WebSphere MQ High Availability

Smarter Process

WebSphere MQ Industry Practices

Credits

- Talk to your IBM representative
- Talk to your collegues
- Visit The Capitalware site

http://www.capitalware.biz/

WebSphere MQ High Availability

Introduction

- Availability:
  - What does it mean to me.
  - What does it mean to my business
  - What does it mean to my application
- Technology:
  - Cannot solve all your problems
  - How close do you come to 99999
- Planning
  - Planning, the four letter word
  - Approach
- Testing
  - Taken for granted
  - All or nothing
  - Continuous

High Availability – Industry view

In information technology, high availability refers to a system or component that is continuously operational for a desirably long length of time. Availability can be measured relative to "100% operational" or "never failing." A widely-held but difficult-to-achieve standard of availability for a system or product is known as "five 9s" (99.999 percent) availability.

WebSphere MQ High Availability

Weakest Link

- With no redundancy or fault tolerance, a failure of any key component can lead to a loss of availability
- Every component is critical. The system relies on the:
  - Power supply, system unit, CPU, memory
  - Disk controller, disks, network adapter, network cable
  - ...and so on
- Various techniques have been developed to tolerate failures:
  - UPS or dual supplies for power loss
  - RAID for disk failure
  - Fault-tolerant architectures for CPU/memory failure
  - ...etc
- Elimination of SPOFs is important to achieve HA

WebSphere MQ High Availability

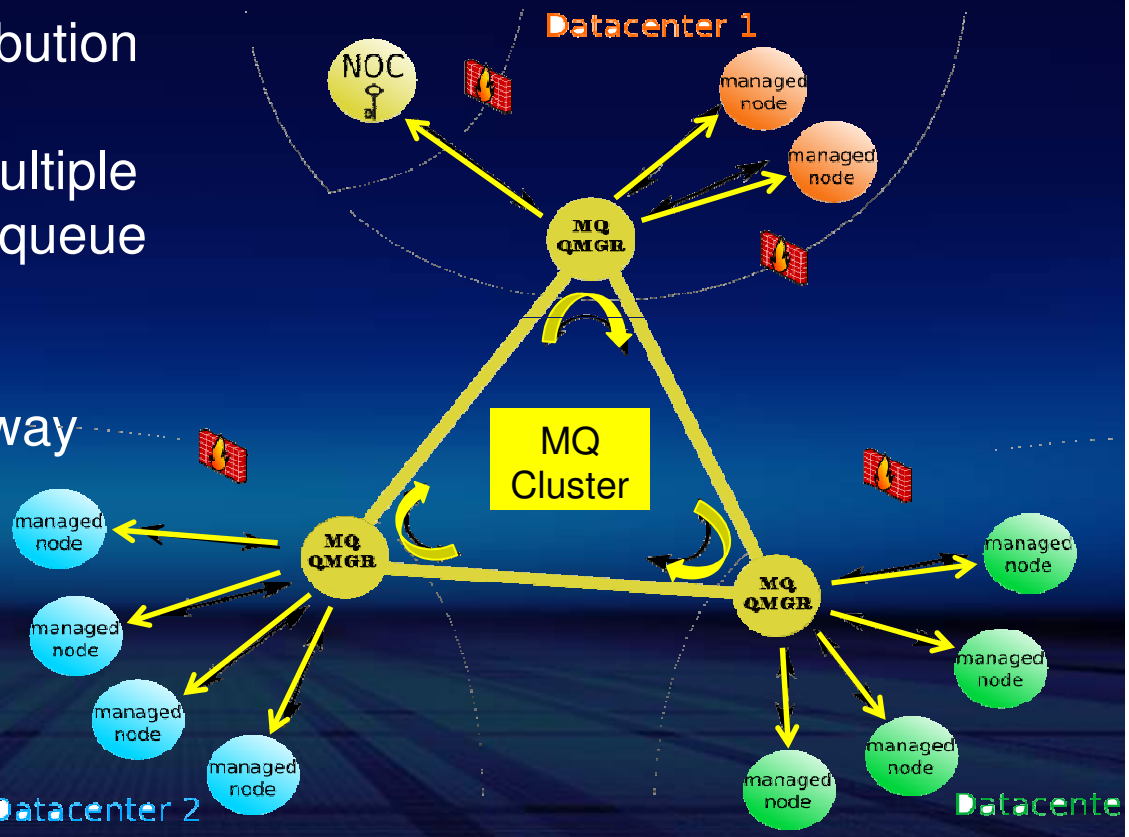
WebSphere MQ HA Technologies

- Queue Manager Clusters
- Queue-sharing groups
- Support for networked storage
- Multi-instance Queue Managers
- Multi-instance Message Broker
- HA Clusters
- Client reconnection

