Unit 15.WebSphere MQ high availability

What this unit is about

This unit covers methodologies available to use WebSphere MQ with high availability (HA) systems

What you should be able to do

After completing this unit, you should be able to:

- Describe the steps needed to switch a running queue manager between two high availability (HA) systems
- Plan for using HA systems with WebSphere MQ

How you will check your progress

- Checkpoint quiz

References

WebSphere MQ V7 Information Center

Unit objectives

After completing this unit, you should be able to:

- Describe the steps needed to switch a running queue manager between two high availability (HA) systems
- Plan for using HA systems with WebSphere MQ

Figure 15-1. Unit objectives WM203 / VM2032.0

Notes:

High availability overview topic objective

After completing this topic, you should be able to:
- Describe the concept and purpose of high availability

Figure 15-2. High availability overview topic objective

Notes:

Objectives and terminology

- The objective is to achieve 24x7 processing of all messages
- Avoid application awareness of availability solutions
- Not always achievable but can get close – how many 9's ?

Availability % age	Downtime per year
99	3.65 days
99.9	8.76 hours
99.99	52.6 minutes
99.999	5.26 minutes
99.9999	30.00 seconds

Figure 15-3. Objectives and terminology

Notes:

Potential outage types

- 80% scheduled downtime:
  - New software release
  - Upgrades
  - Maintenance
- 20% unscheduled downtime:
  - 40% operator error
  - 40% application error
  - 20% other (network failure, disk failure, power outage)

Figure 15-4. Potential outage types

WM203 / VM2032.0

Notes:

Some terms

- Fault tolerance and continuous availability
  - Cope with almost any type of failure without interruption to service
- High availability
  - Service can be interrupted but reappears quickly with no manual intervention
- Disaster recovery
  - Recovery from large system or site loss
- Scalability
  - The ability to add extra servers to achieve extra throughput
- Workload balancing
  - Spread the load across multiple servers
- Clusters (lots of meanings)

Figure 15-5. Some terms

Notes:

High availability overview topic summary

Having completed this topic, you should be able to:

- Describe the concept and purpose of high availability

Figure 15-6. High availability overview topic summary WM203 / VM2032.0

Notes:

WebSphere MQ HA strategies topic objectives

After completing this topic, you should be able to:
- Describe WebSphere MQ HA strategies

gure 15-7. WebSphere MQ HA strategies topic objectives WM203 / VM2032.0

Notes:

WebSphere

Figure 15-8. WebSphere

Notes:

Some more term

New message: A

Old message: A

WebSphere MQ

The key feature

- Automatic d
- Automatic c
- Automatic r
- Workload b
- Heterogen

WebSphere MQ clusters as a HA strategy

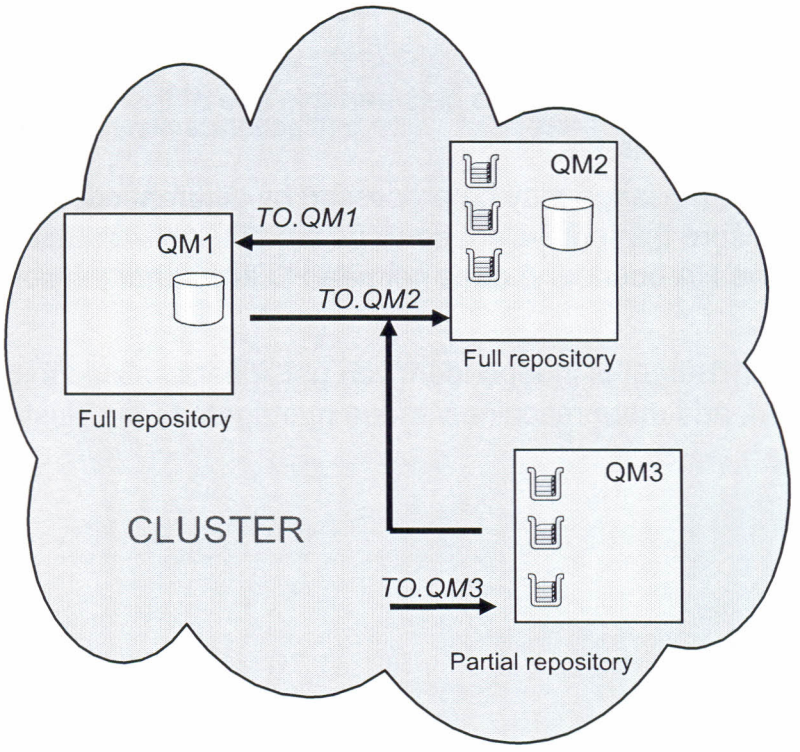

Figure 15-8. WebSphere MQ clusters as a HA strategy

Notes:

Some more terms:

New message: A message sent to a queue manager after the queue manager failure

Old message: A message sent to a queue manager before the queue manager failure

WebSphere MQ clusters provide a part of the solution to high availability.

The key features of WebSphere MQ clusters are:

- Automatic definition of channels
- Automatic discovery of target queues
- Automatic rerouting of messages when a target computer is unavailable
- Workload balancing, with an exit capability to override default choices
- Heterogeneous, available on all V6 and upwards platforms (and some others)

Each queue manager is an independent entity, but they share configuration and topology information using a reliable repository which uses WebSphere MQ messages and queues for its integrity.

