Infrastructure Series

TechDoc
WebSphere MQ

Performance – Queue Buffer Tuning (Distributed)

January 2014 - WebSphere MQ v7.5
Author: Glen Brumbaugh

Introduction

Document Version

This document describes how to add additional buffering to WebSphere MQ (WMQ) queues. This document reflects the capabilities and procedures of the following Products and Versions:

- WebSphere MQ v7.5

This documentation has been created and maintained by:

- Glen Brumbaugh

This documentation was last updated:

- January 2014

WebSphere MQ Message Processing Overview

WebSphere MQ (WMQ) processes messages. These messages are comprised of a "body" and one or more "headers". The "body" is created by the Application using WMQ and the "headers" are created by WMQ. The total message can be up to 100 MB in length. This length can restricted for all queues by reducing the "MaxMsgL" parameter at the Queue Manager, by reducing the length at the Queue level, or both.

Messages can be either "persistent" or "non-persistent". All "persistent" messages must be written to the WMQ log file before they can be processed. Non-persistent messages are not written to the log file. Any message however, regardless of persistence, may need to be written to the queue file (on disk) if there is insufficient memory to store the message in memory until it is processed.

WMQ does not have a "Global" memory pool in which to store messages, but instead has individual buffers, one for each queue. These buffers are not specified through MQSC when the queue is defined, but instead are inherited from a global Queue Manager setting (usually the default value) of a parameter contained in the mqs.ini file (UNIX) or in the Operating System registry (Windows). The values themselves, once established, are stored in the file associated with the queue in question. They persist until the queue is deleted.

The default buffer sizes for queues (with Default Message Type is Non-Persistent) are as follows:

- 64 KB (For 32 bit Queue Managers; normally Windows)
- 128 KB (For 64 bit Queue Managers; normally UNIX)

The default buffer sizes for queues (with Default Message Type is Persistent) are as follows:

- 128 KB (For 32 bit Queue Managers; normally Windows)
- 256 KB (For 64 bit Queue Managers; normally UNIX)

All of these buffers may be extended up to a maximum size of 100MB.

WebSphere MQ Performance Tuning Overview

WMQ performance tuning for high performance systems involves identifying the bottlenecks in a WMQ flow and removing or minimizing the bottlenecks. Physical I/O (e.g. writing to disk) is one of the slowest operations in WMQ and is frequently a bottleneck in performance. While logging for persistent messages cannot be avoided, writing those messages to disk again may be avoided if the messages can be processed while they are still in memory. Since messages are processed in FIFO (First-In First-Out) order, once messages begin being written to disk, all future messages will be processed from disk if the arrival and consumption rates remain unchanged. This happens because the new messages are written to memory while the oldest messages are being read from disk.

Tuning messaging systems often revolves around ensuring that messages can be consumed faster than they are being generated. If messages arrive faster than they are consumed, queuing is inevitable and processing will necessary be slowed down to disk speeds. This can result in orders of magnitude (multiples of ten) differences in performance. For high volume systems that have been tuned, the average consumption rates will be greater than or equal to the average arrival rates of messages for any given queue. Increasing the Queue Buffer sizes for specific selected queues to eliminate arrival spikes and allow messages to continue being processed from memory may be beneficial. Testing is required to determine both the benefit of additional memory and to determine the memory to be allocated.

Setting WMQ Queue Buffer Sizes

Overview

As mentioned previously, the size of the memory buffer associated with a queue is determined when a Queue is created and is stored in the file associated with the queue. The size of these buffers is determined by the following Queue Manager attributes:

- DefaultQBufferSize (Queues with a Default Message Type of Non-Persistent)
- DefaultPQBufferSize (Queues with a Default Message Type of Persistent)

As was previously indicated, the default sizes for these two values depends upon the Queue Manager version (32 bit versus 64 bit). These values are not dynamic (e.g. used when a Queue is opened) but rather are static and fixed when a Queue is defined. These attributes cannot be either viewed or set through MQSC commands.

These values can be modified and will take effect on all queues defined after both the modification of the values and a Queue Manager restart. The mechanism of this modification is different between the Windows and UNIX platforms because these values are stored in the file system on UNIX and in the Registry on Windows.