WebSphere MQ High Availability

Queue Manager Clusters

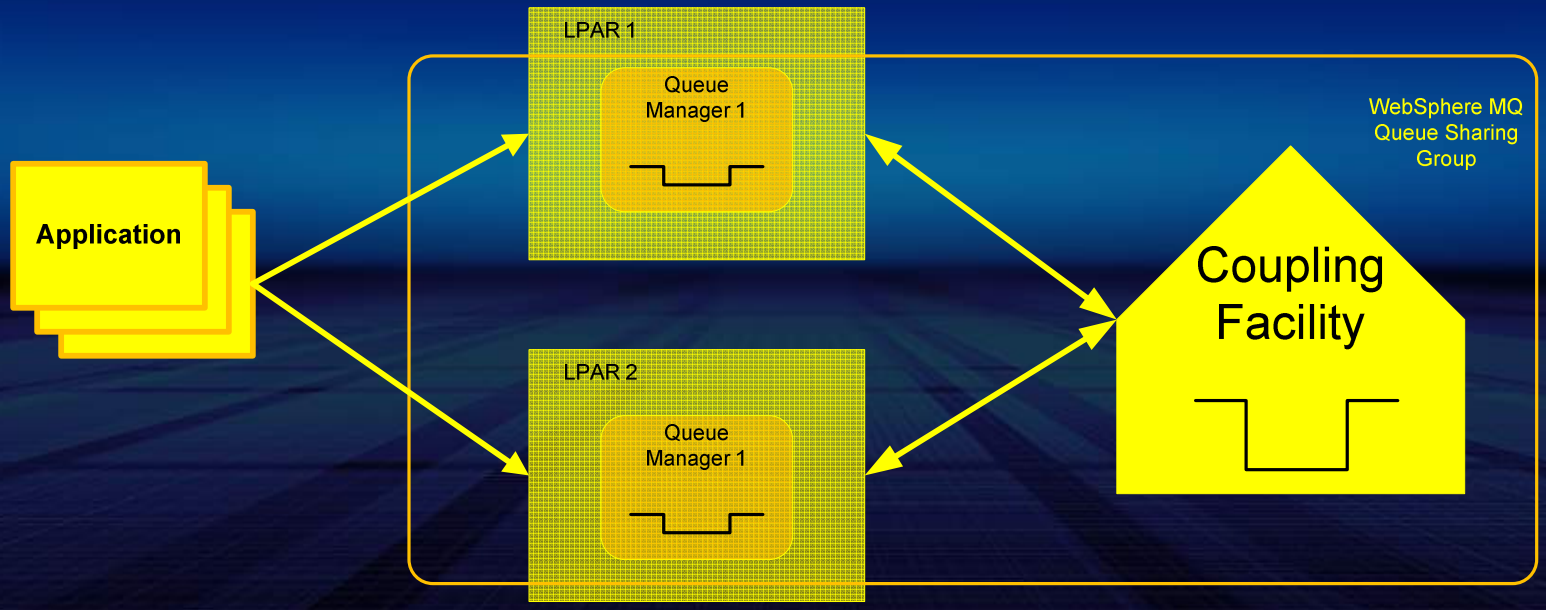
- Cluster Queue manager provide alternate availability for distribution of messages
- Sharing cluster queues on multiple queue managers prevents a queue from being a SPOF
- Cluster workload algorithm automatically routes traffic away from failed queue managers
- New feature even allows the
- QMGR to react when there
- Is no application reading
- messages



WebSphere MQ High Availability

Queue-Sharing Groups

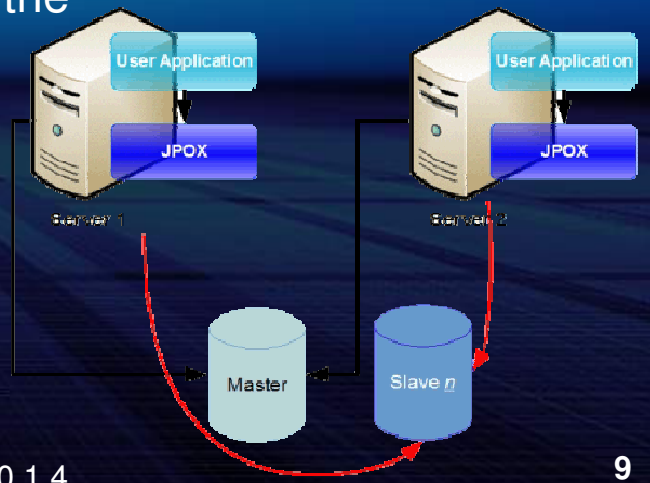
- On z/OS, queue managers can be members of a queue-sharing group
- Shared queues are held in a coupling facility
  - All queue managers in the QSG can access the messages
- Benefits:
  - Messages remain available even if a queue manager fails
  - Pull workload balancing
  - Apps can connect to the group



WebSphere MQ High Availability

Introduction to Failover and MQ

- Failover is the automatic switching of availability of a service
  - For MQ, the “service” is a queue manager
- Traditionally the preserve of an HA cluster, such as HACMP
- Requires:
  - Data accessible on all servers
  - Equivalent or at least compatible servers
    - Common software levels and environment
  - Sufficient capacity to handle workload after failure
    - Workload may be rebalanced after failover requiring spare capacity
  - Start-up processing of queue manager following the failure
- MQ offers two ways of configuring for failover:
  - Multi-instance queue managers
  - HA clusters



WebSphere MQ High Availability

Failover considerations

- Failover times are made up of three parts:
  - Failure Notification
    - Heartbeat missed
    - Bad result from status query
  - Environment switch
    - Shutting down non-effected systems
    - Switching IP addresses, disks, etc
  - Time taken to activate the service
    - Queue Manager restart
    - Application Restart
- Failover involves a queue manager restart
  - Nonpersistent messages, nondurable subscriptions discarded
- For fastest times, ensure that queue manager restart is fast
  - No long running transactions, for example
  - Less Persistent messages
  - Fast Disk, non-contention

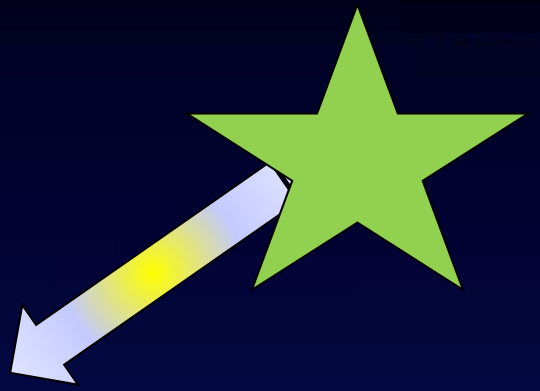
WebSphere MQ High Availability

Multi-instance Queue Managers

- Out of the Box failover
  - No Additional software required
- Single instance queue manager on different machines
  - One is the “active” instance, other is the “standby” instance
  - Active instance “owns” the queue manager’s files
    - Accepts connections from applications
  - Standby instance monitors the active instance
    - Applications cannot connect to the standby instance
    - If active instance fails, standby restarts queue manager and becomes active
- Instances are the SAME queue manager
  - only one set of data files
  - Queue manager data is held in networked storage