While this solution works well for new messages if a back-end computer fails; then any messages which are already on that queue manager (old messages), or which are still in-doubt on the sending computer, are not available.

It needs to be remembered to distinguish between this use of the word *cluster* and any other that you might encounter.

In any scenario where a message may be processed by different queue managers or on different machines, ensure that equivalent services are available, so that the requesting application (outside the HA boundary) does not need to know that the service moves around.

Because each queue manager is independent you get linear scaling - to get more work done by a server farm, add more machines/queue managers to the cluster.

WebSphere MQ clustering HA considerations

- Full repositories should be highly available.
  - All definition changes are propagated by full repositories
  - Recommend at least two full repositories for all clusters regardless of size.
  - Must be fully interconnected

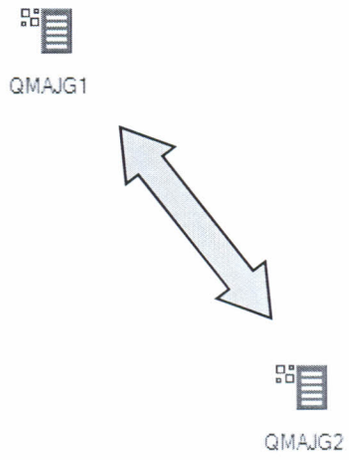

Figure 15-9. WebSphere MQ clustering HA considerations WM203 / VM2032.0

Notes:

Full repositories must be fully connected with each other using manually defined cluster sender channels.

There should always be at least two full repositories in the cluster so that in the event of a failure of a full repository, the cluster can still operate. If there is only one full repository and it loses its information about the cluster, then manual intervention on all queue managers within the cluster are required in order to get the cluster working again. If there are two or more full repositories, then because information is always published to and subscribed for from two full repositories, the failed full repository can be recovered with minimal effort.

Full repositories should be held on machines that are reliable and highly available. This said, if no full repositories are available in the cluster for a short time, this does not affect application messages which are being sent using the clustered queues and channels, however it does mean that the clustered queue managers do not find out about administrative changes in the cluster until the full repositories are active again.

For most clusters, two full repositories are the best number to have. If so, each partial repository manager in the cluster makes its publications and subscriptions to both the full repositories.

It is possible to have more than two full repositories. This may be useful for placing the full repositories in different geographical locations, so that partial repositories use local full repositories.

Backup queue managers

- Problem: Hardware failure => Unrecoverable Queue Manager
- Solution: Backup queue manager
  - Can only be used with linear logging
  - Inactive backup queue manager replays the logs of the active queue manager
  - Logs must be copied to backup location:
  - Can force linear log and script the process
    `RESET QMGR TYPE(ADVANCELOG)`
  - The more frequent to backup, the less data is lost
  - Available on Windows / UNIX systems
  - More a DR solution than a HA solution

Figure 15-10. Backup queue managers

WM203 / VM2032.0

Notes:

An existing queue manager can have a dedicated backup queue manager. A backup queue manager is an inactive copy of the existing queue manager. If the existing queue manager becomes unrecoverable due to severe hardware failure, the backup queue manager can be brought online to replace the unrecoverable queue manager.

The existing queue manager log files must regularly be copied to the backup queue manager to ensure that the backup queue manager remains an effective method for disaster recovery. The existing queue manager does not need to be stopped for log files to be copied, however you should only copy a log file if the queue manager has finished writing to it. Because the existing queue manager log is continually updated, there is always a slight discrepancy between the existing queue manager log and the log data copied to the backup queue manager log. Regular updates to the backup queue manager minimize the discrepancy between the two logs, but the RESET QMGR TYPE(ADVANCELOG) can be used to force the linear log to switch, minimizing to data "left behind" before the log copy.

If a backup queue manager is required to be brought online it must be activated, and then started. The requirement to activate a backup queue manager before it is started is a preventive measure to protect against a backup queue manager being started accidentally. Once a backup queue manager is activated it can no longer be updated.

Creating a backup queue manager

- Creating a backup queue manager:
  - Use **crtmqm**, have the same attributes as the existing queue manager
- Take copies of all the existing queue manager data and log file directories, including all subdirectories
  - Use these to overwrite the backup queue manager directories
- Replay logs on backup queue manager:

```
strmqm -r QmgrName
```


Figure 15-11. Creating a backup queue manager WM203 / VM2032.0

Notes:

Use a backup queue manager only when using linear logging.

To create a backup queue manager for an existing queue manager, do the following actions:

Create a backup queue manager for the existing queue manager using the control command crtmqm. The backup queue manager requires:

- The same attributes as the existing queue manager, for example the queue manager name, the logging type, and the log file size.
- To be on the same platform as the existing queue manager.
- To be at an equal, or higher, code level than the existing queue manager.

Take copies of all the existing data of a queue manager and log file directories, including all subdirectories.

Overwrite the data and log file directories of the backup queue manager, including all subdirectories, with the copies taken from the existing queue manager.

