WebSphere MQ for z/OS Queue Usage Information When SMF116 Class 3 Cannot be used

By the IBM ATS WebSphere MQ team
Lyn Elkins – elkinsc@us.ibm.com
Eugene Kuehlthau – ekuehlth@us.ibm.com
Mitch Johnson – mitchj@us.ibm.com

Queue Usage information gathering, without using SMF116 class 3 data

For those customers that are unable to gather the SMF116 class 3 data, there is another option for gathering some queue usage information. This technique is not as comprehensive as the SMF116 class 3 data, it cannot be used to identify application code issues. It can be used to determine the pattern of use for specific queues. The usage pattern includes the volume of messages through the queues and the amount of time message remain on the queue.

Gathering the information requires multiple steps. First, the queues that need to be evaluated must be identified. Typically usage tracking is needed to research the over use of an individual pageset or buffer pool. In other cases tracking application usage is needed.

Steps to turn on monitoring:

1) Turn on performance events for the queue managers being evaluated. To see if performance events are turned on use the DISPLAY QMGR command

   +cpf DISPLAY QMGR PERFMEV

   Where the '+cpf is the command prefix that is defined for the queue manager. As an example, this series of commands were issued in the ATS Wildfire Workshop environment. In the sample, the command prefix is defined as the queue manager name (QML1). The slash shown is a directive to SDSF to submit the command.

```
SDSF OUTPUT DISPLAY QML1MSTR STC28813  DSID     2 LINE
COMMAND INPUT ===> /QML1 DISPLAY QMGR PERFMEV
```

   The response showed that performance events are turned off, as shown.

```
SDSF OUTPUT DISPLAY QML1MSTR STC28813  DSID     2
COMMAND INPUT ===> _
RESPONSE=MPX1
 CSQM201I QML1 CSQMDRTC  DISPLAY QMGR DETAILS
 QMNAME(QML1)
 PERFMEV(DISABLED)
  END QMGR DETAILS
```

   The following command turns on performance events

   +cpf ALTER QMGR PERFMEV(ENABLED)

   Again from the ATS Wildfire Workshop environment the following examples were captured.

```
SDSF OUTPUT DISPLAY QML1MSTR STC28813  DSID     2 LINE
COMMAND INPUT ===> /QML1 ALTER QMGR PERFMEV(ENABLED)
```

The response was:

```
 SDSF OUTPUT DISPLAY QML1MSTR STC28813  DSID     2 LINE 1,773   COLUMNS 82- 133
 COMMAND INPUT ===> _                                          SCROLL ===> CSR
10.37.29 STC28813  CSQ9022I QML1 CSQMAMMS ' ALTER QMGR' NORMAL COMPLETION
```

Repeating the display command above, the results now show the performance events turned on. The response from the Wildfire environment now looks as shown:

```
SDSF OUTPUT DISPLAY QML1MSTR STC28813  DSID
COMMAND INPUT ===> _
RESPONSE=MPX1
 CSQM201I QML1 CSQMDRTC  DISPLAY QMGR DETAILS
 QMNAME(QML1)
 PERFMEV(ENABLED)
  END QMGR DETAILS
```

2) Identify the queues to be monitored. If the investigation is into the overuse of a buffer pool or pageset, the queues that are defined to the pageset and buffer pool must be identified.

a. Review the queue manager JES log for the queue manager, looking for CSQP007I messages. These messages identify the pageset buffer pool association, as shown:

```
CSQP007I·QML2·Page·set·0·uses·buffer·pool·0¶
CSQP007I·QML2·Page·set·1·uses·buffer·pool·0¶
CSQP007I·QML2·Page·set·2·uses·buffer·pool·1¶
CSQP007I·QML2·Page·set·3·uses·buffer·pool·2¶
CSQP007I·QML2·Page·set·4·uses·buffer·pool·3¶
CSQP007I·QML2·Page·set·5·uses·buffer·pool·3¶
```

b. To identify the queues that are defined to a specific pageset and check the monitoring setting for those queues, a simple display command is used. In the sample shown, replace the '?' with the pageset ID, and the +cpf with the command prefix string.

++cpf DISPLAY QUEUE(*) PSID(?) MONQ

Using an ATS environment partial results of the display command are shown:

```
CSQM201I QML1 CSQMDRTC  DISPLAY QUEUE DETAILS  008
QUEUE(TEAMXX.PERF.EVENT)
TYPE(QLOCAL)
QSGDISP(QMGR)
PSID(4)
MONQ(QMGR)
 END QUEUE DETAILS
```

c. The list of queues that are returned from this display includes queues that are monitoring candidates. In some instances. 'problem queue' identification is possible from a review of this list. For example, if highly active application queues and the cluster transmission queue (SYSTEM.CLUSTER.TRANSMIT.QUEUE) are identified on the same pageset and buffer pool, then by moving the application queues to an underutilized buffer pool may resolve the problems without further investigation. Another example is that if a highly active application has both the request and reply queue defined on the same pageset, moving the reply queue may resolve the problem.

