IMQ09 - IBM MQ V9 for z/OS Wildfire Workshop

L02 – Comparing Buffers Above and Below the Bar

Version V6.0

October 2018

Table of Contents

Lab Objectives

This lab has the following objectives:

1) To compare the use and costs of above and below the bar buffer pool allocation
2) To familiarize administrators with the new parameters

General Lab Information and Guidelines

- Information required to complete this exercise will be provided on a 'worksheet' prior to the start of this exercise. Refer to this worksheet for which user identity and password are to be used and for other values, for example:

  - ✓ This exercise requires using TSO user *USER1* on the *wg31.washington.ibm.com* system

  - ✓ As a reminder, if a value from your worksheet should be used, the values in the instructions will be in red rather than black.

  - ✓ ***Bold italicized*** text indicates values that need to be entered on a screen.

  - ✓ *Italicized* text indicates values that are constants or names that appear on a screen.

  - ✓ **Bold** text indicates the name of buttons or keyboard keys that need to be pressed.

- Please note that you should use the JCL data set USER1.BPBAR.JCL for this exercise.

Part 1 - Defining the queues

Use the MQ Explorer to define your queues for these exercises. If your explorer session has been shut down, please restart and connect to the QMZ1 queue manager.

For this test, two queues will need to be defined, one on each of the predefined storage classes. These storage classes, STGCLS10 and STGCLS11, have the attributes shown below:

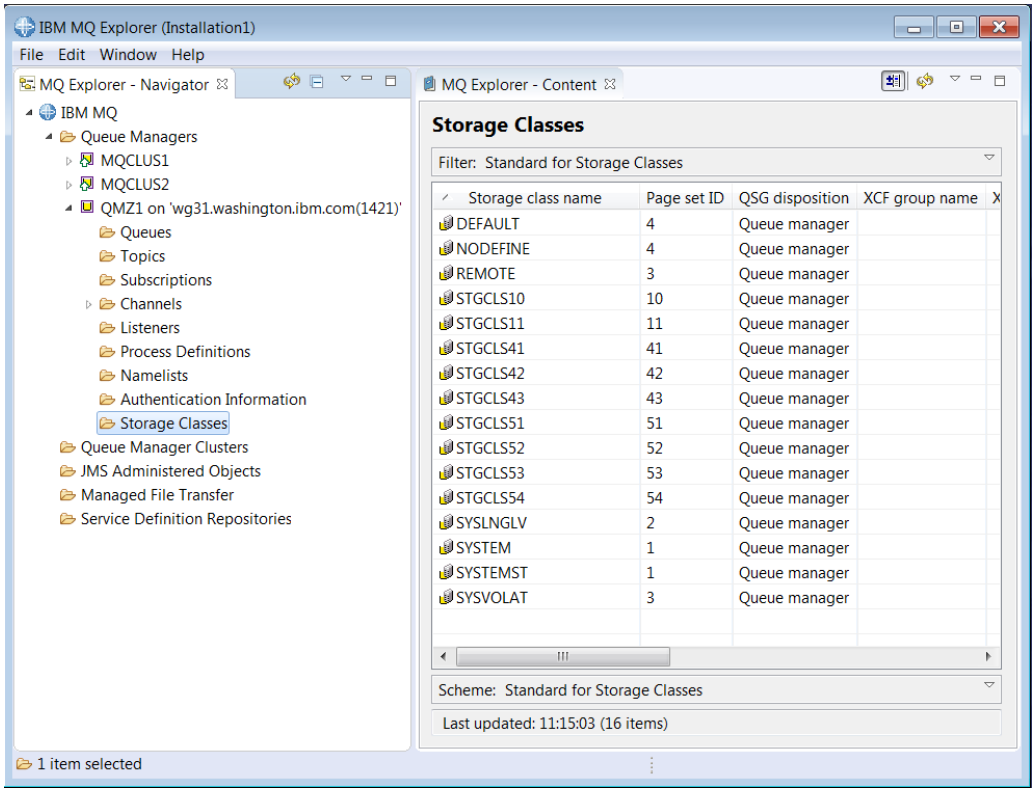
Storage Class	Buffer Pool	Location	Page Class
STGCLS10	10	Below	4KB
STGCLS11	11	Above	FIXED4KB

Verify the storage classes

____1. The storage classes can be displayed from the MQ Explorer, but that does not give all the critical bit of information about the location of the buffer pool the storage class uses.

> **Tech-Tip:** The storage class display has never shown the association with a buffer pool. There has also never been a Bufferpool display from the MQ Explorer, or a 'Display Usage' capability that would show the relationship between the STGCLASS and buffer pool. To get this information, you must use the commands in z/OS and review the JES log.

____2. From the queue manager resource list in the *Navigator* pane, click on the *Storage Classes* folder to display the currently defined storage classes (see below).



> **Tech-Tip:** MQ V8 increased the number of available buffer pool ranges from 0-15 (16) to 0-99 (100), the same number of page sets available. For customers concerned about performance and isolation of resource use, in particular those customers using QREP, defining a one-to-one relationship between the buffer pool and page set is recommended for application queues. That relationship helps identify problem area more quickly, and in some cases, move resources around to alleviate temporary performance and capacity issues more easily.

____3. In a TSO session use SDSF and go to the JES log and enter the command:

/QMZ1 DISPLAY USAGE

> **Tech-Tip:** *QMZ1* is the *command prefix* string of this queue manager. A command prefix string is used to direct commands to a queue manager. On this system the queue manager's *cpf* string is the same as the queue manager's name.

