Understanding the MQ distributed log statistics

ColinPaice
Published on 12/04/2018 / *Updated on 30/07/2018*

MQ distributed can publish a variety of statistics every 10 seconds. These include

1. API requests
2. Logging information
3. Usage of the box

This blog covers what is provided from the logging data.
The data is published to a topic. You can create your own subscription, have the data sent to a queue and write your own program to process the data, or use the sample amqsrua to subscribe and display the data

Create your own topic

You need a subscription to the topic in order to get the data for example

```
DEF SUB(MONLOG)

 TOPICSTR('$SYS/MQ/INFO/QMGR/colin/Monitor/DISK/Log')

 TOPICOBJ( )

 DEST(MONLOG)
```

Where my queue manager is called colin, and this is in the topic string.
In this example the data gets published to the queue called MONLOG. You can write your own programs to process the data.

AMQSRUA sample.

You can use the sample amqsrua to print out the data. This subscribes to the topic for you.

There are a couple of blog articles on amqsrua

- Statistics – publish and be dammed
- Statistics published to the system topic in MQ v9

On my linux system

```
/opt/mqm/samp/bin/amqsrua -m colin
```

When it prompts for data types, reply

```
DISK

Log
```

This prints out information like

```
Publication received PutDate:20180331 PutTime:13044235
Interval:10.000 seconds

Log - bytes in use 83886080

Log - bytes max 83886080

Log file system - bytes in use 1014132736

Log file system - bytes max 6207111168

Log - physical bytes written 598876160 59871780/sec

Log - logical bytes written 532921878 53278096/sec

Log - write latency 280 uSec

Log - write size 36809

Log - current primary space in use 150.82%

Log - workload primary space utilization 146.05%
```

The information is described below

Log – bytes in use 83886080
Log – bytes max 83886080
In my queue manager defaults file (qm.ini) I have
LogPrimaryFiles=3
LogSecondaryFiles=2
4096 LogFilePages = 16MB
(3+2 =) 5 logs at 16 MB is 80MB = 83886080 bytes. This matches the Log bytes in use.

Log file system – bytes in use 1014132736
Log file system – bytes max 6207111168
These values are similar to the Linux command
6207111168 is 5.7808 GB, 1014132736 bytes is 0.944 GB

```
df -H /var/mqm/

Filesystem                Size Used Avail Use% Mounted on

/dev/mapper/uservg-varmqm 6.3G 1.1G  4.9G  18% /var/mqm
```

6207111168 bytes is 5.7808 GB, 1014132736 bytes is 0.944 GB

Log – physical bytes written 598876160 59871780/sec
Log – logical bytes written 532921878 53278096/sec
Data is written to the log in units of 4KB pages. This is the physical bytes written.
Consider an MQPUT followed by a commit. 1000 bytes are written to the log. The 1000 bytes are the logical bytes written.
One page is written so there 1 * 4KB physical bytes written.
If there is a second MQPUT followed by a commit, if there is space in the page, the same 4KB page is used, so the page now has 1000+ 1000 bytes in the page.
The amount of logical data written for the second put + commit is 1000 bytes from the first request + 1000 from the second request. One page is written, so this is 1 * 4KB physical bytes + 2000 logical bytes
If a big message is put to a queue, so 4 full pages and the 5th page has 500 bytes, 5*4KB physical bytes are written. and 4*4KB+ 500 logical bytes are written. If there is now a put of a small message 2000 bytes are written in the 5th page and this page is rewritten.
First request had 5 * 4KB physical bytes and 4*4KB+500 logical bytes.*
Second request write 1 4KB page physical and 500 + 2000 logical bytes.

A big difference between logical and physical bytes written, indicates lots of partial paging (e.g. small non-concurrent load, or PUTs out of syncpoint). See section 2.3 of here.

Log – write latency 280 uSec. This is an average of the time to write the data.
Log – write size 36809, This is an average of the size of the data written in each IO request
The write time and the amount of data written are like many MQ fields, a “geometric average”.
For example if the current value of the write time is 200 microseconds, and the some data is written to the log, taking 250 microseconds.
The new write time is ((1023* 200) + 250)/1024 = 200.05
Similarly with the size of data written
If the put+ commit of a message wrote 60KB(61440) bytes and the put of a second message + commit wrote 4KB (4096)
If the previous value of size of data written was 30000,
the new value after the first put+commit is

NV1 = ((102*30000) + 61440)/1024 = 30031*
the new value after the second put + commit is
*NV2 = ((102*30031) + 4096)/1024 = 30006
Note this different to (total data written/number of requests).
This geometric mean value is recalculated whenever and IO is done. So if the system is doing no work, the value will not change.

Log – current primary space in use… see LOGINUSE
Log – workload primary space utilization… see LOGUTIL