The Dark Arts of MQ SMF Evaluation

Lyn Elkins
elkinsc@us.ibm.com
Session # 15024
10 March 2014
Code!
The witch trial – MQ is broken!
Why is this session first?

• The SFM data, especially the SMF115 data is more interesting after you have learned about the internals.
  • 1:30 on Wednesday
• It’s interesting after you have learned more about problem determination
  • 9:30 on Thursday
• So as to WHY this is the first MQ session in the agenda
  • It just works out that way!
Agenda

• Review of SMF 115 and SMF 116 class 3 data
• Hunting down the culprit
  • SMF115 Data
    • Bufferpool behaving badly
    • Volume growth
    • Log manager getting cranky
    • Other SMF115 data of interest
  • SMF116 Data
    • What queues are being used and how?
    • Can I find out which queues are the most active?
    • Pulling the data for one CICS transaction or batch job
    • Long running tasks
• Summary

Complete your session evaluations online at www.SHARE.org/AnaheimEval
Objectives

• This session is to delve a bit deeper into how the ATS team use the SMF data to find transaction and WMQ problems, based on situations we’ve tried to resolve.
• It will bore you to death.
Review of SMF115

• The SMF 115 data is the statistical information produced by a WMQ for z/OS queue manager.
  • Primarily used to track major trends and resolve performance problems with the queue manager
  • Very lightweight
  • Broken down into the major ‘managers’ within WMQ
• The ‘old’ MP1B provides several views into the data:
  • MQ1150 – detailed SMF115 report
  • MQCSMF – extracts specific information from SMF115 and 116 in a column format
    • Particularly useful for building spreadsheets
• The ‘new’ MP1B provides two views of the data
  • Report from for each manager
  • Comma separated values
Review of SMF116 – Class 3 data

• The SMF 116 data is the accounting information produced by a WMQ for z/OS queue manager.
  • Primarily used to determine what is going on within WMQ workload
  • Heavyweight
  • Broken down into the transactions within WMQ
• The old MP1B provides several views into the data:
  • MQ1160 – prints the SMF116 class 1 report
  • MQ116S – prints the detailed SMF116 class 3 report, including the queue information
  • MQCSMF – extracts specific information from SMF115 and 116 in a column format
    • Particularly useful for building spreadsheets
• The new MP1B provides:
  • The ‘TASK’ output
    • Somewhat like the MQ116S report
    • I am currently writing a paper on the differences/similarities
  • Other files, much like the ‘old’ MQCSMF output
Finding the problem
SMF 115 data – Hunting down the culprit
Red Flags for bufferpools

• In the next few slides, an analysis of a bufferpool under stress is shown
• First the raw SMF data for two weeks was processed thru the “old MP1B” MQCSMF and the MQ1150 format and print programs
• The Buffer Manager statistics were downloaded into a spreadsheet
• The spreadsheet was sorted to find:
  • Non-Zero Short on storage counts
  • Non-Zero DMC counts
  • Percent of free pages
• This showed the areas that needed to be looked at in greater detail, and it became apparent that there were some processing patterns that need evaluation
SMF 115 data – Hunting down the Culprit

- Red Flags for Bufferpools
  - SOS

<table>
<thead>
<tr>
<th>QMGR</th>
<th>BP</th>
<th>NumBuff</th>
<th>%now</th>
<th>%low</th>
<th>dwt</th>
<th>dmc</th>
<th>stl</th>
<th>stla</th>
<th>sos</th>
</tr>
</thead>
<tbody>
<tr>
<td>QML2</td>
<td>3</td>
<td>70000</td>
<td>18</td>
<td>0</td>
<td>109</td>
<td>198908</td>
<td>922354</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>QML2</td>
<td>3</td>
<td>70000</td>
<td>19</td>
<td>0</td>
<td>68</td>
<td>143872</td>
<td>367873</td>
<td>1</td>
<td>13</td>
</tr>
</tbody>
</table>

- Freepages at 5% or less

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>QMGR</th>
<th>BP</th>
<th>NumBuff</th>
<th>%now</th>
<th>%low</th>
<th>dwt</th>
<th>dmc</th>
<th>stl</th>
<th>stla</th>
<th>sos</th>
</tr>
</thead>
<tbody>
<tr>
<td>201133408:15:21</td>
<td>QML1</td>
<td>3</td>
<td>70000</td>
<td>98</td>
<td>5</td>
<td>9</td>
<td>27</td>
<td>32557</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>201133420:41:19</td>
<td>QML1</td>
<td>3</td>
<td>70000</td>
<td>95</td>
<td>5</td>
<td>2</td>
<td>384</td>
<td>61145</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
SMF 115 data

