WebSphere MQ for z/OS
Restart and (Disaster) Recovery

Paul Kettley
PAULK@uk.ibm.com

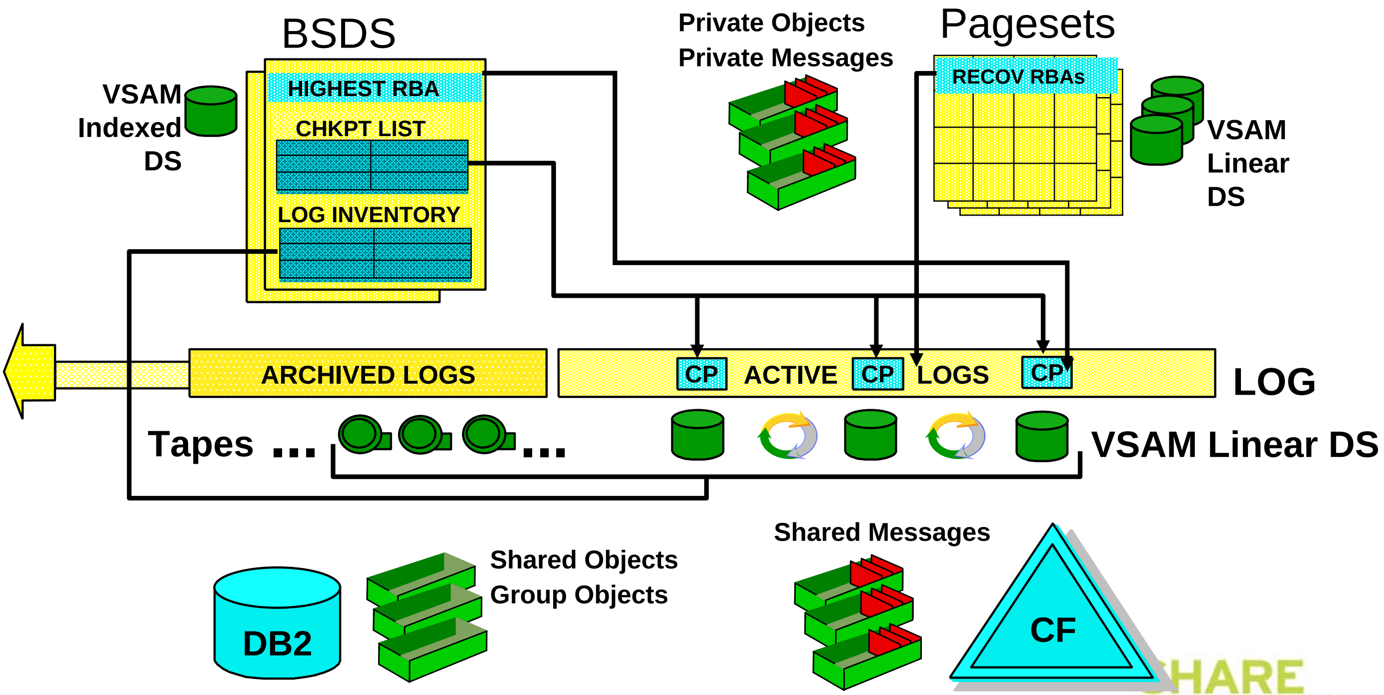
Presentation Positioning

- It is not the intention of this presentation to go into every detail about how to backup and recover a queue manager.
- This presentation should be used as an overview of how WebSphere MQ for z/OS recovery is designed.
- For detailed steps on how to perform backup and recovery, please refer to the product documentation.

Agenda

- The MQ Subsystem
  - What makes up a Queue Manager?
  - Backing up a Queue Manager
  - What makes up a Queue Sharing Group?
  - Backing up a Queue Sharing Group
- Restart/Recovery Processing
  - Restart processing
  - Log Shunting
  - Index queue rebuild
  - Pub Sub status queues
  - Peer Recovery
  - Structure Recovery
- Recovery Scenarios
  - Local restart
  - Damaged pageset
  - DR without mirrored DASD
  - DR with QSG

What makes up a Queue Manager?



What makes up a Queue Manager?

- Queue manager started task procedure
- zParm module
- BSDS
- Active and Archive Logs
- Pagesets

What makes up a Queue Manager?