WebSphere MQ High Availability

Setting up Multi-instance Queue Manager



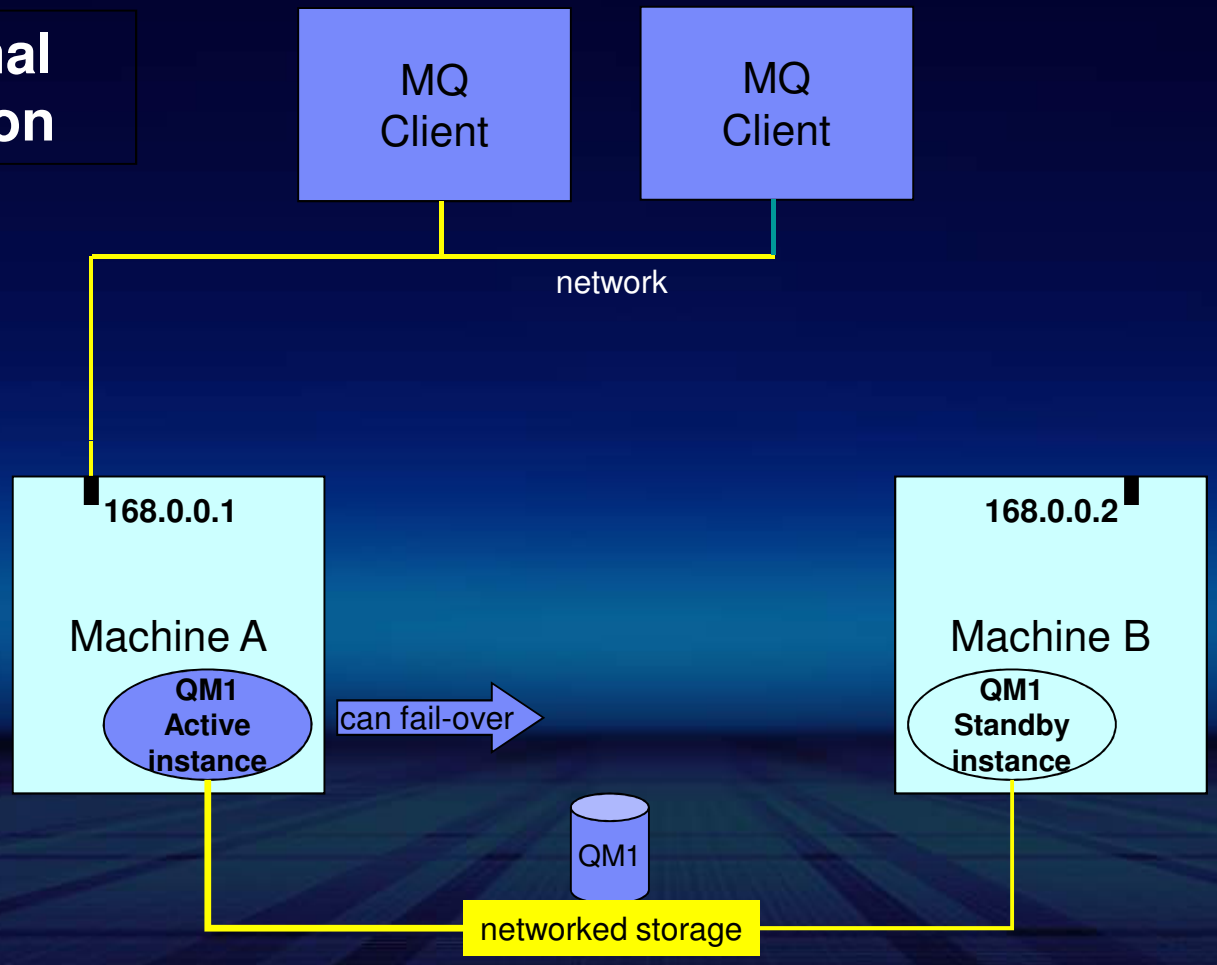
- Set up shared file systems for QM data and logs
- Create the queue manager on machine1
  - crtmqm **–md /shared/qmdata –ld /shared/qmlog QM1**
- Define the queue manager on machine2 (or edit mqs.ini)
  - addmqinf –v Name=QM1 –v Directory=QM1 –v Prefix=/var/mqm -v DataPath=/shared/qmdata/QM1
- Start an instance on machine1 – it becomes active
  - strmqm –x QM1
- Start another instance on machine2 – it becomes standby
  - strmqm –x QM1
- Conversion Command

That's it. If the queue manager instance on machine1 fails, the standby instance on Machine2 takes over and becomes active