In the system log you should see output that looks something like this:

```
RESPONSE=S0W1
 CSQI010I QMZ1 Page set usage ...
   Page Buffer     Total    Unused  Persistent  NonPersist Expansion
    set   pool     pages     pages  data pages  data pages        count
 _    0      0      1078      1042          36           0 USER       0
 _    1      0      1078      1061          17           0 USER       0
 _    2      1      1078      1074           4           0 USER       0
 _    3      2      1078      1078           0           0 USER       0
 _    4      3      1078      1070           8           0 USER       0
 _   10     10      1078      1078           0           0 USER       0
 _   11     11      1078      1078           0           0 USER       0
 _   41     41     20157     20157           0           0 USER       0
 _   42     42     20157     20157           0           0 USER       0
 _   43     43     20157     20157           0           0 USER       0
 _   44     44     20157     20157           0           0 USER       0
 _   51     51     20157     20157           0           0 USER       0
 _   52     52     20157     20157           0           0 USER       0
 _   53     53     20157     20157           0           0 USER       0
 _   54     54     20157     20157           0           0 USER       0
  End of page set report
CSQI065I QMZ1 Buffer pool attributes ... 481
   Buffer  Available  Stealable   Stealable  Page      Location
     pool    buffers    buffers  percentage  class
 _      0      50000      49967          99  4KB       BELOW
 _      1      20000      19999          99  4KB       BELOW
 _      2      50000      49994          99  4KB       BELOW
 _      3      20000      19980          99  4KB       BELOW
 _     10       1000        999          99  4KB       BELOW
 _     11       1000        999          99  FIXED4KB  ABOVE
 _     41      10000       9999          99  4KB       BELOW
 _     42      10000       9999          99  4KB       BELOW
 _     43      10000       9999          99  4KB       BELOW
 _     44      10000       9999          99  4KB       BELOW
 _     51      50000      49999          99  FIXED4KB  ABOVE
 _     52      50000      49999          99  FIXED4KB  ABOVE
 _     53      50000      49999          99  FIXED4KB  ABOVE
 _     54      50000      49999          99  FIXED4KB  ABOVE
  End of buffer pool attributes
```

What may be different on the display is the number of stealable buffers and percentage, in addition to more buffer pools. If no queues have been defined or used in the buffer pool then these values will be different. Please also note the *Page class* and *Location* values, these were new for V8

Tech-Tip: From the MQ V9 Knowledge Center on the Location value:

LOCATION(LOC)(BELOW or ABOVE)
The LOCATION or LOC parameter specifies where the memory used by the specified buffer pool is located. LOCATION and LOC are synonyms and either, but not both, can be used. This memory location can be either ABOVE (64 bit) or BELOW (31 bit) the bar. Valid values for this parameter are BELOW or ABOVE, with BELOW being the default. ABOVE can only be specified if OPMODE(NEWFUNC, 800) is in effect. BELOW can be specified regardless of OPMODE(NEWFUNC, 800) being used and has the same effect as not specifying the LOCATION parameter.

When altering a buffer pool care should be taken to make sure there is sufficient storage available if increasing the number of buffers or changing the LOCATION value.

Tech-Tip: From the MQ V9 Knowledge Center on the Page class value:

PAGECLAS(4KB or FIXED4KB)
Optional parameter that describes the type of virtual storage pages used for backing the buffers in the buffer pool.

This attribute applies to all buffers in the buffer pool, including any that are added later as a result of using the ALTER BUFFPOOL command. The default value is 4 KB, which means that pageable 4 KB pages are used to back the buffers in the pool.

4 KB is the only valid value if the buffer pool has its location attribute set to BELOW. If the buffer pool has its LOCATION attribute set to ABOVE, it is also possible to specify FIXED4KB. This means that fixed 4 KB pages, which are permanently in real storage and will never be paged out to auxiliary storage, are used to back the buffers in the buffer pool.

FIXED4KB can only be specified if OPMODE(NEWFUNC, 800) is in effect whereas 4 KB can be specified regardless of the setting of OPMODE(NEWFUNC, 800).

The PAGECLAS attribute of a buffer pool can be altered at any time. However, the alteration only takes place when the buffer pool switches location from above the bar, to below the bar, or the other way round. Otherwise, the value is stored in the log of the queue manager and is applied when the queue manager next restarts.

When you specify PAGECLAS(FIXED4KB) the whole buffer pool is backed by page-fixed 4 KB pages, so ensure that there is sufficient real storage available on the LPAR. Otherwise, the queue manager might not start up, or other address spaces might be impacted; for more information, see Address space storage.

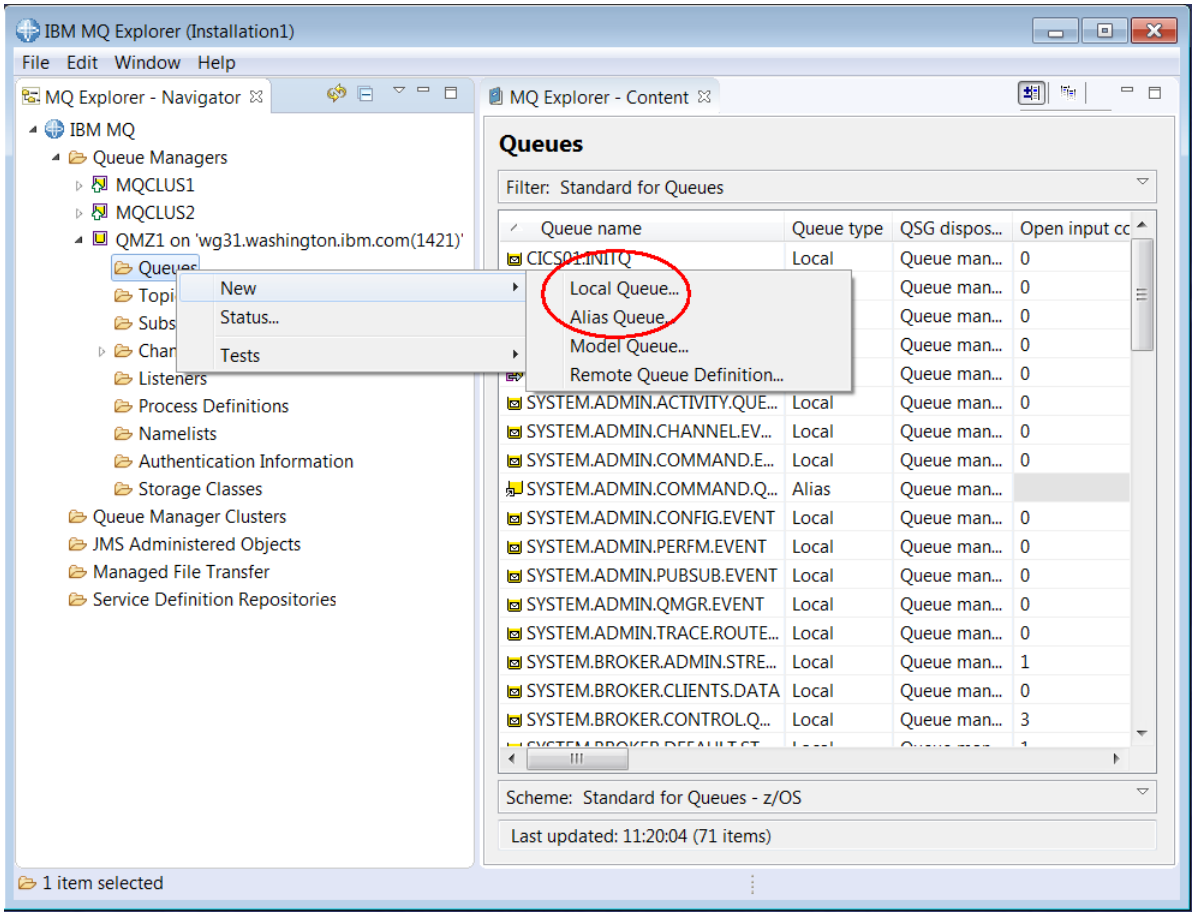
See WebSphere MQ Support Pac MP16: WebSphere MQ for z/OS - Capacity planning & tuning for advice on when to use the FIXED4KB value of the PAGECLAS attribute.

