Paging Dr. MQ - Health Check Your Queue Managers to Ensure They Won't Be Calling in Sick! [z/OS] – Session 17046

Lyn Elkins
elkinsc@us.ibm.com
Code!

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Is MQ sick? What can we do about it!

"Some patients like the magic wand prop."

www.shutterstock.com - 100107758

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
What is a MQ for z/OS Health Check?

• A health check is a point in time evaluation of the queue manager and channel initiator.
  – It is to provide singular focus and evaluation

• It does not replace:
  – Real time monitoring
  – Regular reviews of statistics
  – Regular reviews of accounting data

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
In messaging trouble? CRASHED QUERS SLOW QUEUEING

Better call Lyn! SMF NOW

CPU

Health checks
Objectives

- This session is to delve a bit deeper into how the team formerly known as ATS (now something less easy to remember) perform a health check and how you can perform your own.
- It is almost exciting as watching moss grow.
What data needs to be collected?

• The data typically requested for an MQ for z/OS health check typically includes:
  – One week of MQ Statistics data – SMF 115
    • For customers on MQ V8 this should include classes 1 and 4
    • For customers prior to V8, if gathering data about channels using the MQCMD program (or similar) if using is also helpful
  – One high volume day’s JES logs for the queue manager and channel initiator
What data needs to be collected? More

– Class 3 accounting for one or more busy, but not peak SMF intervals
  • If there are multiple peaks during a day, this data should be gathered during either the typical ‘ramp up’ or ‘ramp down’
  • A good cross section of tasks using MQ resources is necessary to evaluate the application’s use of MQ
– If on V8, Class 4 accounting for one or more busy but not peak SMF intervals.
What is a ‘busy but not peak period’?

- A sample week of MQPUT and MQPUT1 may illustrate patterns of use
Another example of workload patterns

MQPUT Comparisons

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
How do I find a ‘busy but not peak period’?

• Keep in mind (aka do as I say not as I do)
  – There may be multiple well defined peaks during a day
    • Batch and online workloads are typical
    • Multiple samples may be required to demonstrate all use
  – Some queue managers may have less well defined peaks – the message rates are more constant then the illustrated queue managers
  – Others may have well defined and predictable peaks
Finding the Symptoms

To find out how healthy an individual queue manager is, there are a number of important steps:

- Review the JES logs
  - Look for specific messages that indicate issues, some examples:
    - CSQJ032E - APPROACHING END OF THE LOG RBA RANGE OF FFFFFFFF. CURRENT LOG RBA IS .
    - CSQJ111A - OUT OF SPACE IN ACTIVE LOG DATA SETS
    - CSQJ112E - INSUFFICIENT ACTIVE LOG DATA SETS DEFINED IN BSDS
  - Calculate the number of log switches during a peak interval using the CSQJ033I message
Finding the Symptoms - continued

• JES log evaluation continued:
  – Track the storage use by capturing the CSQY220I messages
    • Example:
      CSQY220I QML1 CSQSCTL Queue manager storage usage: 110 local storage: used 597MB, free 1141MB:
      above bar: used 1GB, free >10GB
  – Look for long running applications:
    • Long running UOWs reported via CSQJ160I
    • Log shunting reported via CSQR026I
  – Look for messages that you have never seen before!
    • Your real time alerting system may not be looking at all the important messages.

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Finding symptoms - 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
  - SupportPac MP1B, no matter which version, provides at least 2 views into the MQ statistics
    - Report form
    - Column or CSV form
      - This is used to import into spreadsheets for analysis
    - Both are necessary because the CSV form may be missing fields you need to review when looking for a problem.
Finding the symptoms - 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
    - Differences between new and old for a CICS transaction are documented in
      https://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD106135
  • Other files, much like the ‘old’ MQCSMF output
Determining the health

Are you now, or have you ever been an MQ administrator????