- **Queue manager started task procedure**
  - **Specifies the MQ libraries to use, location of BSDS and pagesets and INP1, INP2 members to process during start up**
- **System Parameter Module – zParm**
  - **Configuration settings for logging, trace and connection environments for MQ**
- **Both of these items are fairly static, so only need to be backed up when changes are made**

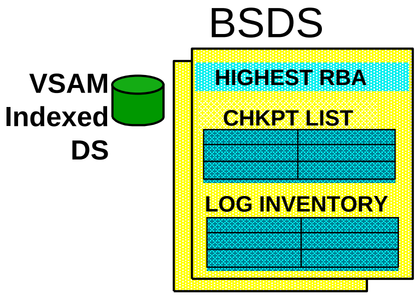
What makes up a Queue Manager?

NOTES

- The queue manager's started task procedure specifies the MQ product libraries that are going to be used by the queue manager. It is important that these product libraries are backed up, and also available on any DR site. Make sure that you refresh the DR copy when applying maintenance to the production site.
- In addition to the MQ product libraries, the started task procedure also specifies the locations of the BSDS, pagesets and the INP1 and INP2 datasets. To ensure that the "same" queue manager can be started on a DR site, the same started task procedure, and all the datasets that is references need to be available. We will looking in more detail in the following slides what this means specifically for the BSDS and pagesets. The INP1 and INP2 datasets should be backed up, and copied to the DR site when they are updated
- The system parameter module (zPARM) contains configuration settings that the queue manager loads during start up processing. The settings include details about the logging, trace and connection environments. This module needs to be available on a DR system, so should be copied whenever it is updated.

What makes up a Queue Manager?

- **BSDS – Boot Strap Data Set**
  - Contains information about log RBAs, checkpoint information and log dataset names
  - Queue Manager can maintain dual copies of BSDS
  - Copy of BSDS taken when logs are archived
  - Vital for Queue Manager start up



SHARE in Anaheim 2011

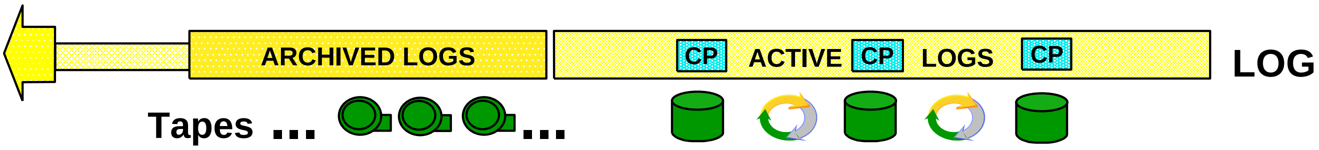
What makes up a Queue Manager?

NOTES

- The BSDS is the queue manager's entry point into the logs that it maintains. The BSDS contains details about when checkpoints have been taken by the queue manager and also the log RBA ranges that are in each log dataset, either active or archive.
- The queue manager is able to maintain two copies of the BSDS, providing for local redundancy of the BSDS. When an archive log is written, a copy of the BSDS is also taken at the same time. Archive log datasets end in .Axxxxxxx, the BSDS archive ends in .Bxxxxxxx (where xxxxxxx is an incrementing number). In a DR situation, the archived BSDS can be restored into the main BSDS using REPRO.
- The BSDS is vital for the start up of the queue manager, without it, the queue manager doesn't know what logs to use, and what RBAs are in each log dataset.

What makes up a Queue Manager?

- **Active and Archive Logs**
  - Contain records of all recoverable activity performed by the Queue Manager
  - Queue Manager can maintain dual copies of logs
  - Vital for Queue Manager start up
  - Active log dataset offload to Archive (if enabled) when it is full, BSDS also offloaded at the same time
  - Use display usage to show lowest RBA needed for restart



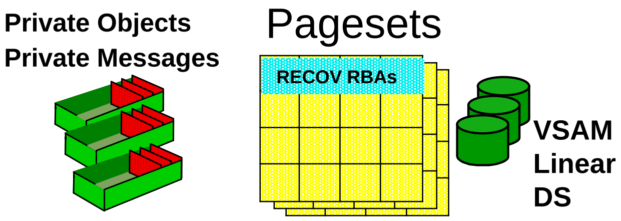
What makes up a Queue Manager?