- **Red Flags for Bufferpools** - Continued
  - DMC – synchronous write process kicks off

<table>
<thead>
<tr>
<th>QMGR</th>
<th>BP</th>
<th>NumBuff</th>
<th>%now</th>
<th>%low</th>
<th>dwt</th>
<th>dmc</th>
<th>stl</th>
<th>stle</th>
<th>sos</th>
</tr>
</thead>
<tbody>
<tr>
<td>QML3</td>
<td>3</td>
<td>70000</td>
<td>16</td>
<td>0</td>
<td>58</td>
<td>210092</td>
<td>863991</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>QML3</td>
<td>3</td>
<td>70000</td>
<td>22</td>
<td>3</td>
<td>132</td>
<td>36526</td>
<td>1232774</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

- The DMC count should be used in conjunction with the IMW field from the SMF115 report to see how many synchronous writes were actually performed.
The NEW SMF print – BUFCSV file

- This spreadsheet image was created from WMQ V7.0.1 data thru the new MP16 print program
  - Note the data produced is different from the MQCSMF report from the old version. Important fields missing include the important SOS counts, deferred writes, and synchronous writes fields.

<table>
<thead>
<tr>
<th>MVS</th>
<th>QM</th>
<th>Date</th>
<th>Time</th>
<th>BP</th>
<th>size</th>
<th>lowest free</th>
<th># get new pg</th>
<th># get old pg</th>
<th># read I/Os</th>
<th># pg writes</th>
<th># write I/Os</th>
<th># sync</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPX1</td>
<td>QML1</td>
<td>2010/09/29</td>
<td>15:32:18</td>
<td>0</td>
<td>5000</td>
<td>4980</td>
<td>0</td>
<td>66219</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MPX1</td>
<td>QML1</td>
<td>2010/09/29</td>
<td>15:32:18</td>
<td>1</td>
<td>15000</td>
<td>7233</td>
<td>15302</td>
<td>31695</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MPX1</td>
<td>QML1</td>
<td>2010/09/29</td>
<td>15:32:18</td>
<td>2</td>
<td>40000</td>
<td>5980</td>
<td>39371</td>
<td>32569</td>
<td>116</td>
<td>17400</td>
<td>4350</td>
<td>0</td>
</tr>
<tr>
<td>MPX1</td>
<td>QML1</td>
<td>2010/09/29</td>
<td>15:32:18</td>
<td>3</td>
<td>20000</td>
<td>3281</td>
<td>18921</td>
<td>29093</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MPX1</td>
<td>QML1</td>
<td>2010/09/29</td>
<td>15:32:18</td>
<td>4</td>
<td>30000</td>
<td>29999</td>
<td>0</td>
<td>68</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MPX1</td>
<td>QML1</td>
<td>2010/09/29</td>
<td>15:32:18</td>
<td>5</td>
<td>30000</td>
<td>29999</td>
<td>0</td>
<td>134</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MPX1</td>
<td>QML1</td>
<td>2010/09/29</td>
<td>15:32:18</td>
<td>9</td>
<td>20000</td>
<td>2583</td>
<td>17521</td>
<td>22273</td>
<td>338</td>
<td>4976</td>
<td>1244</td>
<td>0</td>
</tr>
</tbody>
</table>
SMF115 – Bufferpool Trends and Analysis

Monday Bufferpool Use

Complete your session evaluations online at www.SHARE.org/AnaheimEval
In the chart shown two high volume days were compared to see if there was a pattern to the BP use.

BP 0, 1 and 2 showed almost no utilization.

BP 3 was in very heavy use, some of the time.

BP 3 is under some stress.

Having multiple days worth of data is vital, had there just been one heavy day it may have been an anomaly. Data from longer periods of time, when compared like this can be very useful in tracking usage, etc.

In this case there was a clear pattern of overuse of bufferpool 3, in further evaluation the SMF116 data showed that all the queues that were being used for this queue manager were defined on the same pageset/bufferpool. By moving some of the queues to another resource pool, the stress was reduced, work flowed faster and the CPU usage was reduced.