WebSphere MQ High Availability

Multi-instance Queue Managers - Execution

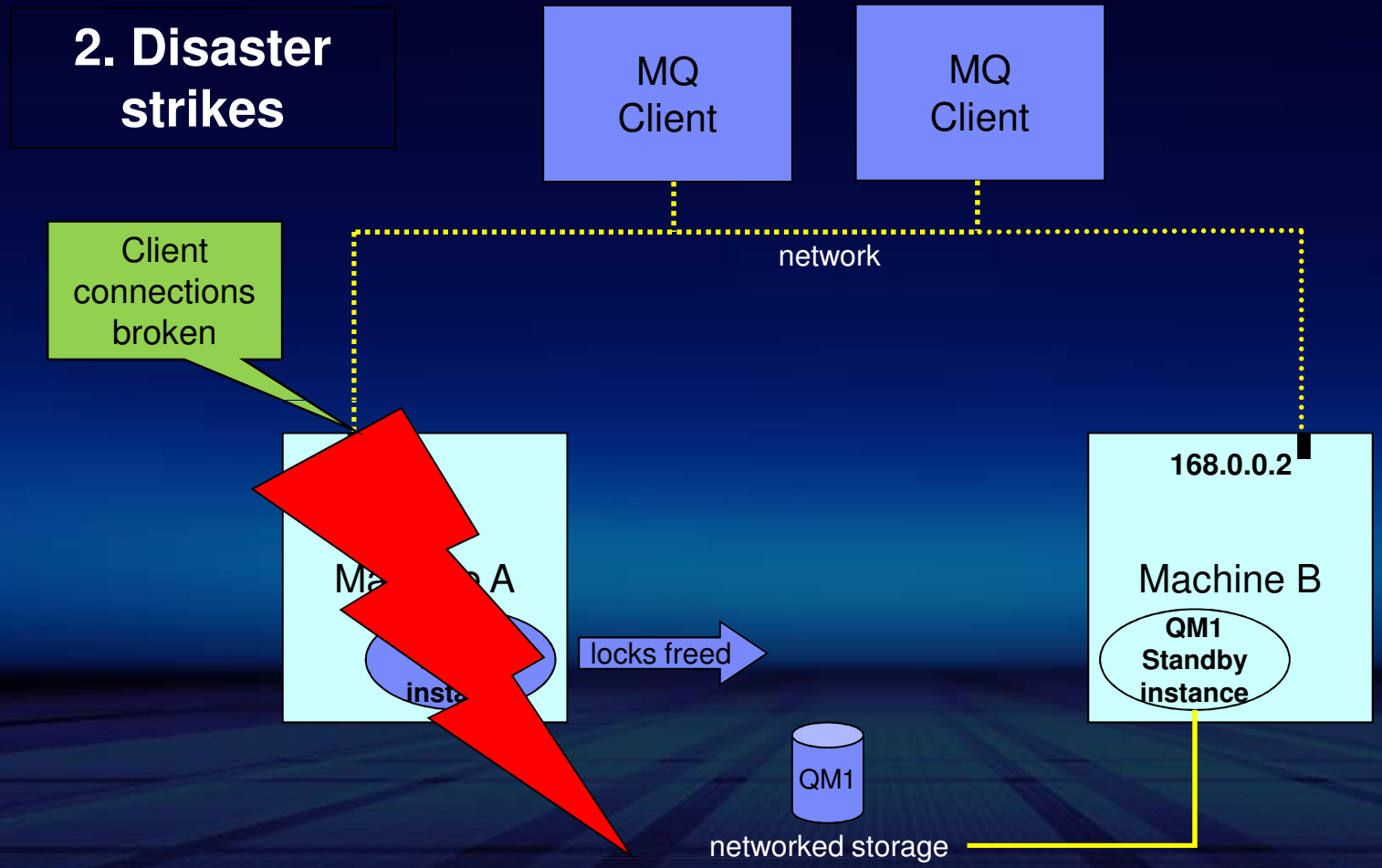
1. Normal execution



Owns the queue manager data

Multi-instance Queue Managers

Multi-instance Queue Managers – Disaster

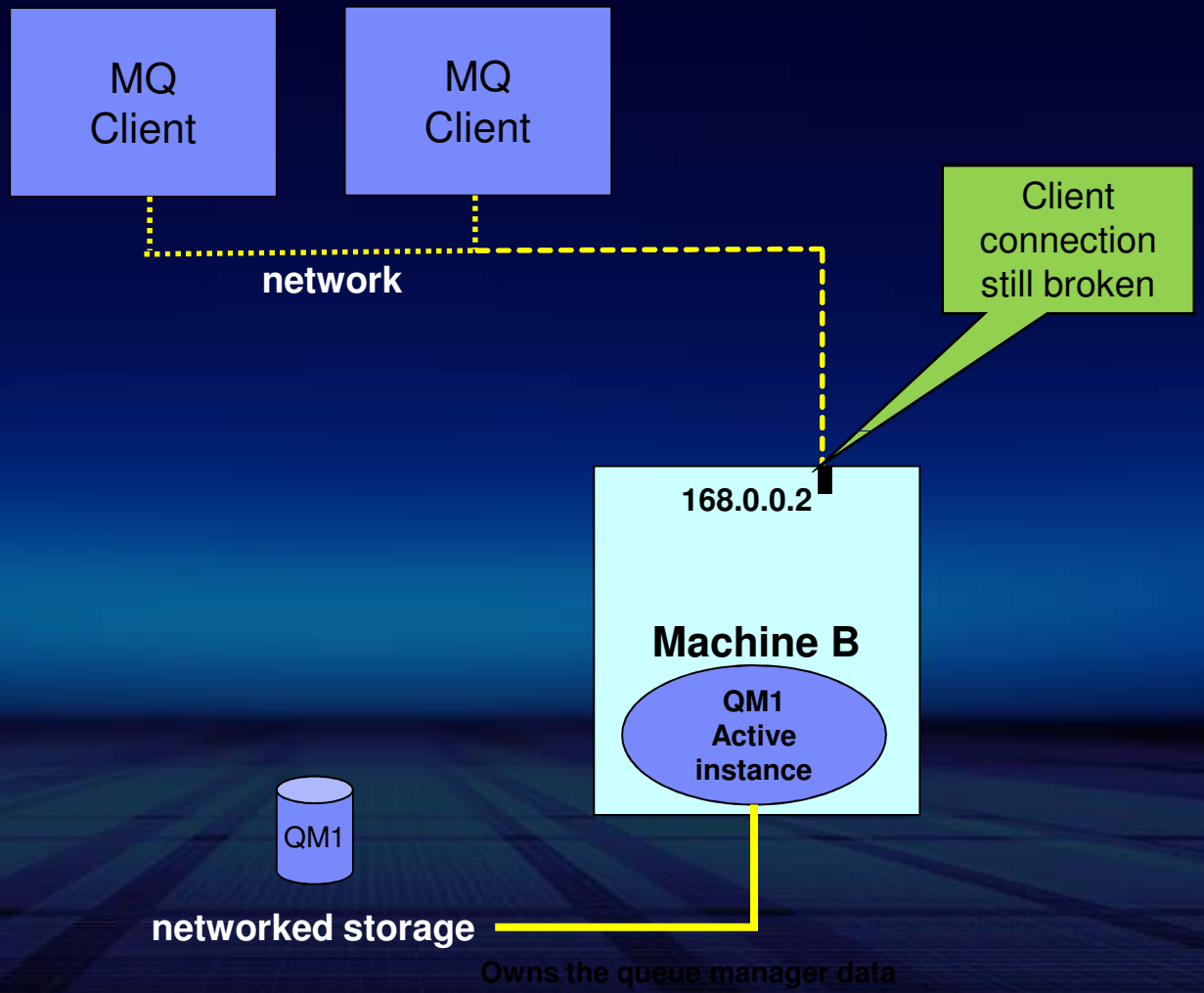


Multi-instance Queue Managers

Multi-instance Queue Managers – Failover

3. FAILOVER

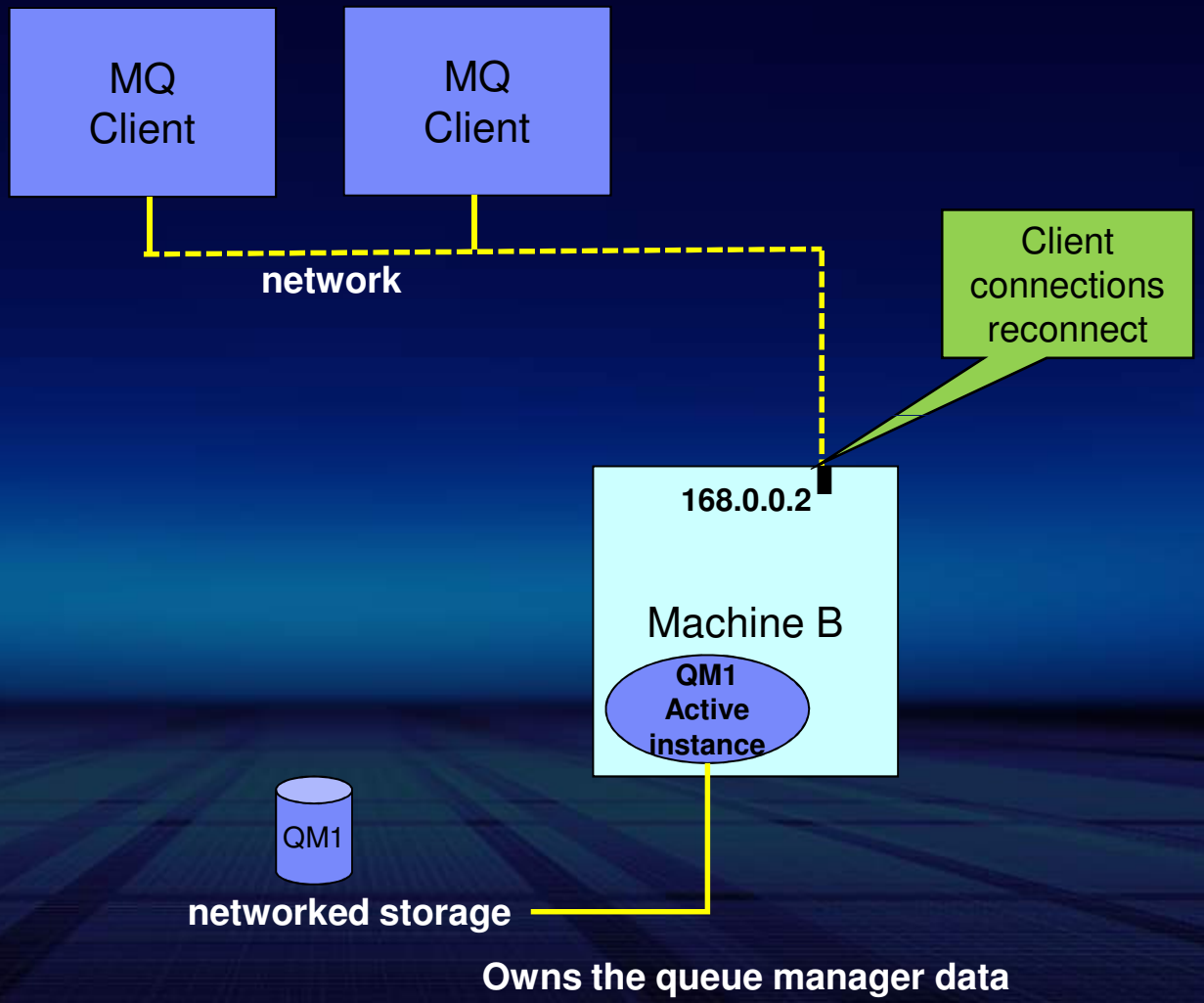
Standby becomes active



WebSphere MQ High Availability

Multi-instance Queue Managers – Failover Complete

4. Recovery complete



WebSphere MQ High Availability

Multi-instance Queue Managers - Restrictions

- MQ is NOT an HA cluster coordinator
  - If other resources need to be coordinated, you need an HA cluster
  - WebSphere Message Broker integrates with multi-instance QM
- Queue manager services can be automatically started, but with limited control
- Client connections
  - Automatic reconnection via Client
  - Pre v7.0.1 reroute via IP Sprayer or CCDT
- Stand By Queue Manager
  - Post Failover no standby Queue Manager
  - Standby Queue Manager must be restarted

System administrator is responsible for restarting another standby instance when failover has occurred

WebSphere MQ High Availability

Dealing with multiple IP addresses

- The IP address of the queue manager changes when it moves
  - So channel configuration needs knowledge of this
- Connection name syntax extended to a comma-separated list
  - CONNAME('168.0.0.1,168.0.0.2')
  - Needs 7.0.1+ qmgr or client
- Unless you use external IPAT or an intelligent router or MR01
- WAS8 admin panels understand this syntax.
- For earlier levels of WAS
  - Connection Factories:
    - Set a custom property called XMSC_WMQ_CONNECTION_NAME_LIST to the list of host/port names that you wish to connect to
    - Make sure that the existing host and port values defined on the connection factory match the first entry in this property
  - Activation Specs:
    - Set a custom property called connectionNameList on the activation spec with the same format