Execute the following control command on the backup queue manager: **strmqm -r** *BackupQMName* This flags the queue manager as a backup queue manager within WebSphere MQ, and replays all the copied log extents to bring the backup queue manager in step with the existing queue manager. Note, the "-r" flag does not start the backup queue manager, it just causes the logs to be replayed so the current queue contents reflect the logs.

Updating a backup queue manager

Backup queue manager must be kept up to date

1. Advance the log on the active queue manager:
   Run script command: `RESET QMGR TYPE(ADVANCELOG)`
2. Obtain current log extent:
   Run script command: `DISPLAY QMSTATUS CURRLOG`
3. Copy logs to backup queue manager
4. Update the backup queue manager:

```
strmqm -r QmgrName
```


Figure 15-12. Updating a backup queue manager WM203 / VM2032.0

Notes:

To ensure that a backup queue manager remains an effective method for disaster recovery it must be updated regularly. Regular updating lessens the discrepancy between the backup queue manager log, and the current queue manager log. There is no need to stop the queue manager to be backed up.

To update a backup queue manager, do the following actions:

1. Issue the following Script (MQSC) command on the queue manager to be backed up: RESET QMGR TYPE(ADVANCELOG) This stops any writing to the current log, and then advances the queue manager logging to the next log extent. This ensures you backup all information logged up to the current time.
2. Obtain the (new) current active log extent number by issuing the following Script (MQSC) command on the queue manager to be backed up: DIS QMSTATUS CURRLOG
3. Copy the updated log extent files from the current queue manager log directory to the backup queue manager log directory - copy all the log extents since the last update,

and up to (but not including) the current extent noted in step 2. Copy only log extent files, the ones beginning with "S".

4. Issue the following control command on the backup queue manager: strmqm -r BackupQMName This replays all the copied log extents and brings the backup queue manager into step with the queue manager. When the replay finishes you receive a message that identifies all the log extents required for restart recovery, and all the log extents required for media recovery. Warning: If you copy a non-contiguous set of logs to the backup queue manager log directory, only the logs up to the point where the first missing log is found are replayed.

Activating the backup queue manager

To use the backup queue manager:

1. Activate the backup queue manager:

```
strmqm -a QMgrName
```

2. Start the backup queue manager:

```
strmqm QmgrName
```

3. Restart all channels

4. Verify results

Figure 15-13. Activating the backup queue manager WM203 / VM2032.0

Notes:

To substitute an unrecoverable queue manager with its backup queue manager, do the following actions:

1. Execute the following control command to activate the backup queue manager:
   `strmqm -a BackupQMName`
   The backup queue manager is activated. Now active, the backup queue manager can no longer be updated.
2. Execute the following control command to start the backup queue manager:
   `strmqm BackupQMName`
   This command is a restart recovery. It uses the log from the backup queue manager. During the last update to the backup queue manager replay has occurred, therefore only the active transactions from the last recorded checkpoint are rolled back. When an unrecoverable queue manager is substituted for a backup queue manager some of the queue manager data from the unrecoverable queue manager can be lost. The amount of lost data is dependent on how recently the backup queue manager was last updated. The more recently the last update, the less queue manager data loss.

3. Restart all channels.

Check the resulting directory structure to ensure that all the required directories are present.

WebSphere MQ HA strategies topic summary

Having completed this topic, you should be able to:
- Describe WebSphere MQ HA strategies

Figure 15-14. WebSphere MQ HA strategies topic summary WM203 / VM2032.0

Notes:

Platform technologies topics objectives

After completing this topic, you should be able to:
- Describe the platform HA strategies

Figure 15-15. Platform technologies topics objectives

Notes:

Single points of failure

- With no redundancy or fault tolerance, a failure of any component can lead to loss of availability
- Every component is critical:
  - Power supply, central processing unit, memory
  - Disk controllers, disks, network adapters, cables, and so on
- Various techniques exist to help tolerate failures:
  - Dual power supplies and uninterruptible power supplies
  - RAID for disk resiliency and recovery
  - Fault tolerant architectures for central processing unit and memory
- High availability clustering
  - Used with some of the above
  - Cost effective means of avoiding single points of failure

Notes:

Standby configuration

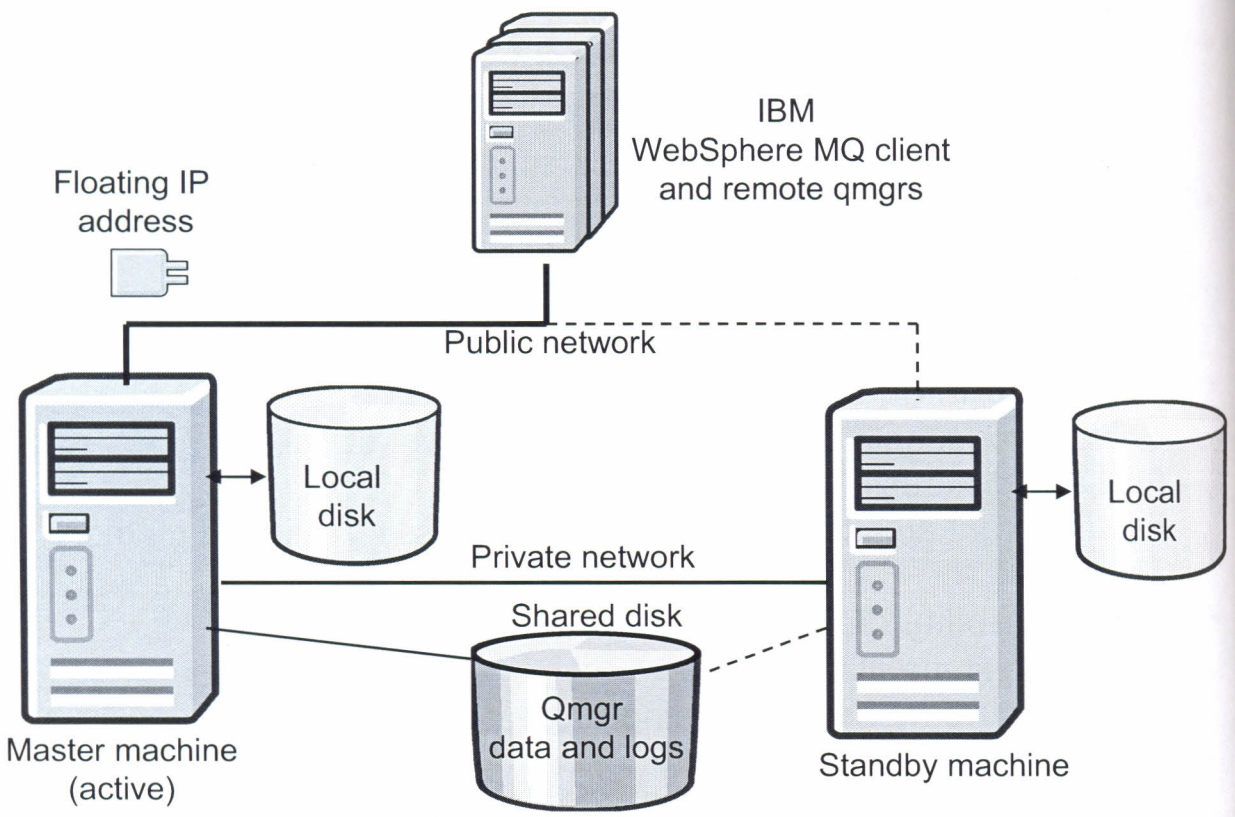

Figure 15-17. Standby configuration

Notes:

A **standby** configuration is the most basic cluster configuration in which one node performs work while the other node acts only as standby. The standby node does not perform work and is referred to as idle; this configuration is sometimes called "cold standby". Such a configuration requires a high degree of hardware redundancy. To economize on hardware, it is possible to extend this configuration to have multiple worker nodes with a single standby node, the idea being that the standby node can take over the work of any other worker node. This is still referred to as a standby configuration and sometimes as an "N+1" configuration.

