MQ1P log range from RBA=000000000007B663 to
   RBA=000000000007B6AB
```
Print log map (BSDS) utility (CSQJU004)

- Changed to always print 8 byte log RBA values
- Now displays BSDS version

<table>
<thead>
<tr>
<th>LOG MAP OF BSDS DATA SET COPY 1, DSN=VICY.MQ1P.BSDS01</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BSDS VERSION</strong> - 1</td>
</tr>
<tr>
<td><strong>UTILITY TIMESTAMP</strong> - 2014-05-23 13:40:09.56</td>
</tr>
<tr>
<td><strong>HIGHEST RBA OFFLOADED</strong> - 0000000000000000</td>
</tr>
</tbody>
</table>
Change log inventory utility (CSQJU003)

- Now accepts RBA values up to 16 characters long in parameters...

```
//CSQJU003 EXEC PGM=CSQJU003,REGION=0M
//SYSUT1 DD DISP=SHR,DSN=VICY.MQ1O.BSDS01
//SYSUT2 DD DISP=SHR,DSN=VICY.MQ1O.BSDS02
//SYSPRINT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
CHECKPT STARTRBA=00ABCD0000040000,ENDRBA=00ABCD0000045000,
TIME=20140201650000
```

- But, a value greater than 0000FFFFFFFFFFFFFFF results in error for V1 BSDS:

```diff
CSQJ443I CSQJU003 CHANGE LOG INVENTORY UTILITY - 2014-06-03 17:20:16
CHECKPT STARTRBA=00ABCD0000040000,ENDRBA=00ABCD0000045000,
TIME=20140201650000
CSQJ456E STARTRBA PARAMETER ARGUMENT EXCEEDS MAXIMUM VALUE FOR BSDS VERSION 1
CSQJ456E ENDRBA PARAMETER ARGUMENT EXCEEDS MAXIMUM VALUE FOR BSDS VERSION 1
CSQJ221I PREVIOUS ERROR CAUSED CHECKPT OPERATION TO BE BYPASSED
CSQJ201I CSQJU003 UTILITY PROCESSING WAS UNSUCCESSFUL
```
Changes to log print utility (CSQ1LOGP)

- CSQ1LOGP has been updated to handle 8 byte RBAs and URIDs
  - RBA of log records, and URIDs in log records, displayed in 8 byte format

- LRSN value now formatted as timestamp in log record header information

8 byte Log RBA

8 byte URID

Formatted LRSN
Changes to log print utility (CSQ1LOGP)

- New message CSQ1219I issued at the start of the output and whenever the format of the log record changes to indicate:
  - Whether the log records are in 6 or 8 byte RBA format
  - Whether the qmgr is in a QSG

```
CSQ1219I LOG RECORDS CONTAIN 6 BYTE RBA - QSG(NO)
CSQ1219I LOG RECORDS CONTAIN 6 BYTE RBA - QSG(YES)
CSQ1219I LOG RECORDS CONTAIN 8 BYTE RBA - QSG(NO)
CSQ1219I LOG RECORDS CONTAIN 8 BYTE RBA - QSG(YES)
```
Backwards Migration

- Backwards migration is NOT possible once NEWFUNC is enabled
- Cannot start a QMgr previously run in 8 byte RBA mode, in 6 byte RBA mode

00D92023 – 8 byte RBA log record read during restart in 6 byte log RBA mode
Channel Initiator (CHINIT) SMF Data
Chinit SMF: The Problem

- Prior to MQ V8 no SMF data for:
  - CHINIT address space
  - Channel activity

- Many customers have had to create their own ‘monitoring’ jobs
  - With periodic `DISPLAY CHSTATUS` commands

- Difficult to:
  - Manage historical data
  - Investigate performance issues
  - Perform capacity planning
CHINIT SMF: The Solution

- New SMF data for CHINIT address space:
  - **Channel Initiator Statistics** (SMF 115, SubType 231)
    - High level view of activity in CHINIT
      - Number of channels and TCB usage
      - Dispatchers, Adapters, DNS, SSL
      - Do I have spare capacity?