Please note that the buffer pools used for this exercise are not fixed, as we do not want to have real storage issues.

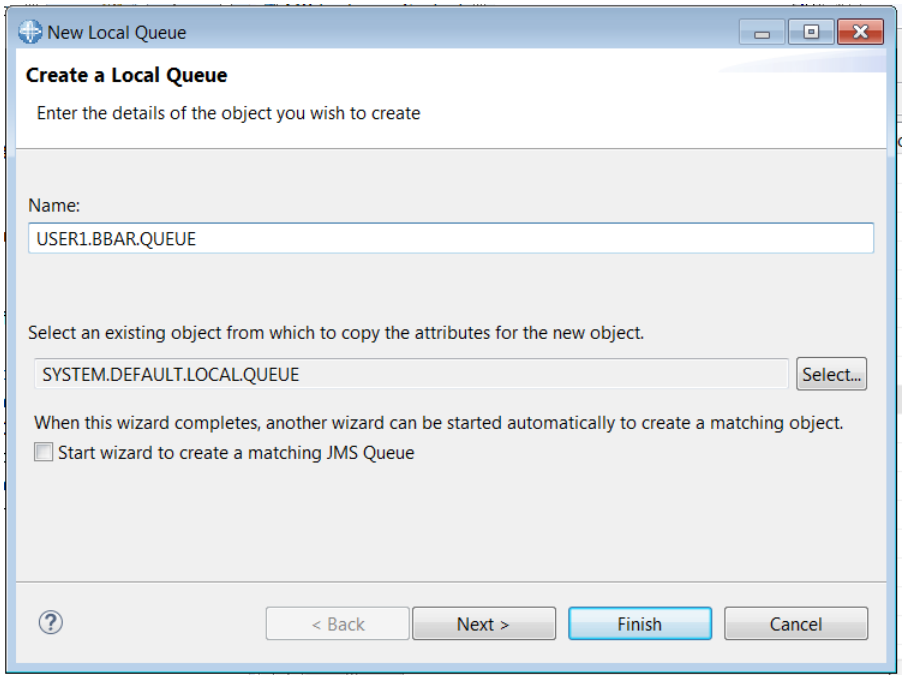
____4. Verify from the display that the even numbered buffer pool defined for your test is below the bar, and the odd numbered buffer pool is above. Please see the table on following the heading Part 1 - Defining the queues on page 4.

Define the new queues using these storage classes

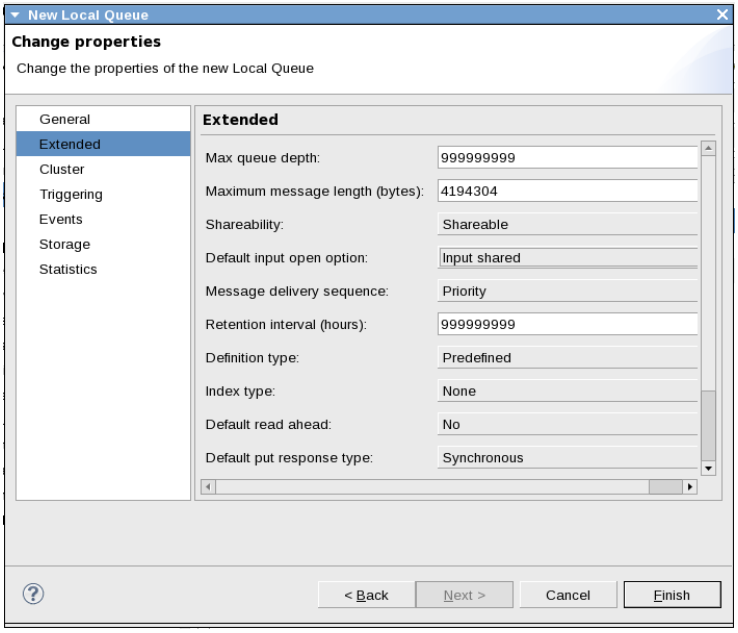
____1. To define the below the bar queue, right click on the *Queues* folder on the *MQ Explorer - Navigator* pane for your queue manager and select *New* and then *Local Queue* as shown below.