Standby fail-over

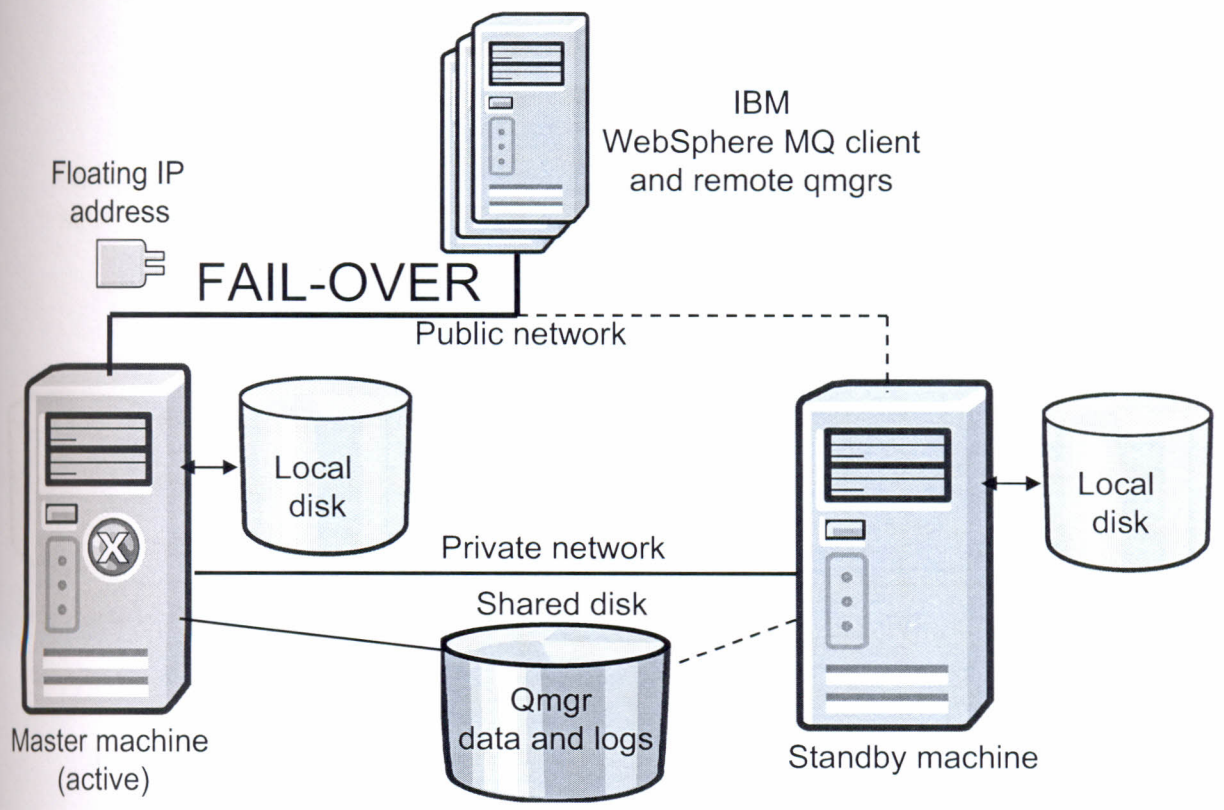

Figure 15-18. Standby fail-over

Notes:

The master server has failed. The standby node initiates a failover to the standby computer.

Standby fail-over complete

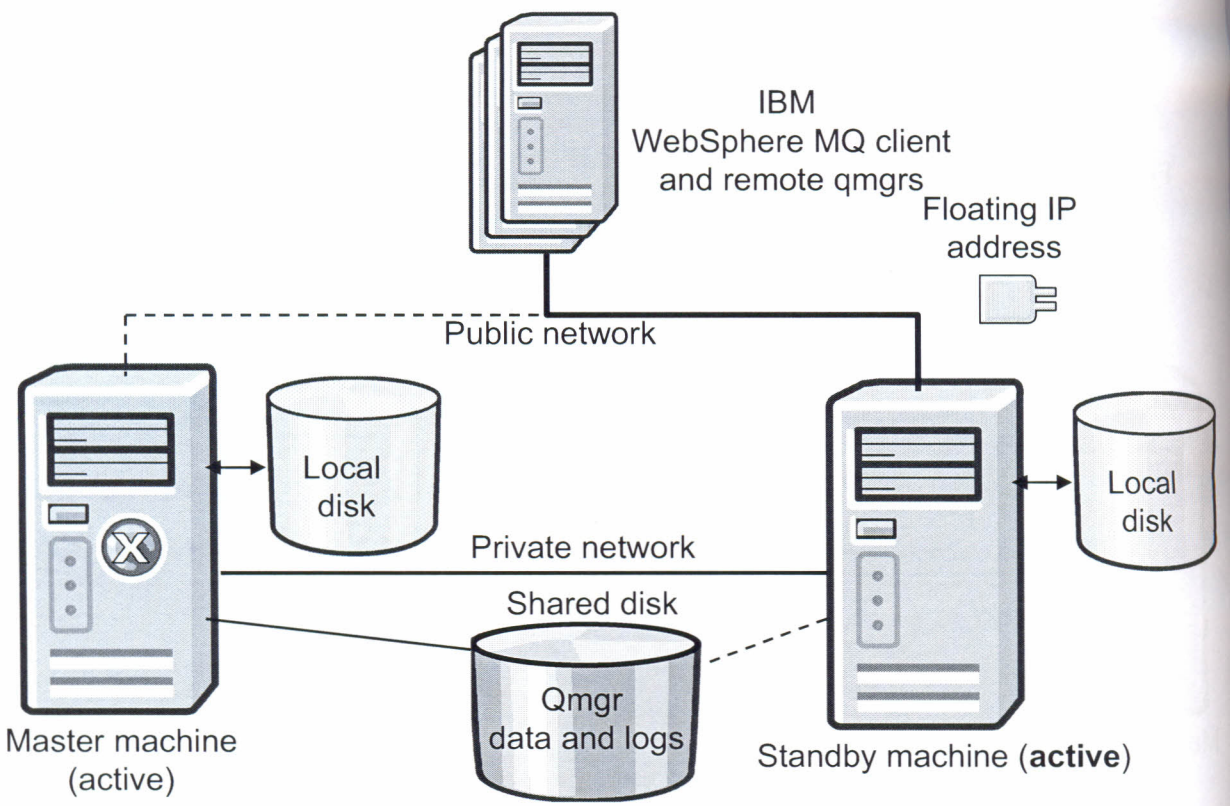

Figure 15-19. Standby fail-over complete

WM203 / VM2032.0

Notes:

The master server has failed. The standby node initiates a failover to the standby computer.