WebSphere MQ High Availability

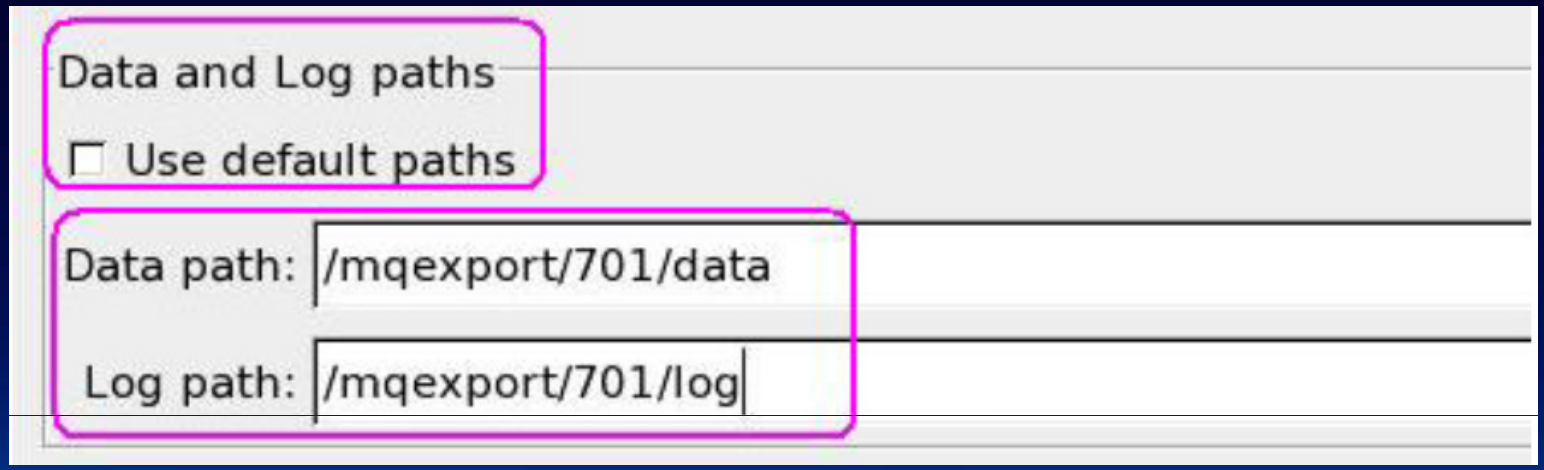
Administering Multi-instance QMgrs

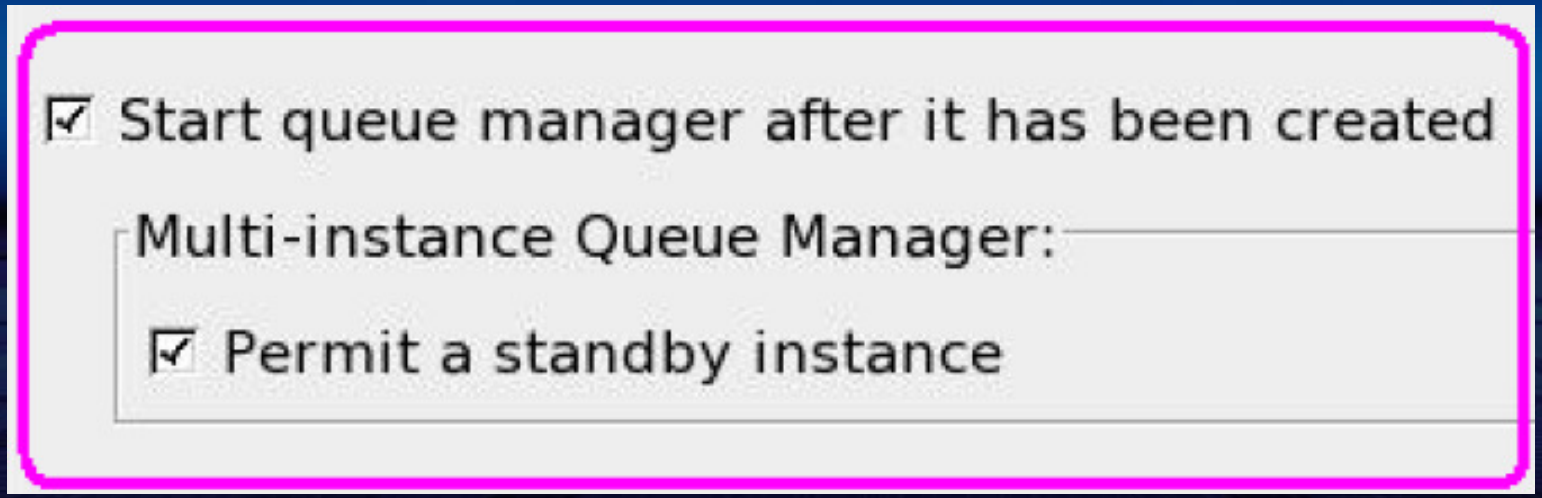
- All queue manager administration must be performed on active instance
- dspmq enhanced to display instance information
  - dspmq issued on “staravia”
  - On “staravia”, there’s a standby instance
  - The active instance is on “starly”

```
$ hostname
staravia
$ dspmq -x
QMNAME(MIQM)        STATUS(Running as standby)
    INSTANCE(starly)      MODE(Active)
    INSTANCE(staravia)    MODE(Standby)
```