Note: An IBM SupportPac (MS0P) contains a program called “QTune”. The QTune program can be used to both see and modify these Queue Buffer values. This program is described in its own (following) Section. There is also a script than can report on the buffer settings for selected queues (any selection pattern that could be used in an MQSC “DISPLAY QLOCAL” command). This script is listed in an Appendix.

UNIX (mqs.ini file)

Queue Managers on UNIX systems describe these two Buffer sizes in the following file:

- /var/mqm/mqs.ini

These settings are described in the following stanza:

- TuningParameters:

The parameter names describe the number of bytes to be allocated to the buffer:

- DefaultQBufferSize=
- DefaultPQBufferSize=

To create or alter Queues to use a buffer size other than the default size, follow these steps:

1. Alter the mqs.ini file to specify the desired size(s) in the *TuningParameters* stanza.
2. Stop and restart the Queue Manager.
3. Ensure that any existing queues to be modified have no messages:
    a. Copy any existing messages to a file while preserving Message Header data.
4. Define a new queue for each queue to be deleted
    a. Define QLocal (queue.to.be.deleted.copy) LIKE (queue.to.be.deleted)
    b. This saves all of the queue's attributes
5. Delete any existing queues that need to be redefined.
6. Define the new queues:
    a. Brand new queues using standard queue definition procedures.
    b. Queues that were deleted are defined with (using DEFINE QLocal … LIKE …).
7. Delete all of the "copy" queues created in step 4.a.
8. Load saved messages back into the updated queues.
9. Alter the mqs.ini file:
    a. Comment out the TuningParameters stanza.
    b. Document i(n the stanza) which queues use the commented buffers.
10. Stop and restart the Queue Manager.

Windows (amqmdain command)

Queue Managers on Windows systems describe these two Buffer sizes in the Windows Registry. The Registry settings are:

- My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\MQSeries\CurrentVersion\Configuration\QueueManager\qmgrName\TuningParameters\
    - ✓ DefaultQBufferSize
    - ✓ DefaultPQBufferSize

To create or alter Queues to use a buffer size other than the default size, the *amqmdain* program must be executed instead of modifying the *mqs.ini* file. This command can be executed as follows:

1. Execute the following commands to <u>display</u> the desired Buffer sizes:

```
a.  amqmdain    -reg  qmgrName                (Queue Manager Name)
                -c  display                   (Display Registry parameters)
                –s  TuningParameters          (Name of Registry Stanza)
                –v  DefaultQBufferSize        (Non-Persistent Buffer size)
                –v  DefaultPQBufferSize       (Persistent Buffer size)
```

2. Execute the following commands to <u>set</u> the desired Buffer sizes:

```
a.  amqmdain    -reg  qmgrName                (Queue Manager Name)
                -c  add                       (Add a Registry parameter)
                –s  TuningParameters          (Name of Registry Stanza)
                –v  DefaultQBufferSize=512000    (Non-Persistent Buffer size)
                –v  DefaultPQBufferSize=1024000  (Persistent Buffer size)
```

3. Execute the following commands to <u>delete</u> the desired Buffer sizes:

```
a.  amqmdain    -reg  qmgrName                (Queue Manager Name)
                -c  remove                    (Delete Registry parameters)
                –s  TuningParameters          (Name of Registry Stanza)
                –v  DefaultQBufferSize        (Non-Persistent Buffer size)
```

```
–v DefaultPQBufferSize          (Persistent Buffer size)
```

To create Queues that use a buffer size other than the default size, follow the same basic steps described above (for UNIX) but with the following changes:

- Prior to step 1, run the amqmdain command previously described to **display** the current values of these parameters (they should be the default values).
- Instead of modifying the mqs.ini file in step 1, run the *amqmdain* command previously described to **add** the required values.
- Instead of modifying the mqs.ini file in step 9, run the *amqmdain* command previously described to **remove** the required values.

Note: The Queue manager should be started using the "*strmqm*" command rather than through WMQ Explorer.