Active-active configuration

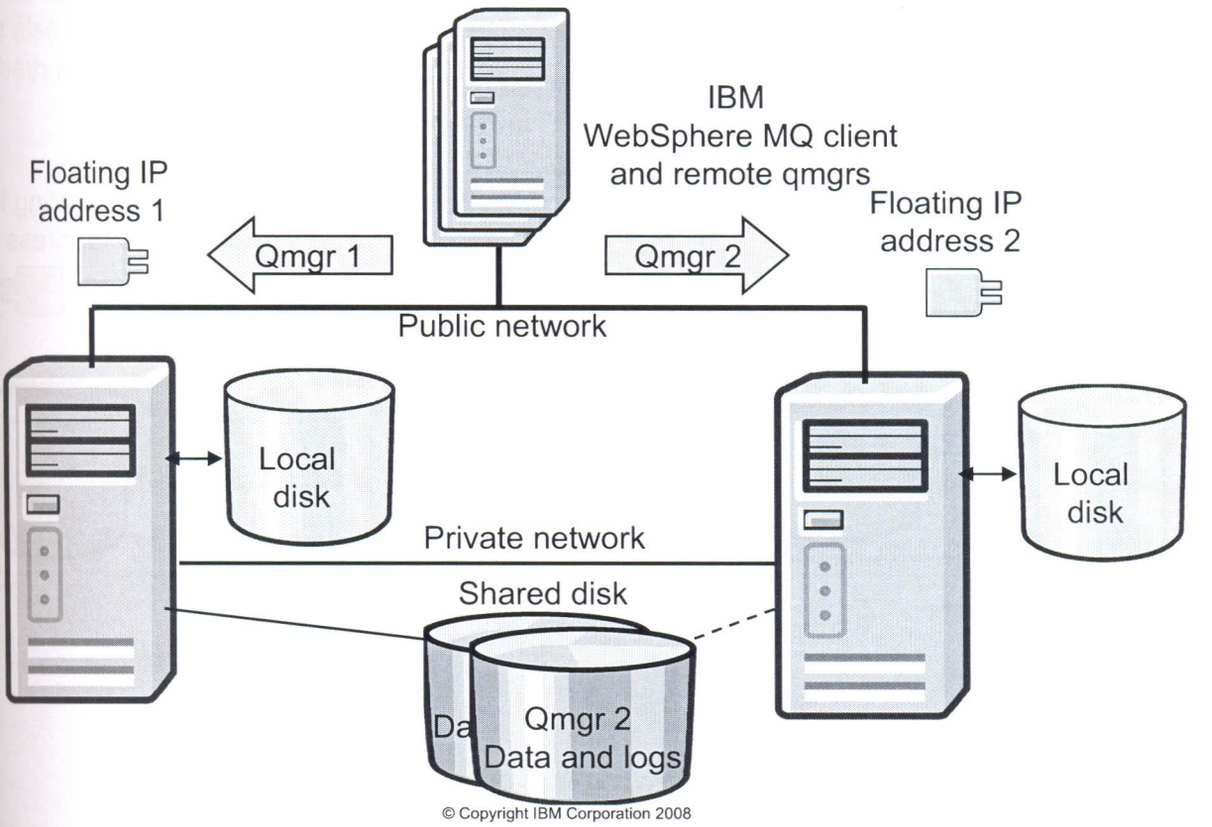

Figure 15-20. Active-active configuration

WM203 / VM2032.0

Notes:

A **takeover** configuration is a more advanced configuration in which all nodes perform some work and critical work can be taken over in the event of a node failure. A "one sided takeover" configuration is one in which a standby node performs some additional, non-critical, and non-movable work. This is rather like a standby configuration but with (non-critical) work being performed by the standby node. A "mutual takeover" configuration is one in which all nodes are performing highly available (movable) work. This type of cluster configuration is also sometimes referred to as "**active/active**" to indicate that all nodes are actively processing critical workload.

With the extended standby configuration of either of the takeover configurations it is important to consider the peak load which may be placed on any node which can take over the work of other nodes. Such a node must possess sufficient capacity to maintain an acceptable level of performance.