      - Do I need more or less dispatchers/adapters?
  - **Channel Accounting Data** (SMF 116, SubType 10)
    - Detailed view of individual channels
      - What work have channels been doing?
      - Which channels are being heavily utilised?
    - Controlled by STATCHL attribute on QMgr and Channel definition
Chinit SMF: The Solution

- Useful for
  - Monitoring
  - Capacity planning
  - Tuning

- Separate controls from queue manager SMF allows 'opt in'

- Supportpac MP1B updated to:
  - Format new data
Storage Class Memory (SCM) (Flash)
Shared Queues

- QMgrs must be in the same sysplex
- Create a Queue Sharing Group (QSG)
- Define shared queues in the Coupling Facility (CF)
CF Flash: Scenarios Planned Emergency Storage

CFSTRUCT OFFLOAD rules cause progressively smaller messages to be written to SMDS as the structure starts to fill.

Once 90% threshold is reached, the queue manager stores the minimum data per message (reference message) to squeeze as many message references as possible into the remaining CF storage.

Once at 90% threshold, CF Flash pre-staging algorithm also starts to move reference messages for new messages arriving into the CF structure into SCM (assume msgs are off the same priority). Older messages, which are likely to be got first are kept in the faster CF storage.

Note: Assume all msgs < 63KB
We want to keep high performance messages in the CF for most rapid access.

CFSTRUCT OFFLOAD are configured with special value '64k' to turn them off.

Once 90% threshold is reached, the CF Flash algorithm starts moving new messages to flash storage, keeping the faster 'real' storage for messages most likely to be gotten next.

As messages are got and deleted, the CF flash algorithm attempts to pre-stage the next messages from flash into the CFSTRUCT so they are rapidly available for MQGET.

In this scenario the flash storage acts like an extension to 'real' CFSTRUCT storage. However it will be consumed more rapidly since all message data is stored in it. Though, you could define a threshold to offload >16KB messages to SMDS if the CF structure is say 40% full. This would mean that only messages <=16KB ever get moved to flash storage.

Note: Assume all msgs < 63KB
<table>
<thead>
<tr>
<th>Scenario</th>
<th>Offload reason (Rule)</th>
<th>Msg Size</th>
<th>Total Msgs</th>
<th># in 'real' SMDS space</th>
<th>SMDS space</th>
<th># in 200 GB flash</th>
<th>Augmented (limit 30GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No SMDS No Flash</td>
<td>n/a</td>
<td>1kB</td>
<td>3M</td>
<td>3M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>n/a</td>
<td>4kB</td>
<td>900,000</td>
<td>900,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>n/a</td>
<td>16kB</td>
<td>250,000</td>
<td>250,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMDS No Flash</td>
<td>MQ 90%</td>
<td>1kB</td>
<td>3.2M</td>
<td>3.2M</td>
<td>800MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MQ 80%</td>
<td>4kB</td>
<td>1.8M</td>
<td>1.8M</td>
<td>5GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MQ 80%</td>
<td>16kB</td>
<td>1.3M</td>
<td>1.3M</td>
<td>20GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Emergency” Scenario</td>
<td>MQ 90%</td>
<td>1kB</td>
<td>190M</td>
<td>2M</td>
<td>270GB</td>
<td>190M</td>
<td>30GB</td>
</tr>
<tr>
<td></td>
<td>MQ 80%</td>
<td>4kB</td>
<td>190M</td>
<td>600,000</td>