WMQ SupportPac MS0P (QTune)

The MS0P WMQ SupportPac contains a program called "QTune". This program reads the WMQ files associated with queues and displays the buffering information stored within those files. This java program comes with both UNIX (.ksh) and Windows (.bat) scripts to launch the program. To use the program, you need to know what file the information for a particular queue is stored in. To determine this you can execute the following command:

- **dspmqfls** -m *qmgrName* -t queue *queueName*

Once you know the filename for the queue, the QTune program can be invoked by using the following command:

```
java -jar qtune.jar   -d /var/mqm/qmgrs/qmgrName/queues/queueFileName
                      -a npBuff=512          (32 – 102400; specified in KB)
                      -a pBuff=1024          (0-102400; specified in KB)
                      -a maxQSize=100        (20-2097151; specified in MB)
```

The QTune program can also be executed from a Korn shell script (UNIX) or a ".bat" script (Windows). Both of these scripts simply start the java program and pass the script parameters to the java program. These scripts therefore use the same parameters as the java program. The scripts are thus invoked as follows:

```
qtune.ksh             -d /var/mqm/qmgrs/qmgrName/queues/queueFileName
    or                -a npBuff=512          (32 – 102400; specified in KB)
qtune.bat             -a pBuff=1024          (0-102400; specified in KB)
                      -a maxQSize=100        (20-2097151; specified in MB)
```

Since a separate command must be executed for each queue, it can be quite cumbersome to display information about a large number of queues. For this reason a script has been created (qTuneDisplay.ksh) for use on UNIX to display information about one or more queues. This script is described in an Appendix to this document.

The output of the QTune program looks like this:

```
File /var/mqm/qmgrs/qmgrName/queues/SYSTEM!CLUSTER!TRANSMIT!QUEUE
Stored npBuff         = 64 kB
```

Stored pBuff	= QMgr default
Stored maxQSize	= 2,097,151 MB

Note: The QTune program documentation describes the usage of the "–m" (Queue Manager) and "–q" (Queue) parameters to request the display (not update) of Queue information. These parameters do not appear to be working on any of the tested platforms!

Warning: Use of this program is not supported by IBM! If you use it, make sure that your version of the program is current and that the SupportPac supports your version of WMQ.

Warning: If you use QTune to change the attributes of a queue, you should stop the Queue Manager first to ensure that the queue cannot be used while being updated by the QTune program!

WMQ Program: amqldmpa

There is an IBM program supplied with WMQ called amqldmpa. This program is used by IBM Support to dump WMQ configuration information and its usage is not publically documented. This program can, however, also be used to display WMQ Queue information; including Buffer sizes. The use of this program is described in a separate TechDoc.

Best Practices

- Test carefully and validate the performance impacts of increasing each queue buffer prior to migrating the setting to an important environment. A quote from the IBM Performance report:

  > *"It is easy to degrade the system performance by inappropriate use of these tuning parameters."*

- Change Queue Buffer sizes using the *mqs.ini* file parameters (UNIX) or the *amqmdain* command (Windows) rather than using the QTune program whenever possible for Production type systems. These methods are officially supported by IBM. The QTune program, despite being an official IBM SupportPac written by a member of the Hursley laboratory, is not officially supported.

References

- IBM TechNote: Queue Buffer Size Tuning Parameters
  http://www-01.ibm.com/support/docview.wss?uid=swg21246541
- IBM TechNote: Changing Default Queue Buffer Sizes
  http://www-01.ibm.com/support/docview.wss?uid=swg21162114
- IBM developerWorks: Configuring and Tuning WMQ for Performance
  http://www.ibm.com/developerworks/websphere/library/techarticles/0712_dunn/0712_dunn.html
- WebSphere MQ SupportPac: MP01 (Performance - Tuning Queue Limits)
  http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24006866&loc=en_US&cs=utf-8&lang=en

- WebSphere MQ SupportPac: MS0P (Extended Management Plug-ins)
  http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24011617&loc=en_US&cs=utf-8&lang=en

Appendix A: Sample mqs.ini File

Overview

The following sample *mqs.ini* file has the performance tuning values specified described. These values are highlighted in red.

Note: All new queues defined after these settings have been changed and the Queue Manager restarted will inherit these values!

mqs.ini File

```
#*******************************************************************#
#* Module Name: qm.ini                                               *#
#* Type       : WebSphere MQ queue manager configuration file        *#
#  Function   : Define the configuration of a single queue manager   *#
#*                                                                   *#
#*******************************************************************#
#* Notes      :                                                      *#
#* 1) This file defines the configuration of the queue manager       *#
#*                                                                   *#
#*******************************************************************#

ExitPath:
  ExitsDefaultPath=/var/mqm/exits/

Log:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=1024
  LogType=CIRCULAR
  LogBufferPages=0
  LogPath=/var/mqm/log/
  LogWriteIntegrity=TripleWrite

Service:
  Name=AuthorizationService
  EntryPoints=10

ServiceComponent:
  Service=AuthorizationService
  Name=MQSeries.UNIX.auth.service
  Module=/opt/mqm/lib/amqzfu
  ComponentDataSize=0

TuningParameters:
  DefaultQBufferSize=1024000
  DefaultPQBufferSize=512000
```

Appendix B: Buffer Size Reporting Script

Overview

The following script requires the QTune program and scripts to be installed in the same directory as the script. This script displays the Buffer settings for all selected queues. The queues to be displayed are specified in the same way they would be in an MQSC "DISPLAY QLocal (…)" command.

The script has two positional parameters. The first parameter is required and is the name of the Queue Manger. The second parameter is optional and provides all or part of a Queue name. This second parameter is formatted in the same way that a Queue name is used in the MQSC "Display QLocal (…)" command. An example of the command invocation is as follows:

- qTuneDisplay.ksh *QmgrName* System.*   (Display buffers for all SYSTEM queues)

If no parameters are entered, help information is displayed.

Script

```
#!/bin/ksh

#################################################################
### Script to display the WebSphere MQ Queue Buffer Settings. ###
### ------------------------------------------------------- ###
### This script must be installed in the same directory that  ###
### the qtune and its script (qtune.sh) are installed in.     ###
### ------------------------------------------------------- ###
### Parameters: (1)  Queue Manager Name (Required).           ###
###             (2)  Queue Name (Optional).                   ###
#################################################################

##################
### Functions. ###
##################

printOut ()
{
    echo "SCRIPT ==> $1" | tee -a  $OutputName
}


showHelp ()
{
    printOut "Script:  $ScriptName.                                  "
    printOut "         (Display WebSphere MQ Queue buffer settings)"
    printOut "                                                      "
    printOut "Required Parameters:                                  "
    printOut "     (1) Queue Manager Name                           "
    printOut "                                                      "
    printOut "Optional Parameters:                                  "
    printOut "     (2)  Queue Name                                  "
    printOut "                                                      "
}


showInvocation ()
{
    date  |  tee $OutputName
    printOut "                                             "
    printOut "$ScriptName (Utility for qtune program).    "
    printOut "                                             "
    printOut "    Output:                 Screen           "
    printOut "                            File ($TempName)"
    printOut "                                             "
    printOut "    Script is:              $scriptInvoked. "
    printOut "                                             "
    printOut "    Number of parameters:   $parmNumber.    "
    printOut "    Parameters are:        $parmList.      "
    printOut "                                           "
}



########################
### Script Constants. ###
########################

ScriptName="qTuneDisplay.ksh"
OutputName=$ScriptName.out
TempName=$ScriptName.temp

RequiredParameterNumber=1
MaximumParameterNumber=2



########################
### Script Variables. ###
########################

scriptInvoked="Undefined"
parmNumber="Undefined"
parmList="Undefined"

qmgrs="Undefined"
queue="Undefined"
dataPath="Undefined"



#####################
### Begin script. ###
#####################

touch $TempName
rm    $TempName

touch  $OutputName
rm     $OutputName
touch  $OutputName



##################################################
### Validate parameters:  No parameters.       ###
###                       Too many parameters. ###
##################################################

if [ $# -eq 0 ]
then
    clear
    printOut "No parameters specified!"
    printOut "                        "

    showHelp
    exit 1

elif [ $# -gt $MaximumParameterNumber ]
then
    clear
    printOut "Too many parameters specified!"
    printOut "                              "

    showHelp
    exit 2

else
    scriptInvoked=$0
    parmNumber=$#
    parmList=$@

    clear
    showInvocation
fi



#############################################
### Validate parameter 1:  Queue Manager. ###
#############################################

if [ $# -ge 1 ]
then
    qmgr=$1

    printOut "    Queue Manager:         $qmgr."
fi



#####################################
### Validate parameter 2:  Queue. ###
#####################################

if [ $# -ge 2 ]
then
    queue=$2
else
    queue="*"
fi

printOut "    Queue(s):              $queue."
printOut "                                  "



####################################
### Verify Queue Manager status. ###
####################################

printOut "                                "
printOut "Displaying Queue Manager status:"
printOut "                                "

dspmq  -m $qmgr

if [ $? -ne 0 ]
then
    printOut "ERROR:  Queue Manager not available!!!"
    printout "ERROR:  Return Code = 3.              "
    printOut "                                      "
    exit 3
fi



##############################################################################
### Discover Queue Manager Data Path.                                      ###
### (If not the default value of /var/mqm/qmgrs).                          ###
### -------------------------------------------------------------------- ###
### Note: The "`" tick marks take the value of the expression and load    ###
### the value into the specified variable.                                 ###
### -------------------------------------------------------------------- ###
###     grep Prefix=          ==>  Select only lines with "Prefix=".      ###
###     grep $qmgr            ==>  Select only the path for this Qmgr.    ###
###     sed /DefaultPrefix/d  ==>  Remove the "DefaultPrex" line.         ###
###     sed s/^.*Prefixi=//   ==>  Delete beginning of line to "Prefix=". ###
##############################################################################
```

```
dataPath=`cat /var/mqm/mqs.ini | grep Prefix=                \
                               | grep $qmgr                  \
                               | sed ' /DefaultPrefix/d'     \
                               | sed 's/^.*Prefix=//'        \
`

if [ $? -eq 0 ]
then
    printOut "                                       "
    printOut "    Queue Manager Data Path: $dataPath."
    printOut "                                       "
else
    printOut "ERROR:  Unable to determine Queue Manager Data Path!!!"
    printOut "ERROR:  Return Code = 4.                              "
    exit 4
fi



##############################################################################
### Discover Queues.                                                       ###
### ---------------------------------------------------------------------  ###
###     sed  /Starting MQSC/d  ==>  Remove MQSC start line.                ###
###     sed  /AMQ8409/d        ==>  Delete WMQ message lines from results. ###
###     sed s/^.*QUEUE(//      ==>  Delete beginning of line to "QUEUE(".  ###
###     sed s/).*$//           ==>  Delete from ")" to end of line.        ###
###     sed /^SYSTEM[.]/d      ==>  Remove SYSTEM. queues from list.       ###
###     sed /^KMQ[.]/d         ==>  Remove Tivoli queues (KMQ.*)           ###
###     sed /^KQI[.]/d         ==>                        (KQI.*)          ###
###     sed /^AMQ[.]/d         ==>  Remove MQExploere queues (AMQ.*).      ###
###     sed s/[.]/!/g          ==>  Change "." characters to "!".          ###
##############################################################################

printOut "                                   "
printOut "Discovering Queues:                "
printOut "    Queue List in file:  $TempName."
printOut "                                   "

echo "dis ql($queue) usage" | runmqsc $qmgr | grep -i queue | sed '/Starting MQSC/d'  \
                                                            | sed '/AMQ8409/d'        \
                                                            | sed 's/^.*QUEUE(//'     \
                                                            | sed 's/).*$//'          \
                                                            | sed '/^SYSTEM[.]/d'     \
                                                            | sed '/^KMQ[.]/d'        \
                                                            | sed '/^KQI[.]/d'        \
                                                            | sed '/^AMQ[.]/d'        \
                                                            | sed 's/[.]/!/g'         \
      >$TempName

if [ $? -ne 0 ]
then
    printOut "ERROR: Queue listing failed!!!"
    printOut "ERROR: Return Code = 5.       "
    exit 5
fi



##########################################################
### Display the Buffer size for each Queue discovered. ###
##########################################################

while  read  record
do
    echo $record
    ./qtune.ksh -d $dataPath/qmgrs/$qmgr/queues/$record

    if [ $? -ne 0 ]
    then
        printOut "ERROR: 'qtune.ksh' program execution failed!!!"
        printOut "ERROR: Return Code = 5                        "
        exit 5
    fi

done  <$TempName



###############################
### Display ending message. ###
###############################

printOut "                                    "
printOut "$ScriptName finished.  No errors."
exit 0
```