N O T E S

- The active and archive logs are the primary record of recoverable activity performed by the queue manager. When recoverable activity occurs, it is always written to the active logs before it is written anywhere else (pagesets for example). As with the BSDS, the queue manager is able to maintain dual copies of the active and archive logs (in the case of the archives these could be written to different media, tape and DASD for example).
- Having copies of the active and archive logs it is possible to restore the queue manager to its most recent state, even after a media failure of the pagesets. Specifically, it is necessary to have copies of the logs from 3 checkpoints before the pagesets were backed up (assuming fuzzy backup is performed).
- The DISPLAY USAGE command may be used to determine the lowest RBA that is required to restart the queue manager (excluding any log range needed for media failure). This RBA will ensure that the oldest unit of work that is un-resolved can be backed out.

What makes up a Queue Manager?

- **Pagesets**
  - **Updates made “lazily”**
    - **bufferpool and logs are primary data source for queue manager**
  - **Brought “up to date” from logs during queue manager restart**
  - **Start up with an old pageset (restored backup) is not really any different from start up after queue manager failure**
  - **Backup needs to copy page 0 of pageset first (don’t do volume backup!)**



What makes up a Queue Manager?

NOTES

- Pageset updates are made “lazily”, if the updates are recoverable (persistent messages) then they are always written into the logs before the pagesets. Updates are also made in the bufferpools before the pagesets. After a failure the pagesets can be brought up to date using the data contained in the logs.
- After a failure and before restart, the pagesets may not contain all the data that has been written to the MQ logs. When going through start up processing, the queue manager will always look at the state of the pagesets and bring them up to the latest queue manager state. This means that if a pageset backup needs to be restored, the handling in the queue manager is very similar to after a failure that hasn’t involved a pageset restore. The difference being in how much log data needs to be available to bring the pageset up to date.
- Because of the way recovery works, it is vital that any pageset backups that are performed, backup page 0 of the pageset first (unless the queue manager is not running when the backup is made). Using REPRO, or ADRDSSU dataset copy will copy page 0 first, don’t however do a volume copy as this will not necessarily copy page 0 first.

Backing Up a Queue Manager

- Backup strategy depends on whether you are catering for local recovery or remote site recovery
- Local Backups
  - Keeps copies of ZPARM, MSTR procedure, product datasets and INP1/INP2 members
  - Use dual BSDS, dual active and dual archive logs
  - Take backups of your pagesets, this can be done while the queue manager is running (fuzzy backups)
    - Make sure you backup Page 0 first, REPRO or ADRDSSU logical copy

SHARE in Anaheim 2011

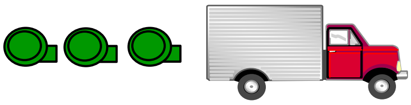
Backing Up a Queue Manager

N O T E S

- The backups that are required depend on whether you are preparing for local site recovery, or remote site disaster recovery. Of course, they may be a large overlap in what is being backed up in each case.
- In the case of local recover, the zPARM module, queue manager started task procedure, product datasets and INP1 and INP2 members should all be backed up. These change relatively infrequently, so should be backed up when they are modified (of course, nothing stopping you backing them up more often).
- Assuming that the queue manger is using dual BSDS and logs then it wouldn't normally be necessary to take additional backups of these for local recovery. Of course, you should ensure that different devices are used for each copy.
- Pagesets should be backed up in case of media failure/corruption. The backups may be taken while the queue manager is running, in which case the backup must copy page 0 of the pageset first.

Backing Up a Queue Manager …

- Remote Site Recovery
  - DASD mirroring
    - Recovery at remote site effectively the same as local site
    - If async mirroring, queue manager may have lost some data (depending on latency), having effectively been set back a period of time (typically a few seconds)
  - PTAM (Pick Truck Access Method)
    - Data loss will occur!
    - Recovery is performed using backups of pagesets and archive log datasets
    - Active logs are not used (not copied), new active log datasets defined and BSDS recreated from archive copy