<td>850GB</td>
<td>190M</td>
<td>30GB</td>
</tr>
<tr>
<td></td>
<td>MQ 80%</td>
<td>16kB</td>
<td>190M</td>
<td>150,000</td>
<td>3TB</td>
<td>190M</td>
<td>30GB</td>
</tr>
<tr>
<td>“Speed” Scenario</td>
<td>CF 90%</td>
<td>1kB</td>
<td>150M</td>
<td>2M</td>
<td>150M</td>
<td></td>
<td>26GB</td>
</tr>
<tr>
<td></td>
<td>CF 90%</td>
<td>4kB</td>
<td>48M</td>
<td>600,000</td>
<td>48M</td>
<td></td>
<td>8GB</td>
</tr>
<tr>
<td></td>
<td>CF 90%</td>
<td>16kB</td>
<td>12M</td>
<td>150,000</td>
<td>12M</td>
<td></td>
<td>2GB</td>
</tr>
</tbody>
</table>
Other Enhancements
64 bit application support

- 64 bit application support for C language
  - LP64 compile option
  - supported by cmqc.h

- Restricted environments
  - Batch, TSO, USS
  - CICS® and IMS® do not support 64 bit apps
  - WebSphere Application Server is already 64 bit

- Must use sidedeck & DLL, not stubs:
  - csqbmq2x (uncoordinated batch & USS)
  - csqbrr2x (RRS coordinated, srrcmit())
  - csqbri2x (RRS coordinated, MQCMIT)
Client Attachment Feature (CAF)

- Now shipped as part of the base MQ for z/OS product

- No longer chargeable on earlier releases of MQ
  - APAR available to enable functionality without installing CAF

- This means that client capability is available by default
  - Use CHLAUTH rules if you don't want Clients to connect to your QMgr
Other z/OS Items

- **Message suppression**
  - CSQ6SYSP / SET SYSTEM property **EXCLMSG**
  - Formalizes service parm to suppress Client channel start/stop messages
  - Extended to be generalized
    - Applicable for most MSTR and CHIN messages

- **DNS reverse (ip address → host name) lookup**
  - Queue Manager attribute **REVDNS(DISABLED/ENABLED)**
  - If DISABLED, prevents channel hangs if DNS infrastructure impacted
    - But CHLAUTH rules that use hostnames are not matched

- **zEnterprise Data Compression (zEDC) hardware exploitation**
  - Channel attribute **COMPMSG(ZLIBFAST)**
  - Need zEC12 GA2 + zEDC card
  - Can yield higher throughput & reduced CPU for SSL channels
MQ platform and product updates

- Split Cluster Transmit Queue availability in MQ for z/OS

- Advanced Message Security (AMS)
  - Better integrated into the base MQ for z/OS product
  - Offers improved performance and usability

- MFT has been updated to reduce reliance on USS
MQ V8.0 migration

- First class migration support from V7.0.1 and V7.1 (mixed QSG and fall-back capability)

- Install migration and compatibility PTFs
  - PI19721 has PTFs for either release

- Install QTYPE APAR for V8.0 for late breaking fixes
  - PI19991
Other bits and bytes .......

- JMS support in CICS
  - CICS V5.2 APAR PI32151
  - MQ APARs
    • for V71: JMS PI29770 (supercedes 7.1.0.6) or later CSD
    • for V8: JMS 8.0.0.2 PI33038 or later CSD + MQ base PI28482

- z13 Performance Expectations
  - Next chart

- Looking Forward ...........
  - Ease of Deployment of MQ
  - Increasing Size of Active Logs
Scalability on z13 with 16 processors

- On a 16-way LPAR, we saw an above LSPR improvement to a non-persistent workload.

This workload is seeing an 18% improvement in rate and transaction cost.
Summary

- LSPR predicts 11% improvement between zEC12 to z13
- Typically we have seen MQ performance measurements match or exceed this expectation for cost per transaction.
- Above 16-way we are seeing variable results which we are still investigating.