Active-active takeover

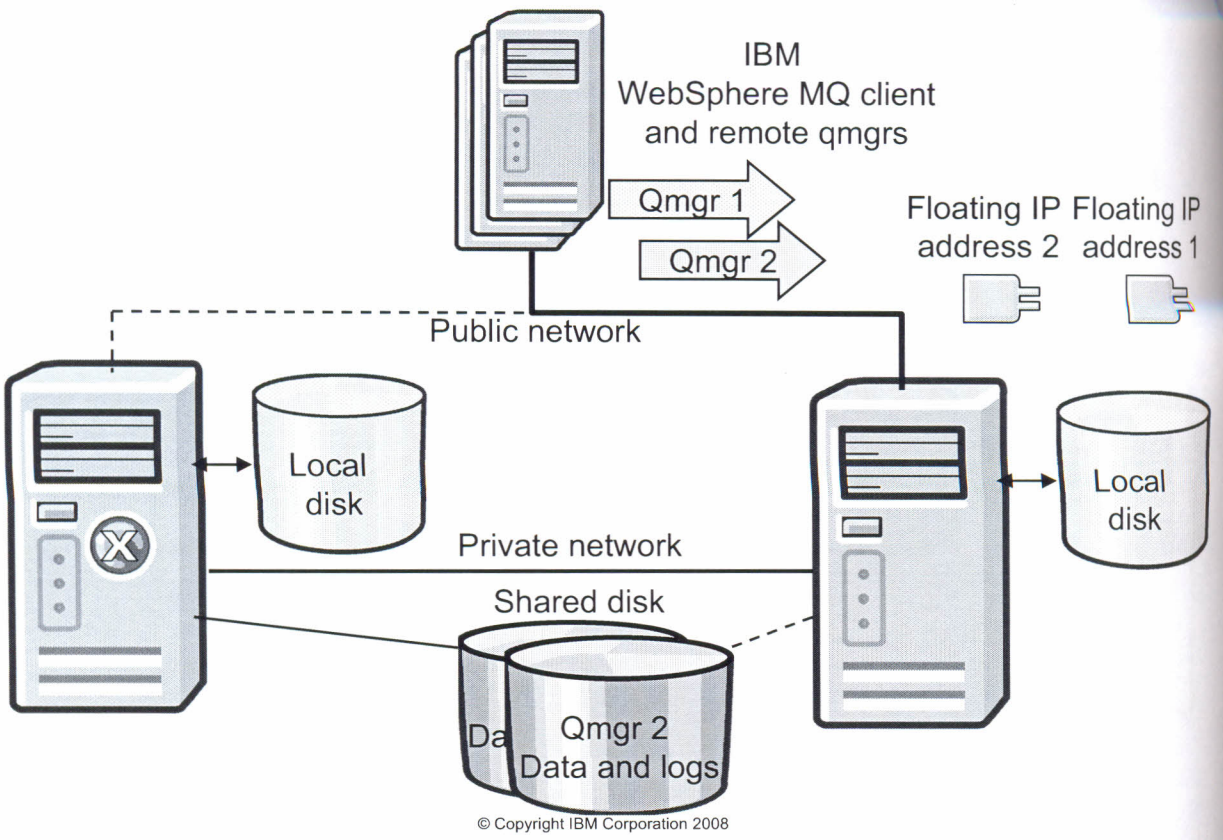

Figure 15-21. Active-active takeover

WM203 / VM2032.0

Notes:

This configuration is also sometimes called a mutual takeover system

In normal operation, both machines are running independent queue managers. If one of the systems fails, this configuration can switch the failed queue manager to the working computer. It still appears to applications outside the HA cluster that there are two queue managers. The throughput of each queue manager may degrade, depending on how heavily they are loaded and how much spare capacity existed, but at least the work is still getting done.

With this setup, there should be a fail-back capability so that the queue manager can be sent back to its original node when the failure has been corrected. Whether the fail-back is automatic or not is a choice but it is recommended that it is done manually so applications which have already connected to the running queue manager do not have their connections broken unnecessarily. It would be advisable to monitor the workload and only initiating fail-back when there is not too much work that is going to be disrupted by the fail-back.

A number of variations on the themes of cold and hot standby are also possible, for example having 3 nodes to host 2 queue managers (an "N+I" configuration). The availability of these options depends on the facilities available in your cluster software.

In this configuration, the IP address associated with each queue manager being switched. You also need to keep a port number reserved for each queue manager the same number on both machines in the cluster) and have listeners defined appropriately.

Fail-over considerations

- Data access
  - Shared disks – not concurrent but "switchable"
  - Alternatively mirrored data (must be true, synchronized mirror)
- Network connectivity
  - IP address take-over
- Node equivalence
  - Common hardware
  - Common software (levels, paths)
- Performance and capacity
  - Sufficient to handle maximum workload
- Take-over time
  - Several factors, including time for transaction recovery

Figure 15-22. Fail-over considerations WM203 / VM2032.0

Notes:

WebSphere MQ in HA clusters

- *Separate* disks for data and log files
  - Not essential, but recommended for performance
- Channel state is hardened
  - *Sender* channels will be automatically restarted (if triggered)
  - Virtual IP address must be used on all incoming sender channels
  - Requester channels are not automatically restarted
- Some services may require manual restart:
  - Trigger monitors
  - Command server
  - Applications
- Effectively a normal queue manager restart
  - Non-persistent messages are lost
- Long running transactions will slow down restart

Notes:

Microsoft Cluster Service (MSCS)

- Support integrated with WebSphere MQ
- Capabilities:
  - Active/Active configuration
  - Unit of failure = queue manager (with one or more queue managers per MSCS group)
  - Automatic queue manager registration on standby node
  - Security and registration synchronization
  - Supports WebSphere MQ custom services
- Utility programs:
  - Registration/deregistration (haregtyp)
  - Move a queue manager to MSCS storage (hamvmqm)
  - Remove queue manager from node (hadltmqm)
  - Check and save setup details (amqmsysn)

Figure 15-24. Microsoft Cluster Service (MSCS) WM203 / VM2032.0

Notes:

Consult the WebSphere MQ V7 Information Center:
`http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/topic/com.ibm.mq.amqzag.doc/fa14110_.htm` for full details on configuring MSCS for WebSphere MQ.

Putting a queue manager under MSCS control (1 of 2)

Preparation:

1. Verify MSCS is correctly installed
2. Use **haregtyp** to register WebSphere MQ on all nodes
3. Create a queue manager on one node
4. Move queue manager to MSCS storage
5. Stop queue manager if it is running

Figure 15-25. Putting a queue manager under MSCS control (1 of 2) WM203 / VM2032.0

Notes:

Before you put a queue manager under MSCS control:

1. Ensure that WebSphere MQ and its MSCS Support is installed on both machines in the cluster and that the software on each computer is identical.
2. Use the **haregtyp** utility program to register WebSphere MQ as an MSCS resource type on all of the cluster nodes.
3. If you have not yet created the queue manager.
4. If you have created the queue manager, or it exists, move the queue manager to MSCS storage using the hamvmqm utility as described in detail in the WebSphere MQ V7 Information Center.
5. Stop the queue manager, if it is running, using either a command prompt or the WebSphere MQ Explorer.
6. Test MSCS operation of the shared drives before going on.