Backing Up a Queue Manager

N

O

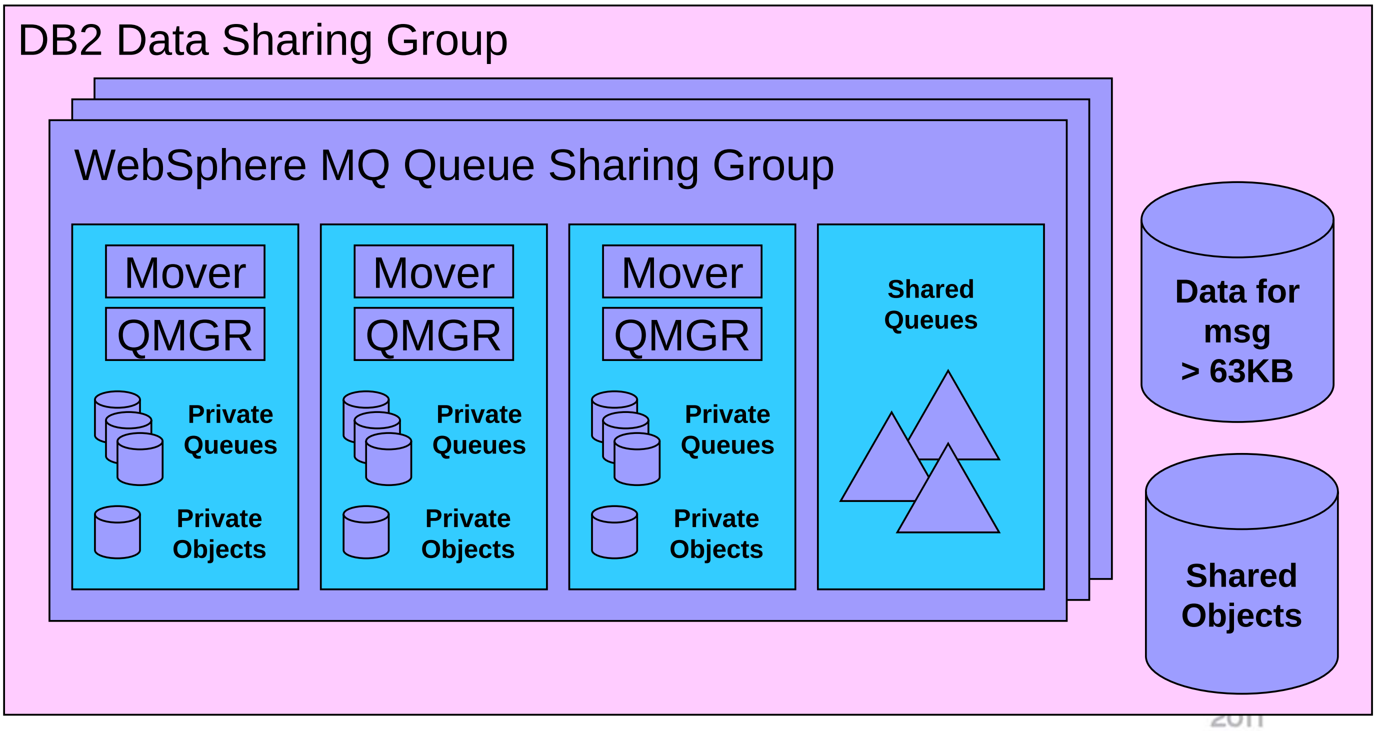
T

E

S

- Remote site DR using mirrored DASD is effectively the same as local site recovery. Assuming all the datasets used by the queue manager are mirrored to the remote site, there should be an exact copy of all the data that makes up the queue manager. With this available, the remote site start up is no different from a primary site startup.
- Performing DR recovery without using mirrored DASD will lead to data lose. In this scenario the archive logs are copied to the DR site, along with pageset backups. Any restart that is going to take place uses the archive logs, but not the active logs from the primary site. This will lead to data loss occurring as not all of the queue manager data is available on the remote site. During recovery, the BSDS needs to be recreated from the archived copy, and new active log datasets created and added to the BSDS. The queue manager will then restart using the restored pagesets and the archive log datasets.

What makes up a Queue Sharing Group?



What makes up a Queue Sharing Group?