Objectives

- Rapid Provision/De-provision (create/destroy) of z/OS subsystems
  - For a Development or Test Environment
    - Automated
    - Reduced complexity
      - Preset properties
      - Minimum input parameters
  - MQ and others
- Little/no need for z/OS skills
  - Service consumers should not need to know JCL
- Workflow based
  - Invoke via REST APIs
  - Host in a Service Portal
IBM z/OS Management Facility (z/OSMF)

- Build XML workflow files in your favourite editor and export to USS
- Workflows we will look at:
  - provisionQMgr.xml
  - deprovisionQMgr.xml
  - Others
# Steps to provision a z/OS Queue Manager

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>JCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specify Queue Manager Criteria</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Define APF Load Library</td>
<td>CSQ4ZAPF.jcl</td>
</tr>
<tr>
<td>3</td>
<td>APF authorise the Load Library</td>
<td>CSQ4ZNAP.jcl</td>
</tr>
<tr>
<td>4</td>
<td>Generate MQ Initialization Parms (ZPARms)</td>
<td>CSQ4ZZPM.jcl</td>
</tr>
<tr>
<td>5</td>
<td>Define BSDS and Logs</td>
<td>CSQ4ZNBL.jcl</td>
</tr>
<tr>
<td>6</td>
<td>Define Pagesets</td>
<td>CSQ4ZNPG.jcl</td>
</tr>
<tr>
<td>7</td>
<td>Construct QMGR startup JCL</td>
<td>CSQ4ZMST.jcl</td>
</tr>
<tr>
<td>8</td>
<td>Use SETSSI to define subsystem to z/OS</td>
<td>CSQ4ZSSI.jcl</td>
</tr>
<tr>
<td>9</td>
<td>Construct CHINIT Startup JCL</td>
<td>CSQ4ZCHN.jcl</td>
</tr>
<tr>
<td>10</td>
<td>Construct DISX (CHINIT startup command)</td>
<td>CSQ4ZDSX.jcl</td>
</tr>
<tr>
<td>11</td>
<td>Disable subsystem security</td>
<td>CSQ4ZNSE.jcl</td>
</tr>
<tr>
<td>12</td>
<td>Start QMGR</td>
<td>CSQ4ZSQM.jcl</td>
</tr>
</tbody>
</table>

Further potential to merge and reduce steps
De-provision everything that was provisioned

# Steps to de-provision a z/OS Queue Manager

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>JCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stop Chinit</td>
<td>CSQ4ZTCN.jcl</td>
</tr>
<tr>
<td>2</td>
<td>Stop Queue Manager</td>
<td>CSQ4ZTQM.jcl</td>
</tr>
<tr>
<td>3</td>
<td>Delete any RACF security profile/settings</td>
<td>CSQ4ZDSE.jcl</td>
</tr>
<tr>
<td>4</td>
<td>Delete MQ specific Datasets</td>
<td>CSQ4ZZPM.jcl</td>
</tr>
</tbody>
</table>
Workflow based provisioning/De-provisioning

- **Useful in a cloud**
  - Rapidly provision subsystems to deploy applications to
  - Can introduce charging based on usage

- **Useful in Development/Test (beta) environments**
  - Can pool and share subsystem resources
  - Stand-up when needed
    - Define resources required by applications
  - Stand-down when finished
    - Delete all resources that were created

- **Useful for Automation**
  - Can automate subsystem configuration tasks
    - Enforce resource naming conventions
    - Reduces skill level required
Increasing Capacity of Active Logs

- **Active Logs today**
  - Max Size of an active log dataset is 4GB
  - Max Number of Active logs is 31 datasets
  - If there is site problem with archiving media
    - Active logs can fill a small number of hours

- **Increasing Active log capacity**
  - Increase size of active logs datasets above 4GB
  - Increase number of active logs from 31
  - Or do both ……

- **Initial solution**
  - Delivered as part of continuous delivery POC
    - Increase number of active logs to max of 310 datasets
    - Each dataset still limited to 4GB
IBM MQ V8 Features and Enhancements

Maximize your investment in IBM MQ

Discover new features that bring value to your business

Learn from scenarios with sample configurations

IBM.com/redbooks
Questions ?
Session : 17895
Using and Abusing New Hardware and the New V8 Features

Paul Kettley

PLM for Messaging on z/OS

paulk@uk.ibm.com