In attempting to replicate the issues, the information on the previous slides was used to create the charts, but also to show that charting the pattern might be helpful in the evaluation.
SMF 115 data

- Yellow Flags for Bufferpools
  - Consistently Approaching/Achieving 20 % Free pages

<table>
<thead>
<tr>
<th>QMGR</th>
<th>BP</th>
<th>NumBuff</th>
<th>%now</th>
<th>%low</th>
<th>Qwt</th>
<th>dmc</th>
<th>stl</th>
<th>stla</th>
<th>sos</th>
</tr>
</thead>
<tbody>
<tr>
<td>QML4</td>
<td>2</td>
<td>70000</td>
<td>53</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>46571</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QML4</td>
<td>3</td>
<td>70000</td>
<td>98</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>46028</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QML4</td>
<td>3</td>
<td>70000</td>
<td>75</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Bufferpool Use - Warnings

- In the data shown, two bufferpools were approaching the 20% freepage threshold.
- At 20% the async write task is initiated, which is not catastrophic, but if it can be avoided it should be.
- In this case, when several weeks worth of data were examined the 20% threshold was being broken on a regular basis. After evaluation of the SMF116 class-3 data it was found this was batch oriented workload, and messages were expected to queue up for long periods of time this was not a problem. It is something to watch.
SMF 115 data – Hunting down the Culprit

- Message Manager Information
  - Good indication of queue manager usage
    - This is only a count of API calls, not one of successful calls
    - Volume trends can be approximated from the MQPUT and MQPUT1 calls, as these are generally successful
    - MQGETs may or may not have data returned

<table>
<thead>
<tr>
<th>QMGR</th>
<th>Open</th>
<th>Close</th>
<th>Get</th>
<th>Put</th>
<th>Put1</th>
<th>Inq</th>
<th>Inql</th>
<th>Set</th>
<th>Total API calls</th>
<th>Total Puts</th>
</tr>
</thead>
<tbody>
<tr>
<td>QML1</td>
<td>160</td>
<td>151</td>
<td>2,925,084</td>
<td>3,417,313</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6,342,709</td>
<td>3,417,313</td>
</tr>
<tr>
<td>QML2</td>
<td>245</td>
<td>226</td>
<td>2,256,084</td>
<td>3,150,666</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>5,407,231</td>
<td>3,150,666</td>
</tr>
<tr>
<td>QML3</td>
<td>897</td>
<td>895</td>
<td>3,468,114</td>
<td>3,093,355</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>6,563,311</td>
<td>3,093,355</td>
</tr>
</tbody>
</table>
Message Manager Statistics

- This data was taken from the message manager output from the old MQCSMF format and print program.
- Two columns were added to calculate the Total API calls and Total Puts (sum of MQPUT and MQPUT1 calls)
- When charted over a few weeks an upward curve was noticed.
Message Manager - Trend Chart

MQPUT total for Monday

Feb 27 MQPUT Total
March 05 MQPUT Total

Complete your session evaluations online at www.SHARE.org>AnaheimEval
Message Manager Trend Chart

• Two days data is not really a trend analysis, but it’s a start
• If more Mondays are charted, a real trend may emerge and show that volume is increasing allowing a good admin to plan for additional workload.
• This is an overall count for the queue manager, individual queue activity can be evaluated from the SMF116 class 3 data.
New MP1B Print program – Message Manager

<table>
<thead>
<tr>
<th>MVS</th>
<th>QM</th>
<th>Date</th>
<th>Time</th>
<th>Puts</th>
<th>Put1s</th>
<th>Gets</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPX1</td>
<td>QML1</td>
<td>2010/09/29</td>
<td>15:32:18</td>
<td>36070</td>
<td>0</td>
<td>30659</td>
</tr>
<tr>
<td>MPX1</td>
<td>QML2</td>
<td>2010/09/29</td>
<td>15:32:19</td>
<td>21725</td>
<td>0</td>
<td>16433</td>
</tr>
<tr>
<td>MPX1</td>
<td>QML3</td>
<td>2010/09/29</td>
<td>15:32:38</td>
<td>20289</td>
<td>0</td>
<td>16237</td>
</tr>
</tbody>
</table>

Complete your session evaluations online at www.SHARE.org/AnaheimEval
SMF 115 data – Hunting down the Culprit

- Log Manager Information
  - Good indication of persistent messaging use
    - As has been mentioned before some of the counts are not complete, the checkpoints does not include those from queue manager switching