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
SMF 115 – using the data

- The CSV file or the column format can be downloaded from z/OS and then imported into a spreadsheet for evaluation.
- For example, the log data loaded into a spreadsheet looks as follows:

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>QMGR</th>
<th>wr_wait</th>
<th>wr_wait</th>
<th>Aug08 Force Writes</th>
<th>Aug08 Log Buffer Waits</th>
<th>read_buf</th>
<th>read_act</th>
<th>read_arc</th>
<th>r_delay</th>
<th>N_CheckP</th>
<th>Aug08 Num I/O</th>
<th>Aug08 Control Intervals Written</th>
<th>paging</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011220 00:02:01.06</td>
<td>QML1</td>
<td>0</td>
<td>69242</td>
<td>447</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20658</td>
</tr>
<tr>
<td>2011220 00:03:01.05</td>
<td>QML1</td>
<td>0</td>
<td>74069</td>
<td>403</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22446</td>
</tr>
<tr>
<td>2011220 00:04:01.04</td>
<td>QML1</td>
<td>0</td>
<td>72830</td>
<td>453</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22550</td>
</tr>
<tr>
<td>2011220 00:05:01.08</td>
<td>QML1</td>
<td>0</td>
<td>70513</td>
<td>478</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20870</td>
</tr>
<tr>
<td>2011220 00:06:01.07</td>
<td>QML1</td>
<td>0</td>
<td>77959</td>
<td>489</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>23458</td>
</tr>
<tr>
<td>2011220 00:07:01.06</td>
<td>QML1</td>
<td>0</td>
<td>60920</td>
<td>483</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17732</td>
</tr>
<tr>
<td>2011220 00:08:01.06</td>
<td>QML1</td>
<td>0</td>
<td>65818</td>
<td>444</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18960</td>
</tr>
<tr>
<td>2011220 00:09:01.03</td>
<td>QML1</td>
<td>0</td>
<td>68325</td>
<td>476</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>19982</td>
</tr>
<tr>
<td>2011220 00:10:01.08</td>
<td>QML1</td>
<td>0</td>
<td>50712</td>
<td>478</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12088</td>
</tr>
</tbody>
</table>
SMF115 – Using the data

• By itself the SMF 115 data for many of the internal component managers are useful, but some manipulation of the raw data can make it much more consumable and useful.

• Examples include:
  – The message manager summaries as shown earlier
  – Calculating the log data written per second
  – Charting, making it easier to spot patterns and trends

• Some of the calculated data is done for you in the ‘new’ MP1B, but beware – it uses hard coded values for intervals, etc. which can skew results.
SMF115 – Using the data – continued

• Example of using a spreadsheet to calculate the log writes:

<table>
<thead>
<tr>
<th>Num I/O</th>
<th>Num_CI_W</th>
<th>paging</th>
<th>Seconds</th>
<th>Interval Duration</th>
<th>Log writes in MB per second</th>
</tr>
</thead>
<tbody>
<tr>
<td>142446</td>
<td>142460</td>
<td>0</td>
<td>980</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13352</td>
<td>13352</td>
<td>0</td>
<td>1877</td>
<td>897</td>
<td>0.06</td>
</tr>
<tr>
<td>12420</td>
<td>12420</td>
<td>0</td>
<td>2775</td>
<td>898</td>
<td>0.05</td>
</tr>
<tr>
<td>10956</td>
<td>10956</td>
<td>0</td>
<td>3670</td>
<td>895</td>
<td>0.05</td>
</tr>
<tr>
<td>10484</td>
<td>10484</td>
<td>0</td>
<td>4567</td>
<td>897</td>
<td>0.05</td>
</tr>
<tr>
<td>9274</td>
<td>9274</td>
<td>0</td>
<td>5463</td>
<td>896</td>
<td>0.04</td>
</tr>
<tr>
<td>8946</td>
<td>8946</td>
<td>0</td>
<td>6359</td>
<td>896</td>
<td>0.04</td>
</tr>
<tr>
<td>8864</td>
<td>8864</td>
<td>0</td>
<td>7256</td>
<td>897</td>
<td>0.04</td>
</tr>
<tr>
<td>1478</td>
<td>1476</td>
<td>0</td>
<td>8154</td>
<td>898</td>
<td>0.01</td>
</tr>
</tbody>
</table>

• The formulas used to calculate seconds and duration:
  - Seconds = SUM(HOUR(B2)*3600,(MINUTE(B2)*60),SECOND(B2))
  - Interval duration = (Q3-Q2)
  - Interval crossing a date = SUM((86399-Q96),Q97)

• The formula for log writes in MB/Second
  - = ROUND(((N3*4096)/(1024*1024))/(R3),2)
  - Where N3 = the number of CI Writes column and R3 is the Interval duration
Logging Rates - Charted
Reviewing the SMF 115 data – Are there problems?