N O T E S

- Chart shows how queue managers are organized into queue-sharing groups (QSGs) and the relationship to DB2 data-sharing groups.
- A queue-sharing group can contain one or more queue managers:
  - Each queue manager has its own private (not shared) queues and object definitions.
  - All queue managers in a QSG share the same set of shared queues and shared object definitions
  - A queue manager cannot belong to more than one QSG.
- Shared object definitions for a QSG are maintained for WebSphere MQ by DB2. Shared access to these definitions is by DB2 data sharing:
  - You must have DB2
  - You can have more than one data-sharing group, but all members of one QSG must be members of the same data-sharing group
  - Shared object definitions are cached in the queue managers.
  - A DB2 outage does not bring down the QSG (but you cannot add or change shared objects if DB2 is down).
- You do not have to define any queue-sharing groups if you do not run a sysplex (or if you just don't want to).
- If using shared messages > 63KB then a small portion for the message is stored in the CF, and the rest is stored in DB2.

What makes up a Queue Sharing Group?

- DB2 Configuration information & Group Object Definitions (in DB2)
  - DB2 is used to store configuration and status information for the QSG, this includes things like when backups have been taken
  - DB2 is also used to hold the Group and Shared object definitions
- This DB2 data should be backed up as part of the DB2 backup procedures

What makes up a Queue Sharing Group?

NOTES

- MQ uses DB2 to store various configuration information for the QSG. For example, the shared queue definitions are in DB, along with any group object definitions. DB2 is also used to hold things like details of which queue managers are in which QSGs, and when backups were taken for recoverable structures.
- This presentation is not designed to cover DB2 backup and recovery, so please refer to the appropriate material for DB2.
- DB2 should be backed up, and in a DR situation needs to be recovered before any queue managers are started.

What makes up a Queue Sharing Group?

- Coupling Facility Structures
  - **Administration and Application structures, which hold QSG control information and MQ messages**
  - **Recoverable Application structures should be backed up on a regular basis, these backups are made in the logs of the queue manager where the backup was issued**
  - **CFRM policy controls size and attributes of structures, needs to be replicated on DR site (be careful about changes in CFCC microcode levels)**

What makes up a Queue Sharing Group?

NOTES

- MQ uses Coupling Facility List Structures for the administration and application structures. The administration structure is used to hold status information that is used by the queue managers in the QSG to coordinate work and manage recovery. The application structures are where actual MQ messages are stored.
- If the application structures are set to RECOVER(YES) in the MQ CFSTRUCT definition, then they should be backed up on a regular basis (regular is generally 30-60mins). The backup of a structure is made into the logs of the queue manager where the BACKUP CFSTRUCT command has been issued. Taking regular backups means that persistent messages can be recovered after a structure failure, or DR site move, in a timely manner.
- When moving to a DR site it is important that the CFRM policy is up to date with respect to what structures can be defined. Also, care needs to be take about structure sizes, particularly if the CF in the DR site is running at a different CFCC level than the primary site. The amount of structure space available to applications varies between different CFCC levels, so while a structure on the primary site may have enough space, in the DR site, if running with a higher CFCC level, it might not have enough space available to MQ.

Restart Processing

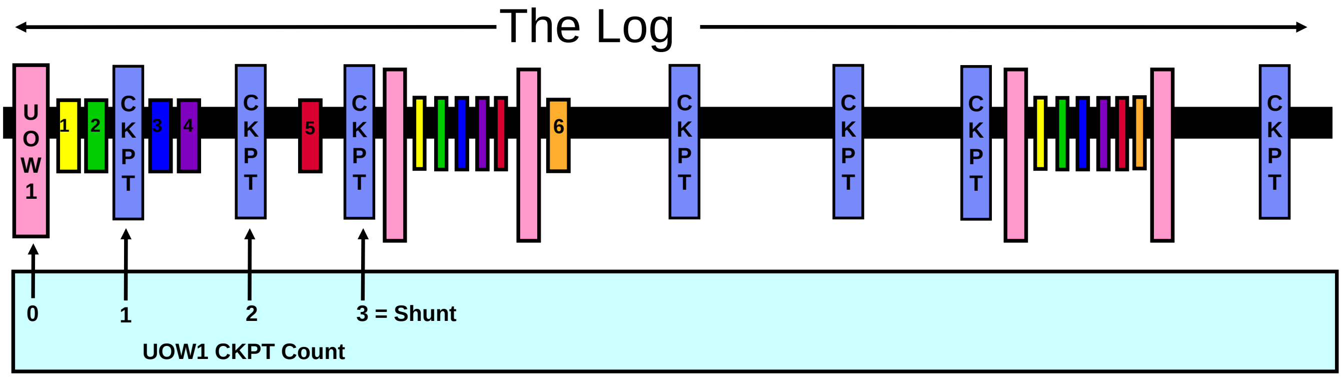
- Establish recovery log range – based on:
  - Media recovery of pagesets
    - Typically just cached changes from abnormal termination
    - But may include media recovery after pageset damage and restore from backup
  - Unit of work recovery
    - Long running in-flight or in-doubt units of work
- Read log forward
  - Media recovery and locks on in-doubt private messages
- Read log backwards
  - Undo for in-flight and in-backout units of work

Restart Processing

NOTES

- Restart processing is made up of a couple of phases. Firstly the recovery log range is established, this is based on the media recovery required for the pagesets. It might just be applying cached changed after failure, or might include media recovery after a pageset is damaged and restored from a backup. The oldest log record that is required for backing out units of work is also used to calculate the recover log range.
- Once the log range has been determined, forward log reading is carried out to bring the pagesets up to the latest state. At the same time, locks are also taken on any messages that are related to in-doubt or in-flight UOWs.
- Once forward log reading has been completed the logs are read backwards to backout any UOWs that were in-flight or in-backout at the time of failure.

Log Shunting



- With log shunting, most restarts need to go back a maximum of 3 checkpoints
- Restart within active log range

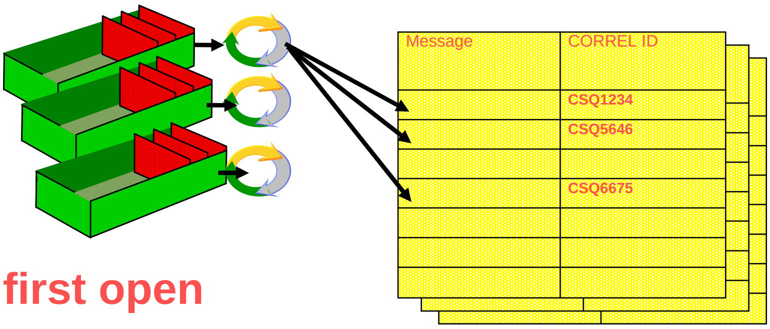
Log Shunting

NOTES

- Log shunting is a technique that is used to minimize the amount of log data that needs to be read during restart processing. This is not applicable for media recovery (bringing pagesets up to date), but is used for backout processing. It is particularly useful when long running UOWs are involved.
- A UOW is deemed to be long running when it has gone through 3 checkpoints. At this point it will be scheduled for “shunting”. As part of the shunting process a copy of the log records relating to the UOW are written at current end of log. Not all the data from the log records is used, only enough necessary to perform backout processing. This means, that if the UOW needs to be backed out, it is only necessary to go back to the most recent shunt for the UOW, not to the beginning of the UOW. Every third checkpoint that the UOW goes through it will be scheduled for shunting again.

Queue index rebuild

- Private queues can be indexed
  - In storage index by MsgId or CorrelId
  - Support Get specific
- Index rebuilt at restart or first open
  - Every message on the queue must be read – pageset I/O
  - Potentially a long time for deep queues
- Usually application queues, CHINIT uses
  - SYSTEM.CHANNEL.SYNCQ
  - SYSTEM.CLUSTER.TRANSMIT.QUEUE



Message	CORREL ID
	CSQ1234
	CSQ5646
	CSQ6675

Queue index rebuild

N O T E S

- Any queue that has in INDXTYPE specified will need to have the index rebuilt before messages can be accessed. This can either be done during restart processing, before the queue manager is available for any connections, or may be completed after restart has completed. Using the CSQ6SYSP option of QINDXBLD=NOWAIT the queue manager can complete startup before all the indices have been rebuilt. Before an application is able to access the messages on the queue it will need to have the index rebuilt, so it may experience a delay when accessing the queue, if the index hasn't been fully built at that time.
- To rebuild the index it is necessary to read the MQMD of every message on the queue, so if there are lots of messages this might take a significant amount of time. This will also cause pageset I/O as the messages need to be read off of disk.