<table>
<thead>
<tr>
<th>QMGR</th>
<th>wr_wait</th>
<th>wr_nwait</th>
<th>Aug09 Log Buffer</th>
<th>Aug09 Num I/O</th>
<th>Aug09 Control Intervals</th>
<th>Aug09 Written</th>
<th>Aug09 Paging</th>
</tr>
</thead>
<tbody>
<tr>
<td>QML1</td>
<td>0</td>
<td>569925</td>
<td>339</td>
<td>1</td>
<td>0</td>
<td>22020</td>
<td>241748</td>
</tr>
<tr>
<td>QML1</td>
<td>0</td>
<td>621641</td>
<td>337</td>
<td>0</td>
<td>0</td>
<td>23758</td>
<td>230944</td>
</tr>
<tr>
<td>QML1</td>
<td>0</td>
<td>753611</td>
<td>363</td>
<td>1</td>
<td>0</td>
<td>27490</td>
<td>285402</td>
</tr>
</tbody>
</table>
Message Manager – New MP1B output

- This is from the Message Manager CSV file.
- Note that all it reports are the MQPUT, MQPUT1 and MQGET requests.
- If you are looking for any of the other requests, the TASK report (from the MP116 data) must be used.
SMF 115 data – Hunting down the Culprit

- Log manager – I/O rate
  - The I/O rate is calculated as
    - The number of CIs written * 4096 (CI size)
    - Divided by 1 M (1024*1024)
    - Divided by the number of seconds in the interval
  - The I/O rate is the throttle for many queue managers

<table>
<thead>
<tr>
<th>Aug08 Control Intervals Written</th>
<th>Aug08 Logging Rate (MB per second)</th>
<th>Aug09 Control Intervals Written</th>
<th>Aug09 Logging Rate (MB per second)</th>
<th>Sept30 Control Intervals Written</th>
<th>Sept30 Logging Rate (MB per second)</th>
<th>Nov05 Control Intervals Written</th>
<th>Nov05 Logging Rate (MB per second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20658</td>
<td>1.34</td>
<td>241748</td>
<td>15.74</td>
<td>58938</td>
<td>3.84</td>
<td>33492</td>
<td>2.18</td>
</tr>
<tr>
<td>22446</td>
<td>1.46</td>
<td>230944</td>
<td>15.04</td>
<td>70570</td>
<td>4.59</td>
<td>25822</td>
<td>1.68</td>
</tr>
<tr>
<td>22550</td>
<td>1.47</td>
<td>285402</td>
<td>18.58</td>
<td>46630</td>
<td>3.04</td>
<td>27688</td>
<td>1.80</td>
</tr>
<tr>
<td>20870</td>
<td>1.36</td>
<td>266212</td>
<td>17.33</td>
<td>79076</td>
<td>5.15</td>
<td>76658</td>
<td>4.99</td>
</tr>
<tr>
<td>23458</td>
<td>1.53</td>
<td>307780</td>
<td>20.04</td>
<td>53588</td>
<td>3.49</td>
<td>74088</td>
<td>4.82</td>
</tr>
</tbody>
</table>
Log Manager Statistics

- This data was taken from the log manager output from the old MQCSMF format and print program.
- The log buffer waits indicates the number of times during the interval there were not free log buffers. This is somewhat tunable, but most production environments have it set to the recommended 40,000. If this count goes very high and the maximum number of buffers are allocated, then the queue manager may be saturated.
- Another critical factor is the I/O rate that can be achieved...
Logging Rates - Charted

Logging Rates
MB per second

Aug08 Logging Rate (MB per second)
Aug09 Logging Rate (MB per second)
Sept30 Logging Rate (MB per second)
Nov05 Logging Rate (MB per second)

Complete your session evaluations online at www.SHARE.org/AnaheimEval
Logging Rates Charted - Notes

- These rates were charted from days when there were reported slowdowns.
- Notice the huge spike in the Aug 09 data, that is an anomaly caused by a change to the SMF interval in the middle of the day. It is left in to illustrate that spikes do happen and should be investigated. They may not indicate a sudden growth rate, but can indicate a problem with the data itself.
- In looking at the data, the logging rate is frequently at the 20/25 MB per second rate. For the environment, this was quite high. It was discussed with the capacity planning team. This is continually being monitored, there may be workload that has to shift to another queue manager in the near future.
Log Manager CSV file from new MP1B

<table>
<thead>
<tr>
<th>z/OS</th>
<th>QM</th>
<th>Date</th>
<th>Time</th>
<th>MB Written</th>
<th>MB/SEC</th>
<th>MB Used</th>
<th>Pages per I/O</th>
<th>Checkpoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPX1</td>
<td>QML1</td>
<td>2010/09/29</td>
<td>15:32:18</td>
<td>400</td>
<td>0</td>
<td>399</td>
<td>34</td>
<td>0</td>
</tr>
<tr>
<td>MPX2</td>
<td>QML2</td>
<td>2010/09/29</td>
<td>15:32:19</td>
<td>340</td>
<td>0</td>
<td>337</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>MPX1</td>
<td>QML3</td>
<td>2010/09/29</td>
<td>15:32:38</td>
<td>441</td>
<td>0</td>
<td>438</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>MPX2</td>
<td>QML4</td>
<td>2010/09/29</td>
<td>15:34:02</td>
<td>876</td>
<td>0</td>
<td>864</td>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>

• The new MQCSMF print program will calculate the MB/Second written
  • A caution, it uses the number of seconds per SMF interval defined for the run. If you allow this to default, your results are likely to be incorrect.
SMF 115 data – CF Statistics

- The CF data from MQ should be used in conjunction with the Coupling Facility Activity Reports (CFRM).

- The average elapsed time is reported in microseconds, and in this example is low. As it happens the CF in use is ‘local’ – in the same CEC, so they should be low.

- Recommendation is to chart the values over time, like the other statistics looking for anomalies and use patterns.
SMF 115 data – DB2 Statistics

- This data includes the average am maximum times spent on DB2 requests
  - Average elapsed time on the DB2 task
  - Average elapsed time on the DB2 server
  - Maximum elapsed time on the DB2 task
  - Maximum elapsed time on the DB2 server

- Recommendation is to chart the values over time, like the other statistics looking for anomalies and use patterns.
SMF 115 data – SDB2 Statistics

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>QMGR</th>
<th>Max_Depth</th>
<th>Num_deadlock</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013351</td>
<td>00:08:24</td>
<td>QML1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2013351</td>
<td>00:44:52</td>
<td>QML1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2013351</td>
<td>01:14:46</td>
<td>QML1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2013351</td>
<td>01:44:40</td>
<td>QML1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- The additional DB2 information shows the
  - Maximum depth of queues requests into DB2
  - Whether there were deadlocks
  - As with the other statistics, these should be charted to show usage patterns and detect anomalies
SMF116 Class 3 data

• Reviewing this copious data can feel like searching for the spell to turn lead into gold. It’s more like panning for gold.
• As a WMQ admin, you have more information at your fingertips about your environment than we at IBM reviewing this data will have. There are a number of things that we do to look for patterns or particular problems that are discussed.
SMF116 - Hunting the culprit

- The scenario is simple:
  - ‘We are missing our SLAs on some of our transactions’
  - The SMF 115 may or may not show bottlenecks
  - You have over 3M SMF116 class 3 records from one SMF interval to see if you can find the problem
  - And, of course, ‘MQ is the problem’
What queues are being used and how?

- SMF116 class 3 data shows the use of queues
- Helpful because even as a WMQ admin, it may be a challenge to find out where the queues are
- Some specific problems:
  - Non-indexed queues
  - High volume request/reply queues in same resource pool
  - Overuse of Temporary dynamic queues
What queues are being used and how?

- We have seen some specific problems/issues at a number of customers that were addressed with an evaluation of the SMF116 data.
- In this section we are going to show some of the more common ones, and how the SMF data lead to the improvement
What queues are being used and how?

- **Queue Indexing**
  - Messages that are retrieved using an index-able field benefit from being indexed even when the depth is not high.
    - Message ID
    - Correlation ID
    - Token
    - Group ID
  - The greater the depth of the queue the greater the benefit.
  - The SMF116 queue records show when messages are retrieved using a ‘known’ field
Queue Indexing - Notes

- Queue indexing is unique to WMQ on z/OS
- The use of a proper index can substantially improve performance and CPU consumption, as will be shown
- Anecdotally, we’ve heard of it making a difference when queue depths were as low as 5 on a busy system
- Often the first report of a problem is when there has been a slowdown elsewhere and queue depths have grown unexpectedly
Non-Indexed Queue retrieval

Open name TEAMXX.NON.INDEXED  Object type: Local Queue
Base name TEAMXX.NON.INDEXED  Base type : Queue
Queue indexed by NONE
First opened 12-03-2012 15:12:58.55
Last closed 12-03-2012 15:12:58.55
Page set ID 4, Buffer pool 3
Current opens 1, Total requests 61
Generated messages : 0
Persistent messages: GETs 0, PUTs 0, PUT1s 0
Put to waiting getter: PUT 0, PUT1 0
GETs: Valid 20, Max size 80, Min size 80, Total bytes 2240
GETs: Dest-S 20, Dest-G 0, Brow-S 0, Brow-G 0, Successful destructive 20
Time on queue: Max 4583,730054, Min 257.434901, Avg 3950.326341
-MQ call- N ET CT Susp LOGW PSET Epages skip expire
Get : 20 384 369 0 0 0 0 0 3505 0
Inquire: 20 22 21
Maximum depth encountered 258
Non-Indexed Queue retrieval - Notes

- In the SMF record, the fields of interest are:
  - The Queue Indexing
  - The Type of GET request being made, those with a ‘-S’ are for specific messages (Get by correlid, get by message id, etc.)
  - The total CPU expenditure for the successful gets – the ‘CT’ column highlighted
  - The number of pages skipped while finding matching messages
Indexed Queue Retrieval

Open name TEAMXX.INDEXED
Object type: Local Queue
Base name TEAMXX.INDEXED
Base type: Queue
Queue indexed by CORREL_ID
First opened 12-03-2012 15:16:01.44
Last closed 12-03-2012 15:16:30.35
Page set ID 4, Buffer pool 3
Current opens 0, Total requests 59
Generated messages:
0
Persistent messages: GETs 0, PUTs 0, PUT1s 0
Put to waiting getter: PUT 0, PUT1 0
GETs: Valid 27, Max size 80, Min size 80, Total bytes 2160
GETs: Dest-S 27, Dest-G 0, Brow-S 0, Brow-G 0, Successful destructive 27
Time on queue: Max 4790.946117, Min 422.046399, Avg 4288.437716
-MQ call- N ET CT Susp LOGW PSET Epages skip expire
Get 27 105 99 0 0 0 0 0 0
Inquire: 26 21 20
Maximum depth encountered 258

Complete your session evaluations online at www.SHARE.org/AnaheimEval
Indexed Queue retrieval - Notes

• In the SMF record, the fields of interest are:
  • The Queue Indexing
  • The Type of GET request being made, those with a ‘-S’ are for specific messages (Get by correlid, get by message id, etc.)
  • The total CPU expenditure for the successful gets – the ‘CT’ column highlighted
  • The number of pages skipped while finding matching messages
Indexed vs Non - comparison

• Comparing the CPU time, both queues with the same max message depth:
  • Indexed 27 messages at 99 CPU microseconds
    • 3.667 ms per message retrieved
  • Non-indexed 28 messages at 369 CPU microseconds
    • 13.18 ms per message

• Comparing the number of pages that had to be skipped
  • Indexed = 0
  • Non-indexed = 3585
What queues are being used and how?

- High volume request and reply queue in the same resource pool
  - This is a case of ‘define like’ run amok
  - The request queue and reply queue for a high volume application were defined in the same storage class (same bufferpool and pageset)
- By moving the reply queue to a different storage class, the resource usage was better distributed
High volume request and reply queue in the same resource pool

- Note this is often not seen until there is stress in one or more bufferpools due to volume.
What queues are being used and how?

- Overuse of Temporary dynamic queues
  - Often used for responses on both RYO and traditional monitoring tools
  - All queues created will be in the same resource pool
  - Quite expensive in CPU
- Temp dynamic queues are identifiable by their name
  - For example, for the MQExplorer uses temporary dynamic queues. The name looks like this
    AMQ.MQEXPLORER.1363497285
Temporary Dynamic Queues

Open name TEAMXX.MODEL
Base name AMQ.C9422A60F43B6075

Object type: Local Queue
Base type: Queue

Queue indexed by NONE
First opened 12-03-2012 21:24:16.34
Page set ID 0, Buffer pool 0
Current opens 0, Total requests 10
Generated messages: 0
Persistent messages: GETs 0, PUTs 0, PUT1s 0
Put to waiting getter: PUT 0, PUT1 0

PUTs: Valid 3, Max size 9, Min size 9, Total bytes 27
-MQ call-
Open : 1 050 125 727
Close : 1 113 111 0
Put : 3 106 104 0 0
Inquire: 5 17 17
Maximum depth encountered 3
Permanent Queues