- While not always the culprit, tuning the buffer pools has been a never ending cycle of opportunities
  - Lazy queue definitions
    - I copy what works, might not be what is best for the queue manager
  - Workload pattern changes
    - What flows today, might become a log jam without any warning
      - Well except the business knew they were opening 2600 new branches on Tuesday
SMF 115 data – Urgent symptoms – Call 911

- Unless there has been an unexpected spike in volume, this should never been seen!
  - SOS
  - Freepages at 5% or less

<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>198906</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>

<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>364</td>
<td>61145</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
SMF 115 data – Urgent symptoms

- Red Flags for Bufferpools – Continued
  - DMC – synchronous write process is requested
    - Note that it did not run this often, but this is the number of times that conditions were right!

<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>spo</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>883991</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>
SMF 115 data – Clairvoyance – Ask Doctor

- May be approaching a problem
  - 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>dwt</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>

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
SMF 115 data – Reviewing the Message Manager information for gross use numbers

- **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>Queue Manager</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>248</td>
<td>228</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 - Trend Chart

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Looking for more symptoms? Try the SMF116 Class 3 data

- Reviewing this copious data can feel like searching for the spell to turn lead into gold. Better to think of it as panning for gold.
- As an MQ 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.

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
SMF116 – Looking for things that make us less healthy

- The SMF 116 class 3 data may provide information about the health of the queue manager and the applications using MQ
  - For example, this data can be used:
    - Identify heavily used queues that can benefit from splitting across resource pools
    - Identify shared queues that should be on the same structure
    - Identify queues that should be indexed
    - Identify overly long response times, typically from I/O
    - Identify applications that are consuming excessive CPU due to bad verbs

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
What queues are being used and how?

- SMF116 class 3 data shows the use of queues
- Helpful because even as an MQ admin, it may be a challenge to find out where the queues are.
  - For example, if an application opens an alias, that points to another…this data provides the base name of the object
- Some specific opportunities for prevention:
  - Non-indexed queues
  - High volume request/reply queues in same resource pool
  - Overuse of Temporary dynamic queues
Queue Indexing – an opportunity to reduce the CPU fever

- 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
Non-Indexed Queue retrieval

<table>
<thead>
<tr>
<th>Open name</th>
<th>TEAMXX.NON.INDEXED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base name</td>
<td>TEAMXX.NON.INDEXED</td>
</tr>
<tr>
<td>Queue indexed by</td>
<td>NONE</td>
</tr>
<tr>
<td>First opened</td>
<td>12-03-2012 15:12:58.55</td>
</tr>
<tr>
<td>Last closed</td>
<td><strong>:</strong> <strong>:</strong> <strong>:</strong> <strong>:</strong> <strong>:</strong> <strong>:</strong> <strong>:</strong></td>
</tr>
<tr>
<td>Page set ID</td>
<td>4, Buffer pool</td>
</tr>
<tr>
<td>Current opens</td>
<td>1, Total requests</td>
</tr>
<tr>
<td>Generated messages</td>
<td>0</td>
</tr>
<tr>
<td>Persistent messages: GETs</td>
<td>0, PUTs</td>
</tr>
<tr>
<td>Put to waiting getter: PUT</td>
<td>0, PUT1</td>
</tr>
<tr>
<td>GETs: Valid</td>
<td>28, Max size</td>
</tr>
<tr>
<td>GETs: Dest-S</td>
<td>28, Dest-G</td>
</tr>
<tr>
<td>Time on queue: Max</td>
<td>4583.730054, Min</td>
</tr>
<tr>
<td>-MQ call-</td>
<td>N</td>
</tr>
<tr>
<td>Get</td>
<td>28</td>
</tr>
<tr>
<td>Inquire:</td>
<td>28</td>
</tr>
<tr>
<td>Maximum depth encountered</td>
<td>258</td>
</tr>
</tbody>
</table>

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
**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 15:16:50.35

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 50, Min size 50, Total bytes 2150