Pub Sub Status Queues

- Information about durable subscribers and retained publications are kept on queues
  - SYSTEM.DURABLE.SUBSCRIBER.QUEUE
  - SYSTEM.RETAINED.PUB.QUEUE
- These need to be read before Pub Sub can be used
- Typically relatively few retained message
- Could be lots of durable subscribers
  - Restart processing optimised, 1 million subscribers measured at approx 230 CPU seconds

Pub Sub Status Queues

NOTES

- As part of restart processing, the pub sub status queues are read, and the information contained in them is reloaded.
- SYSTEM.RETAINED.PUB.QUEUE contains copies of the retained messages that have been stored for each topic. Typically there will be a fairly small number of these, each one is read, and details of it stored against the topic node in the queue manager.
- SYSTEM.DURABLE.SUBSCRIBER.QUEUE is used to store the subscriptions for durable subscribers, this enables them to be remade after a queue manager restart. The technique used to store these subscribers has been designed to enable a fast reload of subscribers, although, because there could be a very large number of subscribers this may still take some time to completed. In tests in the lab, 1 million durable subscribers took approximately 230 CPU seconds to reload.

Peer Level Recovery (QSG)

- In the event of Queue Manager outage
  - Performed by peer QMs on notification of structure abnormal disconnection
- UOW recovery
  - Commit in-commit UOWs
  - Backout in-flight and in-backout UOWs
- Queue recovery
  - Release open interest on queues
- Peer Channel restart (shared channels only)
  - Outbound channels from Chinit are restarted on survivors
  - Inbound channels will retry and be reassigned

Peer Level Recovery

N O T E S

- In the event of a queue manager outage in a QSG, peer level recovery enables the other queue managers that are left running in the QSG to resolve work for the failed queue manager in a timely manner. Each queue manager in the QSG gets notified by the CF about the abnormal disconnection from a structure by another queue manager. Any units of work that are in-commit can be completed and any in-flight or in-backout at the time of failure can be backed out. This enables the UOWs to be finished before the failed queue manager is started up again, and hence giving faster access to messages that were part of the UOWs.
- As part of the PLR processing, open interests on queues can also be resolved.
- For shared channels, these can be restarted as necessary on a remaining queue manager.

Structure Recovery (QSG)

- Admin Structure
  - Each QM logs it's own recoverable updates
  - Automatically rebuilds own entries from log
- Application Structure
  - BACKUP CFSTRUCT used periodically to archive contents
    - From any QM in group (including dedicated if necessary)
    - Recommend frequent backups – think hourly not daily
  - RECOVER CFSTRUCT applies most recent backup and forward recovers using merged logs

Structure Recovery

NOTES

- Any recoverable changes that are required to the admin structure will be logged automatically by each queue manager. Then, in the event that the structure needs to be recovered the queue manager logs may be used to rebuild the entries. From V7.0.1 any queue manager is able to rebuild the entries for any other queue manager in the QSG. Prior to this point it was necessary for each queue manager to rebuild its own entries.
- When a recoverable update is made to a shared queue, it is logged by the queue manager that is making the update. This means that the put of a message may be logged by a different queue manager than the one doing the get. It is advisable to take backups of the recoverable structures on a regular basis (30-60minutes is typical). In the event of a failure and recovery being needed, the time since the last backup is the determining factor for how long the recovery will take. When recovery takes place, the logs from all of the queue managers that have accessed the structure are read and merged together to get a single view of what has been done to the structure.

Recovery Scenario 1 – Local Restart

- Scenario: Queue manager accidentally gets cancelled by an operator
- All persistent data was written to logs, so although it may not have been written to the pageset yet, it is still preserved
- Queue manager is restarted locally.
  - Establishes recovery range, based on recovery RBA of pagesets and active UOWs at time of failure
  - Reads logs forwards to recover pagesets to point of failure
  - Reads logs backwards to backout any in-flight UOWs

Recovery Scenario 1 – Local Restart

NOTES

- This is the most basic recovery scenario that we will look at, and is used as a building block for the other scenarios which are slightly more complicated!
- The queue manager has experienced a failure, in this case because it was cancelled, although it could be any failure that has caused it to terminate. As there wasn't any loss or damage to datasets, the queue manager may simply be restarted in place. When the queue manager is started it will determine how much log data needs to be read to bring the pagesets up to the latest status, and also what range is required to backout any units of work that were in-flight or in-doubt.