WebSphere MQ High Availability

Multi-instance QMGR MQ Explorer – Create QMGR

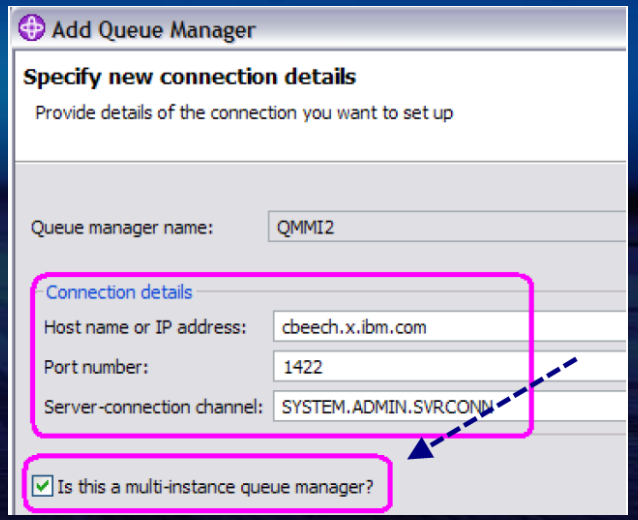




WebSphere MQ High Availability

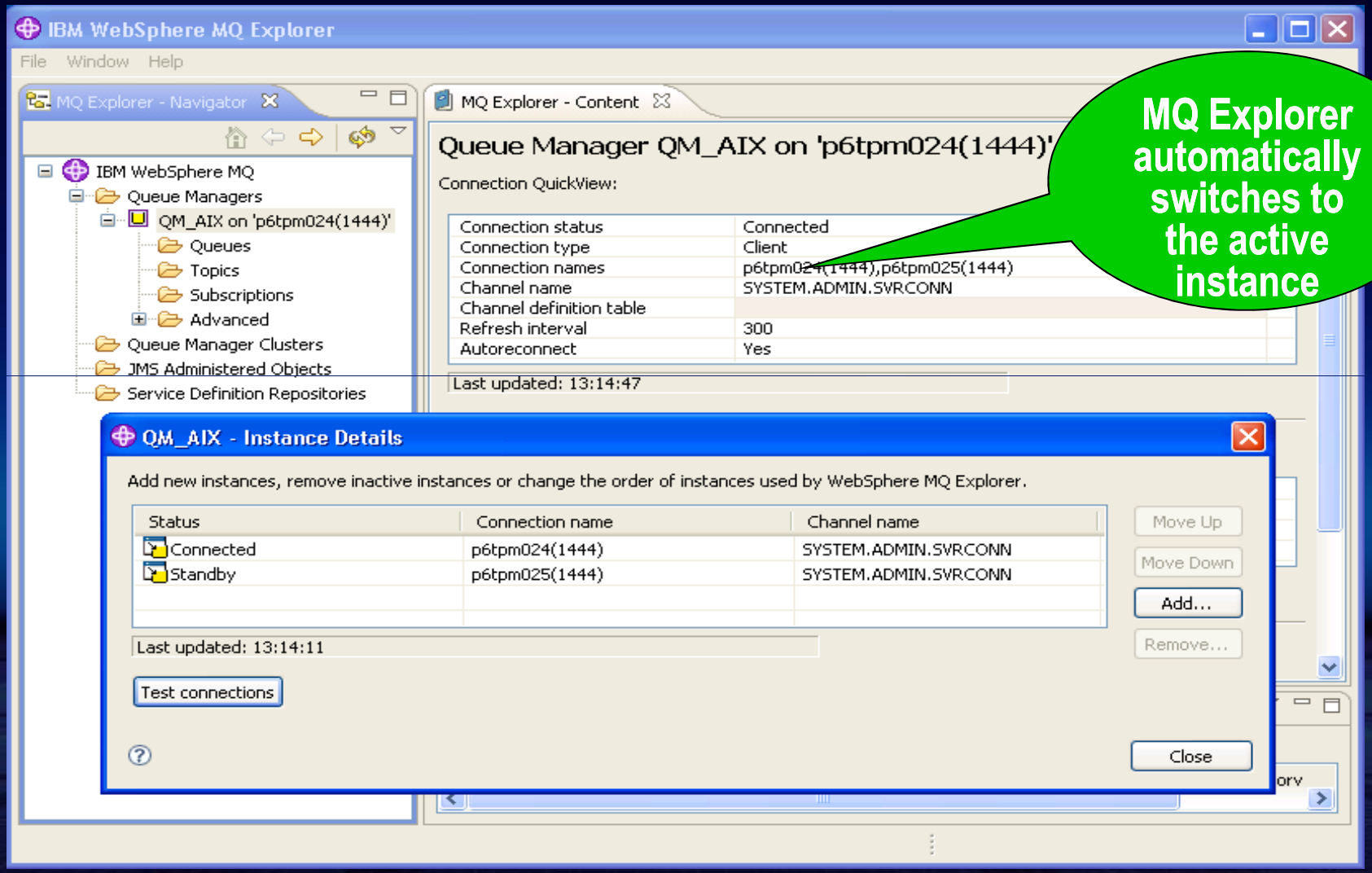
MQSeries Ten Commandments

```
$ dspmqinf -o command QMMI2

The output is:

addmqinf -s QueueManager -v Name=QMMI2 -v Directory=QMMI2 -v
Prefix=/var/mqm -v DataPath=/mqexport/701/data/QMMI2
```



WebSphere MQ High Availability

Multi-instance QMGR in MQ Explorer



WebSphere MQ High Availability

HA clusters

- MQ traditionally made highly available using an HA cluster
  - IBM PowerHA for AIX (formerly HACMP), Veritas Cluster Server, Microsoft Cluster Server, HP Serviceguard, …
- HA clusters can:
  - Coordinate multiple resources such as application server, database
  - Consist of more than two machines
  - Failover more than once without operator intervention
  - Takeover IP address as part of failover
  - Likely to be more resilient in cases of MQ and OS defects

WebSphere MQ High Availability

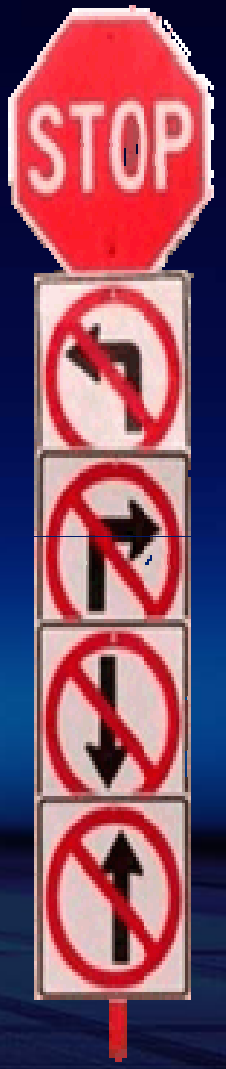
HA clusters (con't)

- In HA clusters, queue manager data and logs are placed on a shared disk
  - Disk is switched between machines during failover
- The queue manager has its own "service" IP address
  - IP address is switched between machines during failover
  - Queue manager's IP address remains the same after failover
- The queue manager is defined to the HA cluster as a resource dependent on the shared disk and the IP address
  - During failover, the HA cluster will switch the disk, take over the IP address and then start the queue manager

WebSphere MQ High Availability

Multi-instance QM or HA cluster?

- Multi-instance queue manager
  - ✓ Integrated into the WebSphere MQ product
  - ✓ Faster failover than HA cluster
    - ✓ Delay before queue manager restart is much shorter
  - ✗ Runtime performance of networked storage
  - ✗ Suitable storage can sometimes be a challenge
- HA cluster
  - ✓ Capable of handling a wider range of failures
  - ✓ Failover historically rather slow, but some HA clusters are improving
  - ✓ Capable of more flexible configurations (eg N+1)
  - ✗ Required MC91 SupportPac or equivalent configuration
  - ✗ Extra product purchase and skills required
- Storage distinction
  - Multi-instance queue manager typically uses NAS
  - HA clustered queue manager typically uses SAN



WebSphere MQ High Availability

Virtual Systems

- Another mechanism being regularly used
- When MQ is in a virtual machine … simply shoot and restart the VM
- “Turning it off and back on again”
- Can be faster than any other kind of failover

WebSphere MQ High Availability

HA applications – Connectivity

- If an application loses connection to a queue manager, what does it do?
  - Crash and Burn
  - Handle the failure and retry the connection
  - Reconnect automatically thanks to application container
    - WebSphere Application Server contains logic to reconnect JMS clients
  - Use MQ automatic client reconnection

WebSphere MQ High Availability

MQ Automatic client reconnection

- MQ client automatically reconnects when connection broken
  - MQI C clients and standalone JMS clients
  - JMS in app servers (EJB, MDB) does not need auto-reconnect
- Reconnection includes reopening queues, remaking subscriptions
  - All MQI handles keep their original values
- Can reconnect to same queue manager or another, equivalent queue manager
- MQI or JMS calls block until connection is remade
  - By default, will wait for up to 30 minutes
  - Long enough for a queue manager failover (even a really *slow* one)