GETs: Dest-S 27, Dest-G 0, Brow-S 0, Brow-G 0, Successful destructive 27

Time on queue: Max 4788.945117, Min 422.046309, Avg 4288.437716

-MQ call-  
Get: 27 105 99 0 0 0 0 0 0 0
Inquire: 26 21 20

Maximum depth encountered 258

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Indexed vs Non - comparison

• Comparing the CPU time, both queues with the same max message depth:
  – Indexed 27 messages at 99 CPU microseconds
  – Non-indexed 28 messages at 369 CPU microseconds
  – An average of 270 microseconds less expensive per get
  – What’s a microsecond or two between friends?
• Comparing the number of pages that had to be skipped
  – Indexed = 0
  – Non-indexed = 3585
Another infection – concentration of resources

- 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
## GET queue use from SMF 116 class 3 data

<table>
<thead>
<tr>
<th>Queue</th>
<th>Get</th>
<th>ValidGet</th>
<th>Bytes</th>
<th>MaxGet</th>
<th>MinGet</th>
<th>MaxTOQ</th>
<th>Total Gets</th>
<th>Total Valid Gets</th>
<th>Total Bytes</th>
<th>Average Message Size</th>
<th>Buffer pool/Structure</th>
<th>Page Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>TESTQ1.QL</td>
<td>1</td>
<td>1</td>
<td>15000</td>
<td>15000</td>
<td>15000</td>
<td>1.1E+04</td>
<td>5</td>
<td>5</td>
<td>75000</td>
<td>15000</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>TESTQ2.QL</td>
<td>2</td>
<td>1</td>
<td>161</td>
<td>161</td>
<td>161</td>
<td>2.1E+03</td>
<td>330</td>
<td>165</td>
<td>120777</td>
<td>731.98</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>TESTQ3.QL</td>
<td>55</td>
<td>54</td>
<td>5886</td>
<td>109</td>
<td>109</td>
<td>3.2E+04</td>
<td>55</td>
<td>54</td>
<td>5886</td>
<td>109</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>TESTQ4.QL</td>
<td>1</td>
<td>1</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>3.5E+05</td>
<td>50</td>
<td>50</td>
<td>6400</td>
<td>128</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>TESTQ5.QL</td>
<td>1</td>
<td>1</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>3.2E+05</td>
<td>50</td>
<td>50</td>
<td>6400</td>
<td>128</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>TESTQ6.QL</td>
<td>1</td>
<td>1</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>2.4E+05</td>
<td>50</td>
<td>50</td>
<td>6400</td>
<td>128</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>TEST.INITQ1</td>
<td>48</td>
<td>24</td>
<td>16416</td>
<td>684</td>
<td>684</td>
<td>4.4E+02</td>
<td>352</td>
<td>176</td>
<td>120384</td>
<td>684</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TEST.INITQ2</td>
<td>64</td>
<td>32</td>
<td>21888</td>
<td>684</td>
<td>684</td>
<td>2.5E+02</td>
<td>3826</td>
<td>1913</td>
<td>1308492</td>
<td>684</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TEST.INITQ3</td>
<td>170</td>
<td>85</td>
<td>58140</td>
<td>684</td>
<td>684</td>
<td>3.0E+02</td>
<td>1650</td>
<td>825</td>
<td>564300</td>
<td>684</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TEST.INITQ3</td>
<td>20484</td>
<td>10252</td>
<td>7012368</td>
<td>684</td>
<td>684</td>
<td>6.3E+04</td>
<td>1061308</td>
<td>53142</td>
<td>36349128</td>
<td>684</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TEST.INITQ4</td>
<td>172</td>
<td>80</td>
<td>58824</td>
<td>684</td>
<td>684</td>
<td>1.4E+04</td>
<td>1466</td>
<td>733</td>
<td>501372</td>
<td>684</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SYSTEM.CHANNEL.INITQ</td>
<td>77</td>
<td>34</td>
<td>26656</td>
<td>784</td>
<td>784</td>
<td>4.7E+02</td>
<td>2216</td>
<td>989</td>
<td>775596</td>
<td>784.22</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SYSTEM.CHANNEL.SYNCO</td>
<td>2</td>
<td>2</td>
<td>856</td>
<td>428</td>
<td>428</td>
<td>2.1E+08</td>
<td>189</td>
<td>176</td>
<td>75328</td>
<td>428</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

---

Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)
## PUT queue use from SMF 116 class 3 data

<table>
<thead>
<tr>
<th>Queue</th>
<th>Number valid PUTs</th>
<th>Number of Bytes Put</th>
<th>Average Message Size</th>
<th>Buffer pool/Structure</th>
<th>Page Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>TESTQ3.QL</td>
<td>7</td>
<td>1282</td>
<td>183.14</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>TESTQ4.QL</td>
<td>54</td>
<td>5886</td>
<td>109</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>ECHO.QL</td>
<td>48</td>
<td>6144</td>
<td>128</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>SYSTEM.CHANNEL.INITQ</td>
<td>4</td>
<td>3456</td>
<td>864</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SYSTEM.CHANNEL.SYNCQ</td>
<td>174</td>
<td>74472</td>
<td>428</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SYSTEM.CLUSTER.COMMAND.QUEUE</td>
<td>2</td>
<td>1000</td>
<td>500</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SYSTEM.COMMAND.INPUT</td>
<td>919</td>
<td>1257221</td>
<td>1368.03</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Possibly mild, possibly virulent symptom

- Use (or 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.C1363497285
```
Temporary Dynamic Queues

Open name TEAMXX.MODEL
Base name AMQ.C9422A60F43865075

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 : 1 850 125 727
Close : 1 113 111 0
Put : 3 105 104 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
Object type: Local Queue
Base type: Queue
Queue indexed by NUNE
First opened 12-03-2012 21:25:09.23
Last closed 10-10-2019 00:31:46.22
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 10 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
Looking for a specific microbe – or one CICS transaction in the SMF116 data

- 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 queue, transaction, or job
- Use SORT

- Remember you will have to omit the short records that the SMF dump program always includes – use the VLSHRT

```bash
//SYSOUT DD SYSOUT=*  
//SORTIN DD DISP=SHR, DSN=*, SEL.OUTDD1  
//SORTOUT DD DISP=(,CATLG), DCB=(RECFM=VBS, BLKSIZE=4096, LRECL=32760),   
// DSN=ELKINSC.SMFC05.AUG01A.OUT,  
// SPACE=(CYL, (10,10))  
//SYSIN DD *  
   OPTION VLSHRT  
   INCLUDE COND=(19,4,CH,EQ,C'CS05')  
   SORT FIELDS=COPY  
/*
```
Finding a transaction

//S1 EXEC PGM=SORT
//SYSOUT DD SYSOUT=* 
//SORTIN DD DISP=SHR, DSN=*.SEL.OUTDD1
//SORTOUT DD DISP=(, CATLG), DCB=(RECFM=VBS, BLKSIZE=4096, LRECL=32760),
// DSN=ELKINSC.SMFCSQ5.AUG01A.OUT,
// SPACE=(CYL, (10,10))
//SYSIN DD *

OPTION VLSHRT
INCLUDE COND=(109,4,CH,EQ,C’ABCD’)
SORT FIELDS=(109,4,BI,A)

/*
Finding a Batch job

```
//SYSOUT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//SYSIN DD *  
  OPTION VLSHRT  
  INCLUDE COND=(73,8,CH,EQ,C'MOVER  
  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

• Performing your own periodic MQ health checks can help you focus on existing issues, or sharpen your foretelling.

• 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>08:30</td>
<td></td>
<td></td>
<td>17060: Understanding MQ Deployment Choices and Use Cases</td>
<td>17051: Application Programming with MQ Verbs [z/OS &amp; Distributed]</td>
<td>16544: Why Shouldn't I Be Able To Open This Queue? MQ and CICS Security Topics</td>
</tr>
<tr>
<td>11:15</td>
<td>17041: First Steps with IBM Integration Bus: Application Integration in the New World [z/OS &amp; Distributed]</td>
<td>16732: MQ V8 Hands- on Labs! MQ V8 with CICS and COBOL</td>
<td>17046: Paging Dr. MQ - Health Check Your Queue Managers to Ensure They Won't Be Calling in Sick! [z/OS]</td>
<td>17053: MQ &amp; DB2 – MQ Verbs in DB2 &amp; InfoSphere Data Replication (Q Replication) Performance [z/OS]</td>
<td></td>
</tr>
</tbody>
</table>

All sessions in Seneca unless otherwise noted.
Further information in real books
And ... already available (draft)
Because we just have not seen this enough!!! Session # 17046

Complete your session evaluations online at www.SHARE.org/Seattle-Eval