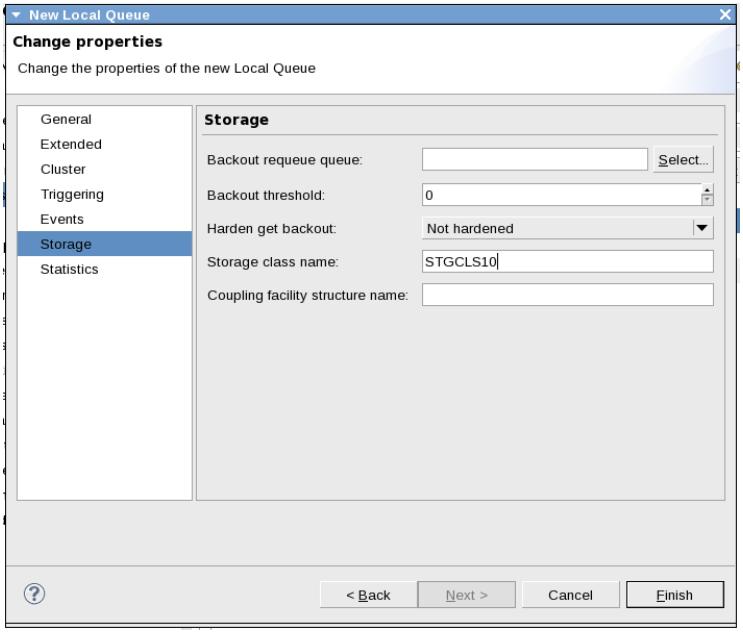
____2. Enter the queue name; please use all caps, such as ***USER1.BBAR.QUEUE***. Then click **Next** to continue.



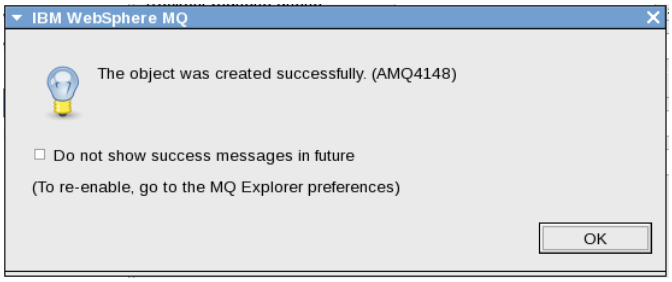
____3. Select the *Extended* tab and change the *Shareability* and *Default input open option* values to *Shareable* and *Input shared* respectively. Then select the *Storage* tab.



____4. Replace the Storage class name *DEFAULT* with the storage class name for the below the bar class e.g. ***STGCLS10***. Then click on the **Finish** button.



____5. The object was created successfully message should be displayed. You can turn it off, so it is not displayed again, or leave it on as you prefer.

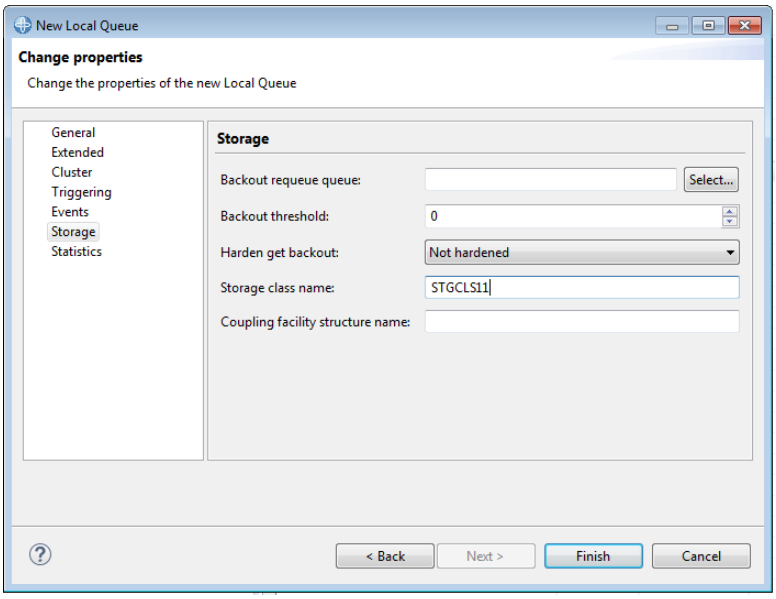


____6. To define the above the bar queue, right click again on the *Queues* folder on the *MQ Explorer - Navigator* pane for the queue manager and select *New* then *Local queue* as in Step 1 above.

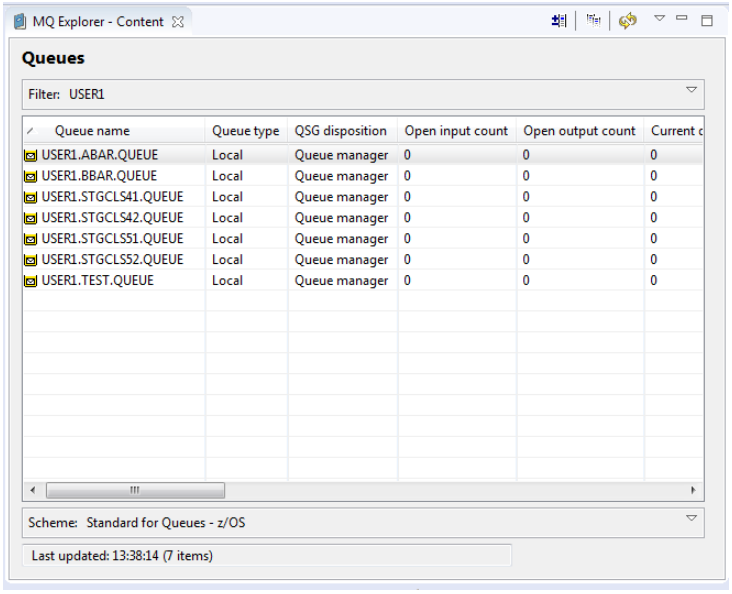
____7. Enter the queue name ***USER1.ABAR.QUEUE*** and then click **Next** button to continue.

____8. On the *Extended* tab, alter the *Shareability* and *Default input open option* values to *Shareable* and *Input shared* respectively. Then select the *Storage* tab.

____9. Replace the *Storage class name*, e.g. *DEFAULT, with* the storage class name for the above the bar, e.g. ***STGCLS11***. Then click on the **Finish** button.