== Task token : 12-03-2012 21:24:23.42, 55FE03F0, 55FD0000

Open name TEAMXX.NOT.TEMP
Base name TEAMXX.NOT.TEMP

Queue indexed by NONE
First opened 12-03-2012 21:25:09.23
Last closed 18-10-2019 00:31:45.22

Page set ID 0, Buffer pool 0
Current opens 0, Total requests 10
Generated messages : 0
Persistent messages: GETs 0, PUTs 0, PUT1s 0
Put to waiting getter: PUT 0, PUT1 0

PUTs: Valid 3, Max size 9, Min size 9, Total bytes 27

-MQ call- N ET CT Susp LOGW PSET Epages skip expire
Open : 1 39 38 0
Close : 1 26 26 0
Put : 3 115 113 0 0
Inquire: 5 18 18

Maximum depth encountered 3
Temp vs. Permanent

• The CPU cost comparison
  • Verb  TDQ  Permanent
  • Open  125  38
  • Close  111  26
  • Put    104  113
  • Inquire 17   18

• The Elapsed Time comparison
  • Verb  TDQ  Permanent
  • Open  850  39
  • Close  113  26
  • Put    106  115
  • Inquire 17   18
What queues are actually in use?

- The SMF116 class 3 data hold the information into the actual queue use
  - This information can be critical in tracking down a performance problem or for capacity planning
  - The MQCSMF GET and PUT files can be used to track the use
  - This file can contain millions of records
What is the volume on the different queues?

<table>
<thead>
<tr>
<th>Queue Name</th>
<th>Bufferpool</th>
<th>Pageset or CF Structure</th>
<th>Number of References</th>
<th>Total Number of GETs</th>
<th>Valid GETs</th>
<th>Bytes Retrieved</th>
<th>Average Message Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LYN.TEST1</td>
<td></td>
<td>APPLPRD01</td>
<td>1086</td>
<td>2172</td>
<td>1086</td>
<td>1650720</td>
<td>1520.00</td>
</tr>
<tr>
<td>LYN.TEST2</td>
<td></td>
<td>APPLPRD02</td>
<td>2843</td>
<td>3882</td>
<td>1069</td>
<td>1710400</td>
<td>1600.00</td>
</tr>
<tr>
<td>LYN.TEST3</td>
<td></td>
<td>APPLPRD01</td>
<td>3394</td>
<td>3394</td>
<td>947</td>
<td>1462168</td>
<td>1544.00</td>
</tr>
<tr>
<td>LYN.TEST4</td>
<td></td>
<td>APPLPRD01</td>
<td>946</td>
<td>1892</td>
<td>946</td>
<td>1437920</td>
<td>1520.00</td>
</tr>
<tr>
<td>LYN.TEST5</td>
<td></td>
<td>APPLPRD03</td>
<td>3191</td>
<td>3191</td>
<td>881</td>
<td>1360264</td>
<td>1544.00</td>
</tr>
<tr>
<td>LYN.TEST6</td>
<td>3 4</td>
<td></td>
<td>672</td>
<td>672</td>
<td>672</td>
<td>167448</td>
<td>249.18</td>
</tr>
</tbody>
</table>

- A spreadsheet can be used to consolidate the queues that are actually used
- In the example shown above the GET file was used to illustrate actual queue use.
  - Should be combined with the PUT information
- Note the size of the files make this better suited for a database
What queues are actually in use? Continued - Notes

• The spreadsheet used a number of formulas to get the totals, an example of the formula to calculate the number of GET references is:
  =COUNTIF(AllGets!D03:D22219;A3)

• The formula to calculate the number of Valid GETs is:
  =SUMIF(AllGets!$D03:$D22219;A3;AllGets!$E03:$E22219)

• The bufferpool, pageset, and CF structure information was manually drawn from the SMF 116 print program
Hunting down the culprit – finding a transaction in the SMF116

• Many times you want to look at the information from a CICS transaction or batch job
  • No way to turn SMF116 class 3 on for just one TX or job
  • Use SORT
    • Remember you will have 2 passes!
    • First pass to sort out the ‘short’ records that the SMFDUMP program applies
    • Second pass to pull out the records for the transaction/batch job you want
Finding a specific transaction or batch job

• In a group of millions of records, pulling the information for a specific transaction to ‘map’ its behavior can be critical in both problem resolution and performance issues.
• The SMFDUMP program has few options for getting subsets of the data.
• Using a simple sort is a quick solution to dividing up this massive volume into manageable groups.
Finding a transaction

/**
  /**  THIS GETS RID OF THE 'FIRST AND LAST' SMF RECORDS THAT CAUSE THE
  /**  SORT TO COUGH UP BLOOD
  /**
  /**SYSIN  DD *
  OMIT COND=(6,1,CH,LT,X'73')
  SORT FIELDS=(19,4,CH,A)
/**

/**SYSOUT  DD  SYSOUT=*
/**SYSUDUMP  DD  SYSOUT=*
/**
/***  SELECT SMF116 BY TRANSACTION
/***
/**SYSIN  DD *
SORT FIELDS=(109,4,BI,A)
INCLUDE COND=(109,4,CH,EQ,C'ABCD')
/**
Finding a Batch job

```/*
/* THIS GETS RID OF THE 'FIRST AND LAST' SMF RECORDS THAT CAUSE THE
/* SORT TO COUGH UP BLOOD
*/
/*
/*SYSIN DD *
  OMIT COND=(6,1,CH,LT,X'73')
  SORT FIELDS=(19,4,CH,A)
*/

/*SYSUDUMP DD SYSOUT=* */
/*
/* THIS PULLS THE SMF RECORD FOR A SPECIFIED BATCH JOB */
/*
/*SYSIN DD *
  INCLUDE COND=(73,8,CH,EQ,'C'ELKINSC2')
  SORT FIELDS=(19,4,CH,A)
*/
```
SMF116 and Long running tasks

- IF the long running task is started after the Class 3 trace
  - SMF 116 records will be cut at each SMF interval and at task end
- If the task is started before the trace is
  - No records are cut
  - APAR PM58798 has been taken on this
Summary

• The SMF data can be used in many ways to find patterns of use, problems with the queue managers, and programming problems.

• There are many other things within the data that are helpful, and more to come with the 7.1 interpretations and print programs.

• Thank you
<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>08:00</td>
<td></td>
<td></td>
<td>What’s Available in MQ and Broker for High Availability and Disaster Recovery?</td>
<td>Best Practices in Enhancing our Security with WebSphere MQ</td>
<td>MQ &amp; CICS Workload Balancing in a 'Plexed' World</td>
</tr>
<tr>
<td>09:30</td>
<td></td>
<td></td>
<td>What’s Wrong with MQ?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:00</td>
<td>The Dark Side of Monitoring MQ - SMF 115 and 116 Record Reading and Interpretation</td>
<td></td>
<td></td>
<td>IIIB - Internals of IBM Integration Bus</td>
<td></td>
</tr>
<tr>
<td>12:15</td>
<td></td>
<td></td>
<td></td>
<td>Hands-on Labs for MQ - Take Your Pick!</td>
<td></td>
</tr>
<tr>
<td>01:30</td>
<td></td>
<td>What’s New in the MQ Family</td>
<td>MQ on z/OS – Vvisection</td>
<td>MQ Clustering - The Basics, Advances and What’s New</td>
<td></td>
</tr>
<tr>
<td>03:00</td>
<td>Introduction to MQ</td>
<td></td>
<td>WebSphere MQ CHINIT Internals</td>
<td>Using IBM WebSphere Application Server and IBM WebSphere MQ Together</td>
<td></td>
</tr>
<tr>
<td>04:30</td>
<td>First Steps with IBM Integration Bus: Application Integration in the new world</td>
<td>What’s New in IBM Integration Bus &amp; WebSphere Message Broker</td>
<td>MQ &amp; DB2 – MQ Verbs in DB2 &amp; InfoSphere Data Replication (Q Replication) Performance</td>
<td>MQ Parallel Sysplex Exploitation, Getting the Best Availability From MQ on z/OS by Using Shared Queues</td>
<td></td>
</tr>
</tbody>
</table>

This was session 15024 - The rest of the week ……

Complete your session evaluations online at [www.SHARE.org/AnaheimEval](http://www.SHARE.org/AnaheimEval)
Shameless Promotion

IBM WebSphere MQ V7.1 and V7.5 Features and Enhancements

Maximize your investment in WebSphere MQ

Discover new features that bring value to your business

Learn from scenarios with sample configurations

Redbooks

ibm.com/redbooks

High Availability in WebSphere Messaging Solutions

Design WebSphere solutions for high availability

Use WebSphere features to increase availability

Use PowerHA clusters to increase availability

Redbooks

ibm.com/redbooks

Complete your session evaluations online at www.SHARE.org/AnaheimEval
Please fill in evaluations at share.org/SFEval  #15024