NOTE: If there are multiple pagesets that are assigned to a single buffer pool, the display command should be repeated for each pageset. The complete queue list is complied from all the responses.

3) Queue monitoring is controlled at two levels, queue manager and queue. Monitoring has three settings, High, Medium, and Low. For most investigation work a 'Medium' setting is sufficient.

Turning on monitoring at the queue manager level is simple, but enables monitoring for ALL queues using the default setting of MONQ(QMGR). Monitoring will be turned on for queues that you may not be interested in.

Verify the queue manager setting, use the display commands as illustrated

+cpf DISPLAY QMGR MONQ

In the ATS environment, the command resulted in:

```
RESPONSE=MPX1
 CSQM201I QML1 CSQMDRTC  DISPLAY QMGR DETAILS
 QMNAME(QML1)
 MONQ(OFF)
  END QMGR DETAILS
```

If all the queues in the previous step have monitoring set to MONQ(QMGR), issuing the alter QMGR command will turn on monitoring for all the queues.

+cpf ALTER QMGR MONQ(MEDIUM)

Continuing with the previous examples, in the ATS environment the command to turn on monitoring at the queue manager level is:

```
COMMAND INPUT ===> /QML1 ALTER QMGR MONQ(MEDIUM)
```

A display of the queue manager's monitoring now looks as shown:

```
RESPONSE=MPX1
 CSQM201I QML1 CSQMDRTC  DISPLAY QMGR DETAILS
 QMNAME(QML1)
 MONQ(MEDIUM)
  END QMGR DETAILS
```

4) If any queues from the pagesets under examination have monitoring turned off, SYSTEM.* queues frequently will, monitoring should be turned on. Not having the information from all the queues can nullify the findings. Monitoring can be turned on for individual queues by issuing the ALTER queue command.

   For example, the SYSTEM.CLUSTER.TRANSMIT.QUEUE has monitoring turned off:

```
CSQM201I QML1 CSQMDRTC  DISPLAY QUEUE DETAILS  861
QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
TYPE(QLOCAL)
QSGDISP(QMGR)
MONQ(OFF)
 END QUEUE DETAILS
```

   To turn on monitoring, an ALTER queue command is used:

   +cpf ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) MONQ(QMGR)

5) After monitoring is turned on, the information about queue usage can be gathered. This requires an automation tool to issue the 'DISPLAY QSTATUS' and 'RESET QSTATS' commands periodically. If the issues being investigated are predictable, scheduling the commands at 5 minute intervals beginning an hour before the problem typically occurs until an hour after it usually clears is sufficient. If there is no predictable pattern, then these commands should be issued at least every 15 minutes for at least a day. The commands to be issued are:
   +cpf DISPLAY QSTATUS(qname or generics) ALL
   +cpf RESET QSTATS(qname or generic)

6) Following the data collection, the information must be extracted from the JES message log for the queue manager. The results from the DISPLAY QSTATUS command contain the following:

```
CSQM201I QML1 CSQMDRTC  DISPLAY QSTATUS DETAILS  216
QSTATUS(TEAMXX.PERF.EVENT)
TYPE(QUEUE)
OPPROCS(0)
IPPROCS(0)
CURDEPTH(3)
UNCOM(NO)
MONQ(MEDIUM)
QTIME(418225201,418225201)
MSGAGE(160649)
LPUTDATE(2012-02-27)
LPUTTIME(16.04.11)
LGETDATE(2012-02-27)
LGETTIME(10.59.04)
QSGDISP(QMGR)
 END QSTATUS DETAILS
```

All the fields are important, the ones that are typically vital in problem determination includes the number of processes that have the queue open (OPPROCS and IPPROCS), the current depth, the QTIME, and the message age.

7) The results from the RESET QSTATS command contain the following:

```
12.48.30 STC28813  CSQM201I QML1 CSQMDRTC  RESET QSTATS DETAILS  223
   223             QSTATS(TEAMXX.PERF.EVENT)
   223             QSGDISP(QMGR)
   223             RESETINT(179749)
   223             HIQDEPTH(3)
   223             MSGSIN(3)
   223             MSGSOUT(3)
   223              END QSTATS DETAILS
```

The QSTATS provides information on the message flow through the queue – the number of messages put and gotten in the interval. While all the fields are important, the ones that are typically vital in problem determination include the number of processes that have the queue open (OPPROCS and IPPROCS), the current depth, the QTIME, and the message age.

8) When trying to identify highly active queues, chart the number of messages flowing through queues. When trying to reduce buffer pool stress, charting the number of messages flowing through the queues by the buffer pool, and charting the time messages spend on the queue will help plan proper queue positioning and whether buffer pools are sized appropriately. One piece of information that is missing is the message size, that is only available on the SMF116 class 3 data.

For more information about monitoring WebSphere MQ, see the WebSphere MQ InfoCenter. It cans be found at:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r1/index.jsp