Recovery Scenario 2 – Pageset damage

- Scenario: DASD device suffers a fatal error! Queue manager terminates
- Restore most recent backup of damaged/lost pageset
- Restart queue manager
  - In addition to the recovery that queue manager performed in scenario 1, it will recover the pageset up to the most recent point in logs
  - Media recovery log range will be from slightly before when backup was taken
    - This may well mean that recovery takes a very long time! Depends on backup frequency

Recovery Scenario 2 – Pageset Damage

N O T E S

- This scenario builds on the previous one in that a pageset has been damage, and hence has to be recovered from a backup. The damaged pageset is deleted and then the backup is copied to the pageset location specified by the queue manager started task procedure. When the queue manager is started it will detect that an old pageset is in place and determine what recovery range is needed to bring the pageset up to date. Depending on how old the backup was it may take a large amount of time to recover the pageset data. It is advisable to backup your pagesets at least daily.

Recovery Scenario 3 – DR without mirrored DASD

- Scenario: Primary site disaster, causing DR plan to be put in place and moving to backup site.
- Most recent pageset backup restored
- BSDS restored from most recent BSDS archive
- New active logs defined
- BSDS details updated with new active log datasets
- Queue manager restarted
  - Amount of data loss that occurs is dependant on frequency that archive logs are copied to DR site, and frequency that archives are written

Recovery Scenario 3 – DR without mirrored DASD

NOTES

- This is perhaps the most complicated scenario that we will look at. There has been a disaster that has caused the DR plan to be put in place. The queue managers that were running on the primary site need to be started on the DR systems. Because mirrored DASD isn't being used, there are no copies of the active logs available to start the queue manager. In this case, all that can be done is start up the queue manager using the archive logs, and accept any data loss that will occur because of not having the active logs. The amount of data lost depends on how much data was written by the queue manager after the last archive log was copied to the DR site.
- In this scenario it is important to have a backup copy of the pagesets, taken before the last archive log that is available. The BSDS is recreated by copying from the archived BSDS that was taken with the last archive log. The active log information is deleted from the BSDS, and new active logs are created and defined in the BSDS. The pageset backups are restored, and then the queue manager is started. The queue manager will effectively be unaware that anything happened after the last archive log, so any puts will effectively not have happened! Similarly with gets.

Recovery Scenario 4 – DR with QSG

- Scenario: Primary site disaster, causing DR plan to be put in place and moving to backup site. Queue managers in a QSG, and using DASD mirroring
- Restore DB2 subsystems, ensuring all locks resolved
- Ensure that the CF structures for the QSG are not allocated (force the structures if needed)
- Start up a single queue manager.
  - As DASD mirroring is in use, this will be just like starting locally (assuming you have copied everything!)
  - Queue manager will recover its own Admin Structure entries.
  - If at V7.0.1 it will also recover the Admin Structure entries for the other queue managers in the QSG
  - If not at V7.0.1 then each queue manager will need to be started to rebuild Admin Structure
  - Once Admin Structure has been rebuilt, application structures can be recovered

SHARE in Anaheim 2011

Recovery Scenario 4 – DR with QSG

NOTES

- In this scenario the DR plan is put into place, but also using a QSG. However, mirrored DASD is being used, which simplifies matters! The first thing to do before attempting to recover any queue managers (although prep work can be done) is to recover the DB2 data sharing group. It is important that this is sorted out first, and that there are no retained locks held by DB2.
- Before starting the queue managers, also ensure that the CF structures are not allocated (unless the structures were duplexed with the primary site!). MQ will recover the CF structures for the log data and any old structures (from a previous DR test) will complicate the recovery. As mirrored DASD is being used, the starting up of each queue manager should be just like a local restart, so it would not be necessary to modify the BSDS etc before starting up. If at V7.0.1 a single queue manager can be started, which will recover the admin structure for the whole QSG. If there are no V7.0.1 queue managers in the QSG then it will be necessary to restart every queue manager in the QSG to complete recovery of the admin structure. These queue managers can all be started on the one LPAR, and once started, can be shutdown again, it is purely to recover the admin structure. Once the admin structure has been rebuilt, it is then possible to recover the application structures.