MQ on z/OS - Vivisection

Lyn Elkins – elkinsc@us.ibm.com
IBM Advanced Technical Skills
Agenda

• One of these things is not like the other
• How are messages stored?
  • Private Queues
  • Shared Queues
• First line managers - the components of a z/OS queue manager
• What happens on a API call?
• Summary
One of these things is not like the other

- MQ is NOT a database

Application:
- MQGET – Input Queue
- EXEC SQL INSERT
- EXEC SQL QUERY
- EXEC SQL QUERY
- MQPUT – Reply queue

Database
Shared Queue Message Storage

- To be covered in detail in session *Getting the best availability from MQ on z/OS by using Shared Queues*, Friday at 11:00 am
First Line Managers – who does the real work

• To provide the qualities of service that are the basis for WMQ, the real work within the queue manager is divided into logical ‘workers’ or manager that interact with the underlying z/OS resource managers.

• They are:
  • Connection Manager
  • Recovery Manager
  • Log Manager
  • Message Manager
  • Topic manager
  • Data Manager
  • Buffer Manager
  • Lock Manager
  • Storage Manager
  • CF Manager
  • Security Manager……
Building Blocks - Resource Managers

Connection Manager → Commit Manager → Recovery Manager → UR → Log Manager

MQI → Lock Manager → Rise → Ckpt

MQOPEN → Message Manager → Data Manager → Buffer Manager

CF Manager → Redo Undo → Ckpt
Handling Applications - Connection Manager

Control Blocks

Logs

Data

Other RMs

Application

MQCONN

MQOPEN

MQPUT

...

SYNCPOINT

COORDINATOR

Connection manager

PHASE 1

PHASE 2

EOT

EOM
Getting requests into WMQ - Stubs and Adapters

CICS Region

Appl
DFHMQSTB
RMI TRUE

Batch/TSO

Appl
CSQBSTUB or.
Adapter

IMS Regions

MPP/BMP/FP
Appl
MQI Stub
Control
ESAF exits

Queue Manager

Program calls (QRPL)
Controlling the MQI and MQSC - Message Manager

Command, Runtime, Group servers

MQSC
CONSOLE
QSG

Application

MQOPEN
MQCLOSE
MQGET
MQPUT
MQPUT1
MQINQ
MQSET

MQI

Message Manager

SYSTEM.CLUSTER

Objects

MQOPEN

BASE

Validation
Consistency
Triggering
Get-Wait

Complete your sessions evaluation online at SHARE.org/SFEval
Buffer Manager – High Performance storage and retrieval

DEFINE BUFFPOOL(bpid)
BUFFERS (nnnn)

buffer pages

STEAL,
LRU

steal

read pageset

write pageset

dirty

CKPT,
STEAL,
WRITE AHEAD

dirty

NO FORCE

dirty

OLDEST

DEFINE PSID(psid)
BUFFPOOL(bpid)

force dirty pages > 3 ckpts
Providing Logging Interfaces - Log Manager

- Log read and write functions
- Log Shunting
- Multiple active log data sets and archive
- Archive inventory management
- Duplexed for reliability
- “Bootstrap” file
  - End of log location
  - Archive inventory
- Various Utilities
Concurrency and Isolation - Lock Manager

MQOO_INPUT_SHARED

MQOO_INPUT_EXCLUSIVE

MQOPEN

vs

DELETE QLOCAL

request
commit
allocation
Shared Message Queue Storage Using CF List Structures

- committed puts
- uncommitted puts
- queue details

List Structure

MQPUT

MQGET

MQCMIT

SQ1 LH
SQ2 LH
SQM1 LH

Keys
Data

State
Priority
Time
Qmgr

MsgId/CorrelId

WRITE
READ
MOVE
DELETE
MONITOR

uncommitted gets
expired messages
Scenario – Persistent MQPut to a Triggered Queue

<table>
<thead>
<tr>
<th>Application</th>
<th>Message Manager</th>
<th>Data Manager</th>
<th>Buffer Manager</th>
<th>Recovery Manager</th>
<th>Log Manager</th>
<th>Lock Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQOPEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ACQUIRE LOCK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOCATE QUEUE IN HASH TABLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECURITY</td>
<td>BASE NAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQUIRE HANDLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MQPUT</td>
<td>USE HANDLE</td>
<td>LOCATE PAGE TO HOLD MSG</td>
<td>BUFFER PAGE</td>
<td></td>
<td>START UR</td>
<td>LOG RECORDS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHECK TRIGGER RULES</td>
<td>START UR</td>
<td>LOCATE PAGE TO HOLD MSG</td>
<td>BUFFER PAGE</td>
<td></td>
<td>LOG RECORDS</td>
<td>LOG RECORDS</td>
</tr>
<tr>
<td>MQCMIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FORCE LOG</td>
<td>RELEASE LOCKS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Scenario - MQGet from a Queue

<table>
<thead>
<tr>
<th>Application</th>
<th>Message Manager</th>
<th>Data Manager</th>
<th>Buffer Manager</th>
<th>Recovery Manager</th>
<th>Log Manager</th>
<th>Lock Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQOPEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ACQUIRE LOCK</td>
</tr>
<tr>
<td></td>
<td>LOCATE QUEUE IN HASH TABLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECURITY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BASE NAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQUIRE HANDLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MQGET</td>
<td>USE HANDLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FIND MSG (INDEX / NEXT)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BUFFER PAGE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>START UR</td>
<td></td>
<td></td>
<td>LOG RECORDS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOG RECORDS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MQCMIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FORCE LOG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RELEASE LOCKS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary

• Delivers transactional messaging
  • Enables robust business applications

• Complex, but well organised
  • Adapters, Address spaces, Resource Managers

• Designed for throughput, availability and scalability
  • Logging, Buffering, Locking, Communications
Shameless Promotion

- New RedBook covering what’s new in WMQ V7.1 and 7.5
<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></td>
<td></td>
<td>Are you running too many queue managers or brokers?</td>
</tr>
<tr>
<td>09:30</td>
<td></td>
<td>What's New in WebSphere Message Broker</td>
<td></td>
<td></td>
<td>Diagnosing Problems for MQ</td>
</tr>
<tr>
<td>11:00</td>
<td>Extending IBM WebSphere MQ and WebSphere Message Broker to the Cloud</td>
<td>WMQ - Introduction to Dump Reading and SMF Analysis - Hands-on Lab</td>
<td>BIG Data Sharing with the cloud - WebSphere eXtreme Scale and WebSphere Message Broker integration</td>
<td></td>
<td>CICS and WMQ - The Resurrection of Useful</td>
</tr>
<tr>
<td>12:15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01:30</td>
<td>Introduction to MQ</td>
<td>MQ on z/OS – Vivisection</td>
<td>Migration and maintenance, the necessary evil</td>
<td>The Dark Side of Monitoring MQ - SMF 115 and 116 Record Reading and Interpretation</td>
<td></td>
</tr>
<tr>
<td>03:00</td>
<td>First Steps With WebSphere Message Broker: Application Integration for the Messy</td>
<td>BIG Connectivity with WebSphere MQ and WebSphere Message Broker</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>WebSphere MQ application design, the good, the bad and the ugly</td>
<td>What's New in the WebSphere MQ Product Family</td>
<td>MQ &amp; DB2 – MQ Verbs in DB2 &amp; Q-Replication</td>
<td>WebSphere MQ Channel Authentication Records</td>
<td></td>
</tr>
<tr>
<td>06:00</td>
<td></td>
<td></td>
<td>Clustering - The Easier Way to Connect Your Queue Managers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This was session MQ Internals - The rest of the week ……