WebSphere MQ High Availability

Automatic client reconnection

- Can register event handler to observe reconnection
- Not all MQI is seamless, but majority repaired transparently
  - Browse cursors revert to the top of the queue
  - Nonpersistent messages are discarded during restart
  - Nondurable subscriptions are remade and may miss some messages
  - In-flight transactions backed out
- Tries to keep dynamic queues with same name
  - If queue manager doesn't restart, reconnecting client's TDQs are kept for a while in case it reconnects
  - If queue manager does restart, TDQs are recreated when it reconnects

WebSphere MQ High Availability

Automatic client reconnection

- Enabled in application code, ini file or CLNTCONN definition
  - MQI: MQCNO_RECONNECT, MQCNO_RECONNECT_Q_MGR
  - JMS: Connection factory properties
- Plenty of opportunity for configuration
  - Reconnection timeout
  - Frequency of reconnection attempts
- Requires:
  - Threaded client
  - 7.0.1 server – including z/OS
  - Full-duplex client communications (SHARECNV >= 1)

WebSphere MQ High Availability

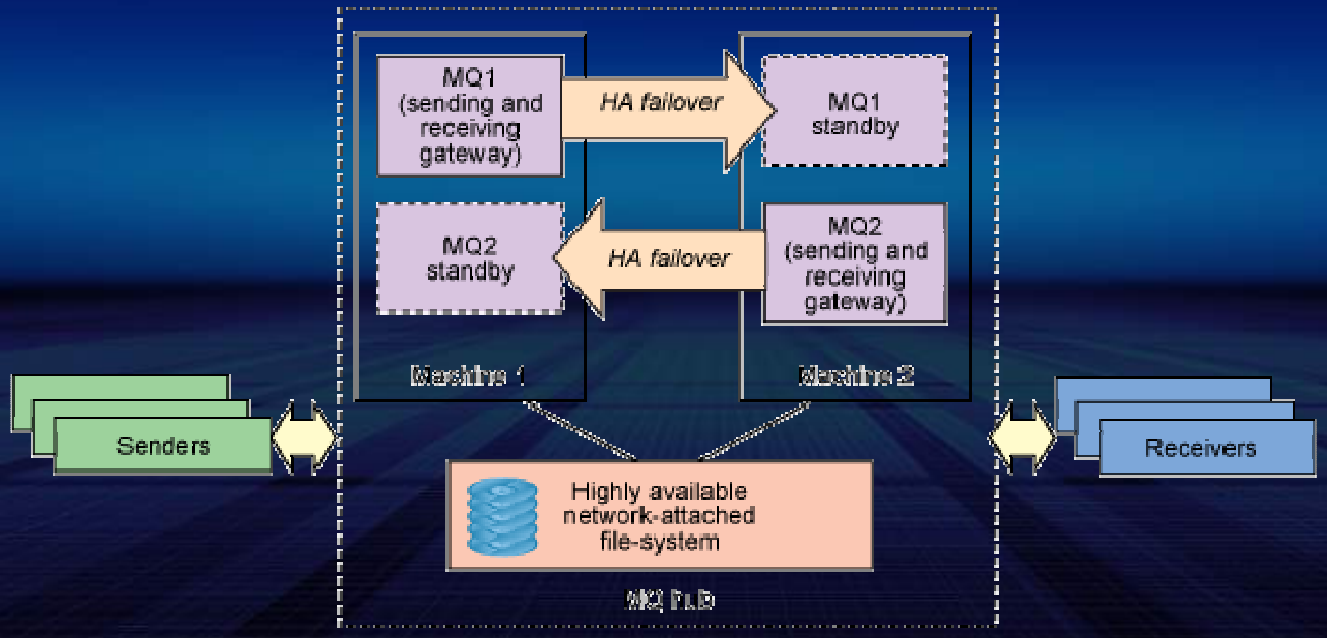
Client Configurations for Availability

- Use wildcarded queue manager names in CCDT
  - Gets weighted distribution of connections
  - Selects a “random” queue manager from an equivalent set
  - Setup MQExplorer or MO72 Support Pack
- Use multiple addresses in a CONNAME
  - Could potentially point at different queue managers
  - More likely pointing at the same queue manager in a multi-instance setup
- Use automatic reconnection
- Pre-connect Exit from V7.0.1.4
- Use IP routers to select address from a list
  - Based on workload or anything else known to the router
- Can use all of these in combination!

WebSphere MQ High Availability

Application Patterns for availability

- Article describing examples of how to build a hub topology supporting:
  - Continuous availability to send MQ messages, with no single point of failure
  - Linear horizontal scale of throughput, for both MQ and the attaching applications
  - Exactly once delivery, with high availability of individual persistent messages
  - Three messaging styles: Request/response, fire-and-forget, and pub/sub
- http://www.ibm.com/developerworks/websphere/library/techarticles/1303_broadhurst/1303_broadhurst.html



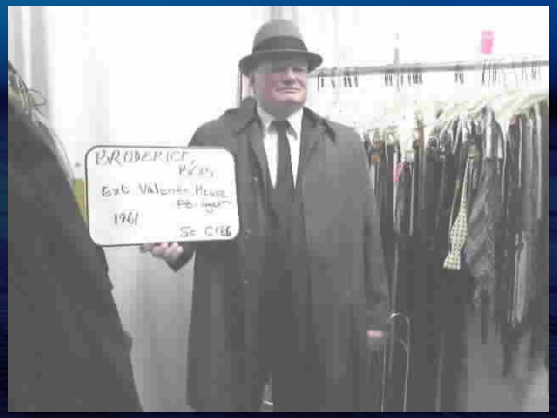
WebSphere MQ Best Practices

Good Bye, So Long and Thanks for the Fish !!!!!!!!!!!

WebSphere MQ Best Practices

Who is this guy ??????????????????

- Bobbee Broderick (1970)
- Experience
  - Wall St Consultant 25+ years (z, CICS, DB2) (MQ, MQSI)
  - MQ/MQSI/WMB since 1998
- IBM – ISSW 8 years
  - Healthchecks
  - Crit Sits
  - Architecture, programming, etc
- Tech Lead for ISSW for MQ and MQFTE (MFT)
  Also for MQAMS/ WMB
- Star of "***The Good Shepherd***"
- **BB Photography**
  www.bb-photography.org
- **Email – rkbroder@us.ibm.com**



WebSphere MQ High Availability

MQSeries Ten Commandments