____10. The queue list should now include the two newly defined queues.



Part 2 – Testing and comparing the buffer pools

The key advantage of above the bar buffer pools is the ability to hold many more messages in memory, avoiding I/O to the page sets. In this exercise we are not demonstrating that, but we are focusing on the comparison of runtime costs. Above the bar addressing can be slightly more expensive in CPU costs, but far less expensive than I/O!

For anyone with experience of DB2 moving buffers above the bar, the initial implementation was reportedly much more expensive than below the bar. MQ for z/OS development has learned from that experience and has the advantage of newer versions of the operating system.

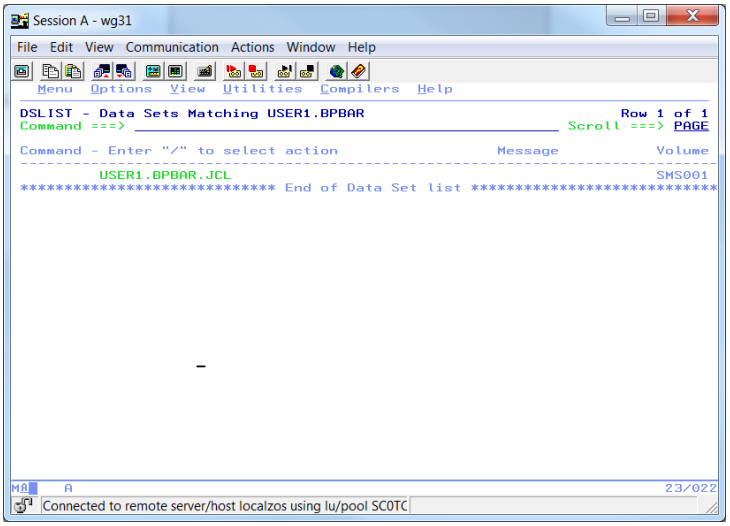
These tests are designed to compare the costs, and more importantly give some sample tests that customer can reproduce in their environments. It uses the OEMPUT program from the MP1B SupportPac to provide the sample programs, the older version of the MP16 SupportPac to evaluate the MQ SMF data (we will be using canned data, not running the jobs), and standard MQ commands.

Two sets of tests will be run. The first will compare the run characteristics of below and above buffer pools where no I/O must be done. The second will compare them when I/O does take place. We would caution everyone running these tests, the numbers presented were gathered when the environment was not being used for any other testing. Your results may (probably will!) vary. A major difference in performance characteristics is when z/OS paging occurs in the environment, a situation that we have observed from time to time.

Submitting the jobs

____1. In a TSO session enter ***=3.4*** in the command line to navigate to the *Data Set List Utility* panel and enter ***USER1.BP**** in the *Dsname level* field. Click the **ENTER** to display the list of data sets.

____2. Edit data set *USER1.BPBAR.JCL* by entering an ***E*** in the *Command* column as shown below.



____3. There are 4 jobs in this data set and each job has the four steps as described below:

Step Name	Program executed	Purpose
LOADQ	OEMPUTX from IP13	Loads 4K messages onto the specified queue
LOADUSE	CSQUTIL	Issues a DISPLAY USAGE command to show the current Buffer and pageset use
MGETQ	MGET from IP13	Reads the messages from the specified queue
MQGETUSE	CSQUTIL	Issues a DISPLAY USAGE command to show the current Buffer and pageset use

The 4 jobs are:

- *V8NOIO10* Loads 300 messages into a queue and should not use a page set
- *V8NOIO11* Loads 300 messages into a queue and should not use the page set
- *V8YSIO10* Loads 1200 messages into a queue and should use the page set
- *V8YSIO11* Loads 1200 messages into a queue and should use the page set

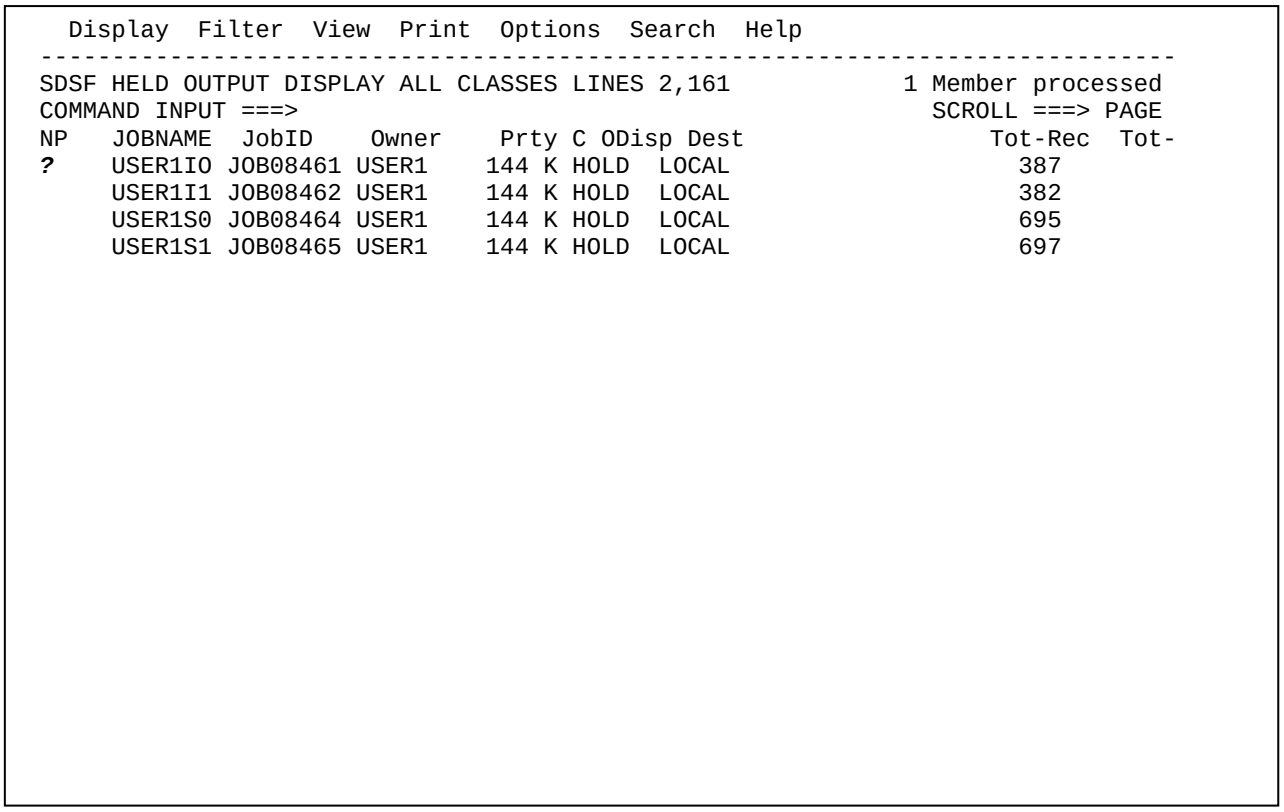
____4. Submit V8NOIO10. The does notify the submitter when it job is complete, but you may have to click **ENTER** a couple of times to get the notification.

```
16.57.21 JOB02190 $HASP165 USER1IO ENDED AT S0W1  MAXCC=0000 CN(INTERNAL)
```

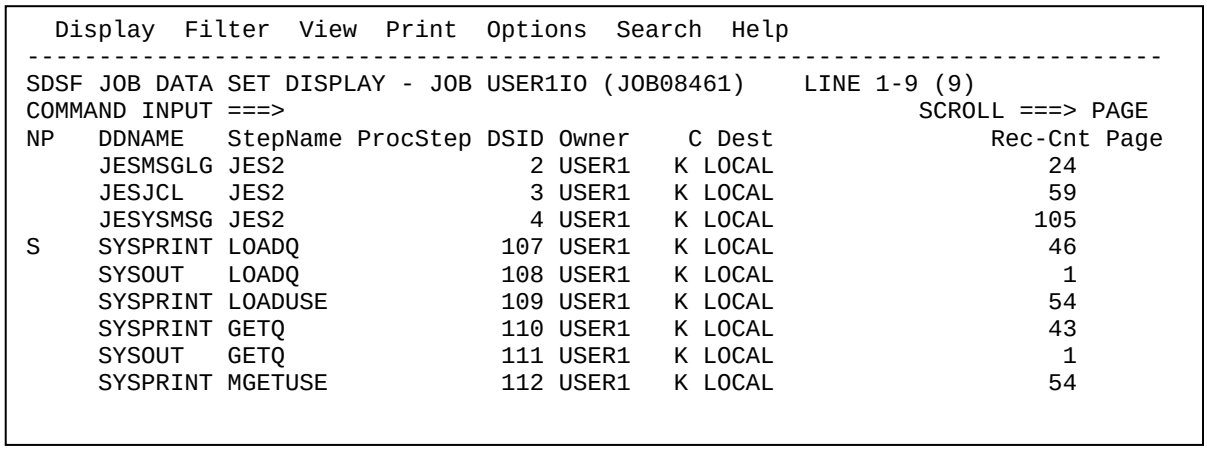
____5. Follow thes same steps for submitting the other JCL members, waiting for each job to complete before submitting the last one. This is very important, as the running two jobs against the same resource pool will contaminate the results.

Evaluating the findings

____1. Once all the jobs have run to completion, navigate to the SDSF status display panel. To do this, you can use ***=SDSF.H*** in the command line of any TSO screen. If you do not see your jobs in the list, the prefix probably needs to be changed. Use the SDSF command ***prefix USER1****.



____2. Enter a question mark (**?**); see above, to expand the first job run, the one with the lowest job ID number. Select the output for DDNAME *SYSPRINT* for step LOADQ and press **Enter**. If unfamiliar with the output of OEMPUT, please read the documentation on the test job from the MP1B SupportPac.



____3. Find the string "*Total Transactions"* as shown below.

```
Total Transactions  : 300
Elapsed Time        :    0.316 seconds
Application CPU Time:    0.231 seconds  (73.1%)
Transaction Rate    :  949.406 trans/sec
-----------------------------------------------------
Round trip per msg  :     1053 microseconds
Avg App CPU per msg :      769 microseconds
```

Make note of the following information from the test:

Total Transactions: _____
Transaction Rate: _____
Round trip per msg: _____
Avg App CPU per msg:_____

____4. Return to the output list and select the output for DDNAME *SYSPRINT* for step *LOADUSE*. This is the output from the 'DISPLAY USAGE' command. Find string *CSQI010I.*

```
CSQI010I QMZ1 Page set usage ...
  Page Buffer     Total    Unused  Persistent  NonPersist Expansion
   set   pool     pages     pages  data pages  data pages        count
_    0      0      1078      1021          57           0 USER       0
_    1      0      1078      1061          16           1 USER       0
_    2      1      1078      1074           4           0 USER       0
_    3      2      1078      1077           0           1 USER       0
_    4      3      3238      2614          17         607 USER       4
_   10     10      1078      1078           0           0 USER       0
_   11     11      1078      1078           0           0 USER       0
_   41     41     20157     20157           0           0 USER       0
_   42     42     20157     20157           0           0 USER       0
_   43     43     20157     20157           0           0 USER       0
_   44     44     20157     20157           0           0 USER       0
_   51     51     20157     20157           0           0 USER       0
_   52     52     20157     20157           0           0 USER       0
_   53     53     20157     20157           0           0 USER       0
_   54     54     20157     20157           0           0 USER       0
 End of page set report
CSQI065I QMZ1 Buffer pool attributes ...
  Buffer  Available  Stealable   Stealable  Page      Location
    pool    buffers    buffers  percentage  class
_      0      50000      49964          99  4KB       BELOW
_      1      20000      19999          99  4KB       BELOW
_      2      50000      49994          99  4KB       BELOW
_      3      20000      16777          83  4KB       BELOW
_     10       1000        999          99  4KB       BELOW
_     11       1000        393          39  FIXED4KB  ABOVE
_     41      10000       9999          99  4KB       BELOW
_     42      10000       9999          99  4KB       BELOW
_     43      10000       9999          99  4KB       BELOW
_     44      10000       9999          99  4KB       BELOW
_     51      50000      49999          99  FIXED4KB  ABOVE
_     52      50000      49999          99  FIXED4KB  ABOVE
_     53      50000      49999          99  FIXED4KB  ABOVE
_     54      50000      49999          99  FIXED4KB  ABOVE
 End of buffer pool attributes
```

What is the value of *Unused pages* in the page set your team is using? _____
Does it differ from the sample given (using page set 10)? If so, what might be the reason?
_____

How many stealable buffers remain in the buffer pool being used for this test after this job has run? (using bufferpool 10) _____

____5. Return to the output list and select the output for DDNAME *SYSPRINT* for the *GETQ* step. This is the output step that reads the messages from the queue.

```
About to get 99999999 messages from:
  Qname  = USER1.BBAR.QUEUE
  Qmgr   = QMZ1
Wait interval is 10 seconds
Quiet mode - Messages will not be printed
Buffer size is 1000 bytes
--------------------------------------------------------
Starting at 2015-12-10 19:35:04.312856

--------------------------------------------------------
Total Messages      : 300
Elapsed Time        : 0.147896 seconds
Message Rate        : 2028.45 msgs/sec
Average MQGET Time  : 0.033826 seconds
--------------------------------------------------------
Application CPU Time: 0.139200 seconds  (94.1%)
CPU per Message     : 0.4640 milliseconds
--------------------------------------------------------
```

Make note of the following information from the test:

Total Messages: _____
Message Rate: _____
CPU per msg:_____

____6. Return to the list of completed jobs, and expand the next one that ran. It should be the test for the above the bar buffer pool with no page set I/O. Selecting the *LOADQ SYSPRINT* output the queue name should be *USER1.ABAR.QUEUE*.

____7. Find the string '*Total Transactions'* as shown below.

```
Total Transactions  : 300
Elapsed Time        :    0.302 seconds
Application CPU Time:    0.217 seconds  (71.9%)
Transaction Rate    :  993.315 trans/sec
-----------------------------------------------------
Round trip per msg  :     1006 microseconds
Avg App CPU per msg :      723 microseconds
-----------------------------------------------------
```

Make note of the following information from the test:

Total Transactions: _____
Transaction Rate: _____
Round trip per msg: _____
Avg App CPU per msg:_____

____8. Compare the numbers with those from the below the bar test. The sample test captured the following:

```
Total Transactions  : 300
Elapsed Time        :    0.316 seconds
Application CPU Time:    0.231 seconds  (73.1%)
Transaction Rate    :  949.406 trans/sec
-----------------------------------------------------
Round trip per msg  :     1053 microseconds
Avg App CPU per msg :      769 microseconds
-----------------------------------------------------
```

Note that the difference observed by the OEMPUTX process is about 47 microseconds in the round trip time, and slightly lower transaction rate. The average CPU consumption was very close to the same.

____9. Return to the output list and select the output from DD *SYSPRINT* from the *LOADUSE* step. This is the output from the 'DISPLAY USAGE' command.

```
CSQI010I QMZ1 Page set usage ...
  Page Buffer     Total    Unused  Persistent  NonPersist Expansion
   set   pool     pages     pages  data pages  data pages        count
_    0      0      1078      1021          57           0 USER       0
_    1      0      1078      1061          16           1 USER       0
_    2      1      1078      1074           4           0 USER       0
_    3      2      1078      1077           0           1 USER       0
_    4      3      3238      3219          17           2 USER       4
_   10     10      1078      1078           0           0 USER       0
_   11     11      1078       473           0         605 USER       0
_   41     41     20157     20157           0           0 USER       0
_   42     42     20157     20157           0           0 USER       0
_   43     43     20157     20157           0           0 USER       0
_   44     44     20157     20157           0           0 USER       0
_   51     51     20157     20157           0           0 USER       0
_   52     52     20157     20157           0           0 USER       0
_   53     53     20157     20157           0           0 USER       0
_   54     54     20157     20157           0           0 USER       0
 End of page set report
CSQI065I QMZ1 Buffer pool attributes ...
  Buffer  Available  Stealable   Stealable  Page      Location
    pool    buffers    buffers  percentage  class
_      0      50000      49964          99  4KB       BELOW
_      1      20000      19999          99  4KB       BELOW
_      2      50000      49994          99  4KB       BELOW
_      3      20000      16777          83  4KB       BELOW
_     10       1000        999          99  4KB       BELOW
_     11       1000        393          39  FIXED4KB  ABOVE
_     41      10000       9999          99  4KB       BELOW
_     42      10000       9999          99  4KB       BELOW
_     43      10000       9999          99  4KB       BELOW
_     44      10000       9999          99  4KB       BELOW
_     51      50000      49999          99  FIXED4KB  ABOVE
_     52      50000      49999          99  FIXED4KB  ABOVE
_     53      50000      49999          99  FIXED4KB  ABOVE
_     54      50000      49999          99  FIXED4KB  ABOVE
 End of buffer pool attributes
```

What is the value of *Unused pages* in the page set your team is using? _____
Does it differ from the sample given (using page set 21)? If so, what might be the reason?

_____
Does the number of *Unused pages* in the job you ran differ from the below the bar test?
How many stealable buffers remain in the buffer pool being used for this test after this job has run? (using bufferpool 21) _____
Does the number of stealable buffers in the job you ran differ from the below the bar test?

____10. Return to the output list and select the output for DDNAME *SYSPRINT* for the *MGETQ* step. This is the output step that reads the messages from the queue.

```
About to get 99999999 messages from:
  Qname  = USER1.ABAR.QUEUE
  Qmgr   = QMZ1
Wait interval is 10 seconds
Quiet mode - Messages will not be printed
Buffer size is 1000 bytes
---------------------------------------------------------
Starting at 2015-12-10 19:35:26.898530

---------------------------------------------------------
Total Messages      : 300
Elapsed Time        : 0.147770 seconds
Message Rate        : 2030.18 msgs/sec
Average MQGET Time  : 0.033826 seconds
---------------------------------------------------------
Application CPU Time: 0.137900 seconds  (93.3%)
CPU per Message     : 0.4597 milliseconds
---------------------------------------------------------
Ending at 2015-12-10 19:35:37.046863.
```

Make note of the following information from the test:

Total Messages: _____
Message Rate: _____
CPU per msg:_____

____11. Compare the MGET result with the below the bar results. The sample given was:

```
About to get 99999999 messages from:
  Qname  = USER1.BBAR.QUEUE
  Qmgr   = QMZ1
Wait interval is 10 seconds
Quiet mode - Messages will not be printed
Buffer size is 1000 bytes
------------------------------------------------------
Starting at 2015-12-10 19:35:04.312856

------------------------------------------------------
Total Messages      : 300
Elapsed Time        : 0.147896 seconds
Message Rate        : 2028.45 msgs/sec
Average MQGET Time  : 0.033826 seconds
------------------------------------------------------
Application CPU Time: 0.139200 seconds  (94.1%)
CPU per Message     : 0.4640 milliseconds
------------------------------------------------------
```

How does the Message Rate vary from the below the bar test you ran:

_____

And the CPU per msg rate?:_____

Interestingly enough, in the sample test the get process from the above the bar buffer pool the performance was actually better.

> **Tech-Tip:** Your mileage will vary – it is critical that the use of above the bar buffer pools be tested in a production like environment

____12. Return to the list of run jobs and select the next one, below the bar with I/O.

____13. Again, examine the output from the *LOADQ* step. The results from the sample test look as follows:

```
-----------------------------------------------------
Total Transactions  : 1200
Elapsed Time        :    1.356 seconds
Application CPU Time:    0.943 seconds  (69.5%)
Transaction Rate    :  884.733 trans/sec
-----------------------------------------------------
Round trip per msg  :     1130 microseconds
Avg App CPU per msg :      785 microseconds
-----------------------------------------------------
```

____14. In this test there should be I/O to the page set. Compare the transaction rate, round trip and average CPU between this test and the below the bar BP with no I/O. In the samples the comparison would looks as follows:

Test Type	Transaction Rate	Roundtrip	Average CPU
BP below, no IO	949.406	1053	769
BP below, IO	884.733	1130	785

Were your results similar?

____15. Return to the output list, and examine the display usage results following the LOADQ (called the LOADUSE) step. Were there differences in the usage shown for either pageset or bufferpool from the earlier tests?

_____

____16. Return to the output list and examine the MGETQ output. The sample output shows the following:

```
About to get 99999999 messages from:
  Qname  = USER1.BBAR.QUEUE
  Qmgr   = QMZ1
Wait interval is 10 seconds
Quiet mode - Messages will not be printed
Buffer size is 1000 bytes
---------------------------------------------------
Starting at 2015-12-10 19:57:34.557462

---------------------------------------------------
Total Messages      : 1200
Elapsed Time        : 0.532363 seconds
Message Rate        : 2254.10 msgs/sec
Average MQGET Time  : 0.008777 seconds
---------------------------------------------------
Application CPU Time: 0.505000 seconds  (94.9%)
CPU per Message     : 0.4208 milliseconds
---------------------------------------------------
```

Compare the message rate and CPU per message values. In the samples test, we observed the following:

Test Type	Message Rate	Average MQGET time	Average CPU
BP below, no IO	2028.45	0.139200	0.4640
BP below, IO	2243.10	0.008777	0.4208 milliseconds

Tech-Tip: I/O can be expensive. The costs, both CPU and responsiveness, are very dependent on the underlying hardware and software that drives the I/O. I/O cannot be avoided on persistent messages, as those must be written to the logs, but it can be for non-persistent messages.

____17. Return to the list of completed jobs, and expand the next one that ran. It should be the test for the above the bar buffer pool with page set I/O. Selecting the *LOADQ SYSPRINT* output the queue name should be *USER1.ABAR.QUEUE*.

____18. Find the string "*Total Transactions*".

```
Total Transactions  : 1200
Elapsed Time        :    2.915 seconds
Application CPU Time:    1.018 seconds  (34.9%)
Transaction Rate    :  411.630 trans/sec
-----------------------------------------------------
Round trip per msg  :     2429 microseconds
Avg App CPU per msg :      848 microseconds
-----------------------------------------------------
```

Compare the rates between the above the bar with and without I/O. The same tests showed the following.

Test Type	Transaction Rate	Roundtrip	Average CPU per msg
BP above, no IO	949.406	1053	769 microseconds
BP above, IO	411.630	2429	848 microseconds

____19. Return to the output list, and examine the display usage results following the LOADQ (called the LOADUSE) step. Were there differences in the usage shown for either pageset or bufferpool?

____20. Return to the output list and examine the MGETQ output. The sample output shows the following:

```
About to get 99999999 messages from:
  Qname  = USER1.ABAR.QUEUE
  Qmgr   = QMZ1
Wait interval is 10 seconds
Quiet mode - Messages will not be printed
Buffer size is 1000 bytes
-----------------------------------------------------
Starting at 2015-12-10 20:12:33.268882

-----------------------------------------------------
Total Messages      : 1200
Elapsed Time        : 1.031317 seconds
Message Rate        : 1163.56 msgs/sec
Average MQGET Time  : 0.009193 seconds
-----------------------------------------------------
Application CPU Time: 0.772900 seconds  (74.9%)
CPU per Message     : 0.6441 milliseconds
-----------------------------------------------------
Ending at 2015-12-10 20:12:44.300884.
```

Compare the message rate and CPU per message values. In the samples test, we observed the following:

Test Type	Message Rate	Average MQGET time	CPU per message
BP above, no IO	2030.18	0.033826	0.4597 milliseconds
BP above, IO	1163.56	0.009193	0.6441 milliseconds

Conclusion

These simple tests illustrate that the runtime costs of using buffer pools defined above the bar. In the workshop environment we have seen that CPU costs of using buffers above the bar are not significantly higher than the buffers below the bar. However, this may not always be the case. If there is not enough real memory to support the above the bar buffers, z/OS paging will occur if the buffer pages are not fixed. If the buffer pages are fixed, the memory requirements could impact other application performance.

Customers should evaluate the costs in their environment to avoid ‘surprises’.