Putting a queue manager under MSCS control (2 of 2)

- As the MSCS administrator create an MSCS group
- Add resource instances
  - For each SCSI drive used
  - One or more virtual IP addresses
  - One or more queue managers
- Define failover policy
- Test

Figure 15-26. Putting a queue manager under MSCS control (2 of 2) WM203 / VM2032.0

Notes:

The figure summarizes the steps needed to make MSCS handle WebSphere MQ failover. Consult the WebSphere MQ Information Center for a detailed description of the steps required.

WebSphere MQ support for UNIX HA

- SupportPac MC91
  - Supersedes *MC63, MC6A and MC6B*
  - Supports HACMP (AIX), Veritas Cluster Service (VCS), MC/ServiceGuard and others
- Scripts provided include:
  - Create queue manager: hacrtmqm
  - Delete queue manager: hadltmqm
  - Start queue manager: hamqm_start
  - Stop queue manager: hamqm_stop
- Two disk partitions per queue manager
  - Each on shared disk, independently defined and failed-over
  - Ignore advice to create `/var/mqm` and `/var/mqm/log` file systems
- Network configuration

Figure 15-27. WebSphere MQ support for UNIX HA WM203 / VM2032.0

Notes:

This SupportPac is a Category-2 SupportPac and as such, no support is provided for the use of these scripts. Although only some HA providers are mentioned, you might be able to customize these scripts for other HA providers.

Putting a queue manager under HA on UNIX

1. Configure a HA cluster
2. Configure the shared disks
3. Create the queue manager
4. Configure the movable resources
5. Configure the application server or agent
6. Configure an application monitor

Figure 15-28. Putting a queue manager under HA on UNIX WM203 / VM2032.0

Notes:

1. Configure a HA cluster
   An initial configuration is straightforward and should present no difficulties for a trained implementer using the product documentation
2. Configure the shared disks
   This step creates the volume group (or disk group) and file systems needed for the queue manager. The suggested layout is based on the advice earlier that each queue manager should be put into a separate resource group. You should perform this step and the subsequent steps for each queue manager that you want to make highly available.
3. Create the queue manager
   When you create the queue manager, it is advised that you use the hacrtmqm script. It is possible to create the queue manager manually, but using hacrtmqm saves much effort. For example, hacrtmqm moves and relinks some subdirectories and for HACMP creates an HACMP/ES Application Monitor for the queue manager. The move and

relink of these subdirectories is to ensure smooth coexistence of queue managers which may run on the same node.

4. Configure the movable resources
   The queue manager has been created and the standby/takeover nodes have been updated. You now need to define a resource or service group which contains the queue manager and all its associated resources. These include IP addresses, shared disks, and so on.
5. Configure the Application Server or Agent
   The queue manager is represented within the resource group by an application server or agent. The SupportPac includes example server start and stop methods which allow the HA products to start and end a queue manager, in response to cluster commands or cluster events.
6. Configure an Application Monitor
   This SupportPac includes a monitor for WebSphere MQ, which allows the HA product to monitor the health of the queue manager and initiate recovery actions that you configure, including the ability to restart the queue manager locally or move it to an alternate system.

Comparison of technologies

After a failure:

Technology	Access to existing messages	Access for new messages
None	No	No
WebSphere MQ clustering	No	Yes
Backup Qmgr	Yes/No	Yes
HA clustering	Yes	Yes

Figure 15-29. Comparison of technologies WM203 / VM2032.0

Notes:

Shown here are the advantages and disadvantages of the various approaches discussed. First, in the do-nothing approach, a queue manager failure is just that. All access to the queue manager is lost until service is restored.

WebSphere MQ clustering provides an intermediate solution. Messages already on a queue on the failed queue manager are unavailable until that queue manager is restored. However new messages are routed to one of the working queue managers and processed.

Backup queue managers provide a way to recover a failed queue manager with its data in tact, however, the service restoration time is very much dependent on how long it takes to replay logs on the backup queue manager and then, depending on how recent the logs are, some messages may be missing.

Finally, using a HA solution, with it is use of shared, highly available disk, means that all messages will be available after a failure, except in the case of non-persistent messages.

Checkpoint questions

1. Why would it not normally be necessary to define more than two WebSphere MQ cluster full repositories?

2. How are logs replayed on a backup queue manager?
   a. `runmqlog -m QmgrName`
   b. `alter qmgr replay(log)`
   c. `strmqm -r QmgrName`
   d. `dmpmqlog -r QmgrName`

3. True or false: All platform HA solutions employ a floating IP address to effect transparent fail-over.

4. True or false: Fail-over is effectively a queue manager restart so all non-persistent messages are lost.

Notes:

Platform technologies topic summary

Having completed this topic, you should be able to:
- Describe the platform HA strategies

Figure 15-31. Platform technologies topic summary WM203 / VM20320

Notes:

Unit summary

Having completed this unit, you should be able to:

- Describe the steps needed to switch a running queue manager between two high availability (HA) systems
- Plan for using HA systems with WebSphere MQ

Figure 15-32. Unit summary

Notes: