MQ HA

Matt Leming
lemingma@uk.ibm.com

Agenda

- Overview
- MQ high availability
- NOTES: MQ disaster recovery

Introduction

- Availability is a very large subject
- You can have the best technology in the world, but you have to manage it correctly
- Technology is not a substitute for good planning and testing!

What is HA?

- **HA is the ability of a system to remain continuously operational for a suitably long period of time, even in the event of some component failures**
- **Typically achieved by…**

1) Eliminating single points of failure (SPOF)

- By adding redundancy
- Need to do this across all components, one SPOF is all that it takes to get a failure

2) Detecting issues as they occur and switching between redundant components

3) Ideally ensuring that the switching technology in #2 is itself redundant

- **However cost is often a factor in how far you go with this**

What is DR?

- **Getting applications running after a major (often whole-site) failure or loss**

- **It is not about high availability although often the two are related and share design and implementation choices**
  - "HA is having 2, DR is having them a long way apart"
  - More seriously, HA is about keeping things running, while DR is about recovering when HA has failed

- **Requirements driven by business, and often by regulators**
  - Data integrity, timescales, geography …

- **One major decision point: cost**
  - How much does DR cost you, even if it's never used?
  - How much are you prepared to lose?

DR vs HA

- Designs for HA typically involve a single site for each component of the overall architecture
- Designs for DR typically involve separate sites

- Designs for HA (and continuous availability) typically require no data loss
- Designs for DR typically can have limited data loss

- Designs for HA typically involve high-speed takeover
- Designs for DR typically can permit several hours down-time

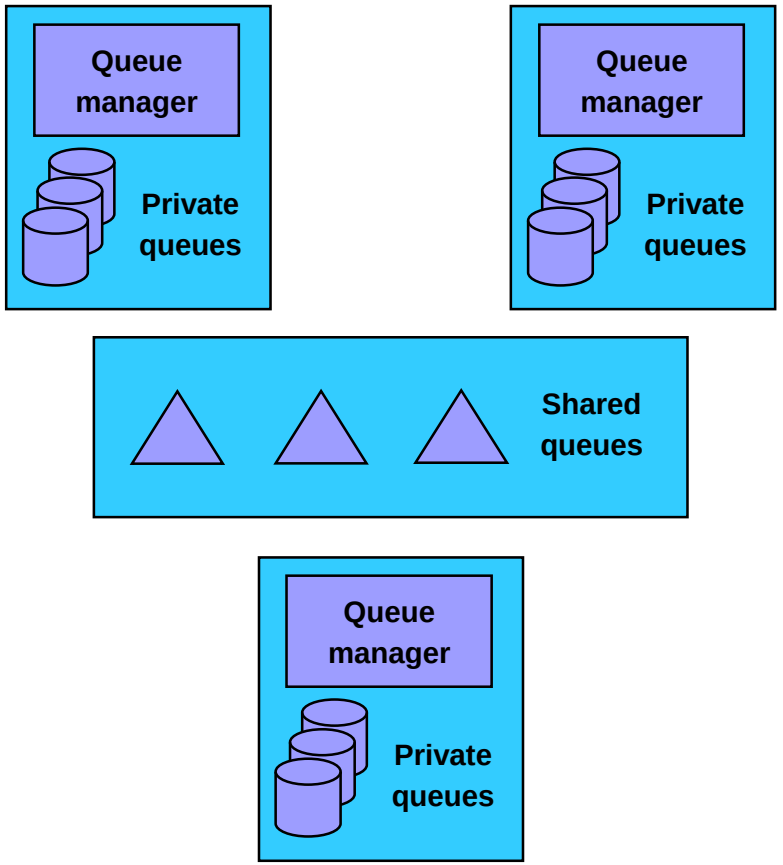
High availability

IBM MQ HA technologies

- **Multiple individual queue managers**
  - Queue sharing groups on z/OS
  - Clusters
- **Failover**
  - Multi-instance queue managers
  - MQ Appliance
  - HA clusters
- **Other options**
- **Client considerations**
  - Multiple IP addresses
  - Auto-reconnect

Queue sharing groups

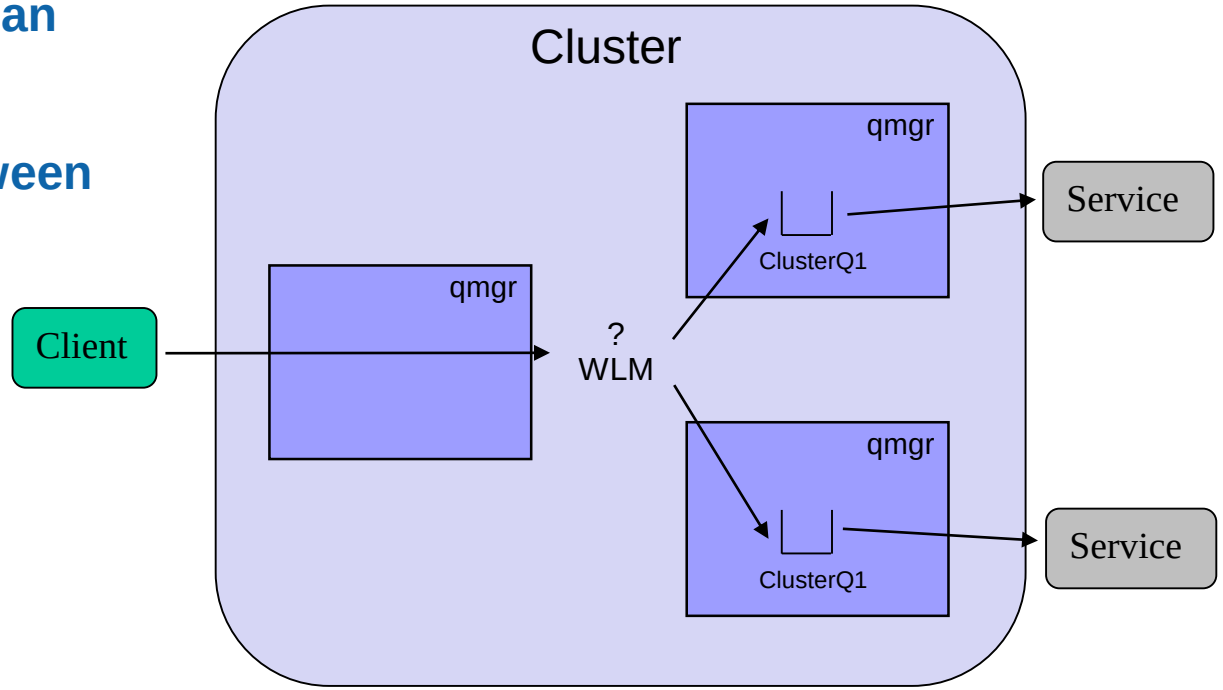
- **On z/OS, queue managers can be members of a queue sharing group**
- **Shared queues are held in coupling facilities**
  - All queue managers in the QSG can access the messages in a shared queue
- **Shared channels available too**
- **Benefits:**
  - Messages remain available even if a queue manager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



Covered in more detail in
“Introduction to z/OS Shared Queues” on Monday

MQ clusters

- Multiple instances of a queue with the same name can be deployed into a cluster
- The cluster WLM routine distributes messages between the instances
- If a queue manager holding one of the instances fails then its messages are unavailable until it recovers
- However the service provided by that queue is still available via the remaining instances



Covered in more detail in
“What can you achieve with MQ clusters” on Wednesday

Introduction to failover and MQ

- **Failover is the automatic switching of availability of a service**
  - For MQ, the “service” is a queue manager
- **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
  - Startup processing of queue manager following the failure
- **MQ offers several options for failover:**
  - Multi-instance queue managers
  - MQ Appliance
  - HA clusters

Failover considerations

- **Failover times are made up of three parts:**
  - Time taken to notice the failure
    - Heartbeat missed
    - Bad result from status query
  - Time taken to establish the environment before activating the service
    - Switching IP addresses and disks, and so on
  - Time taken to activate the service
    - This is queue manager 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

High availability: multi-instance queue managers

Multi-instance queue managers

- **Basic failover support without HA cluster**
  - Added in MQ 7.0.1

- **Two instances of a 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

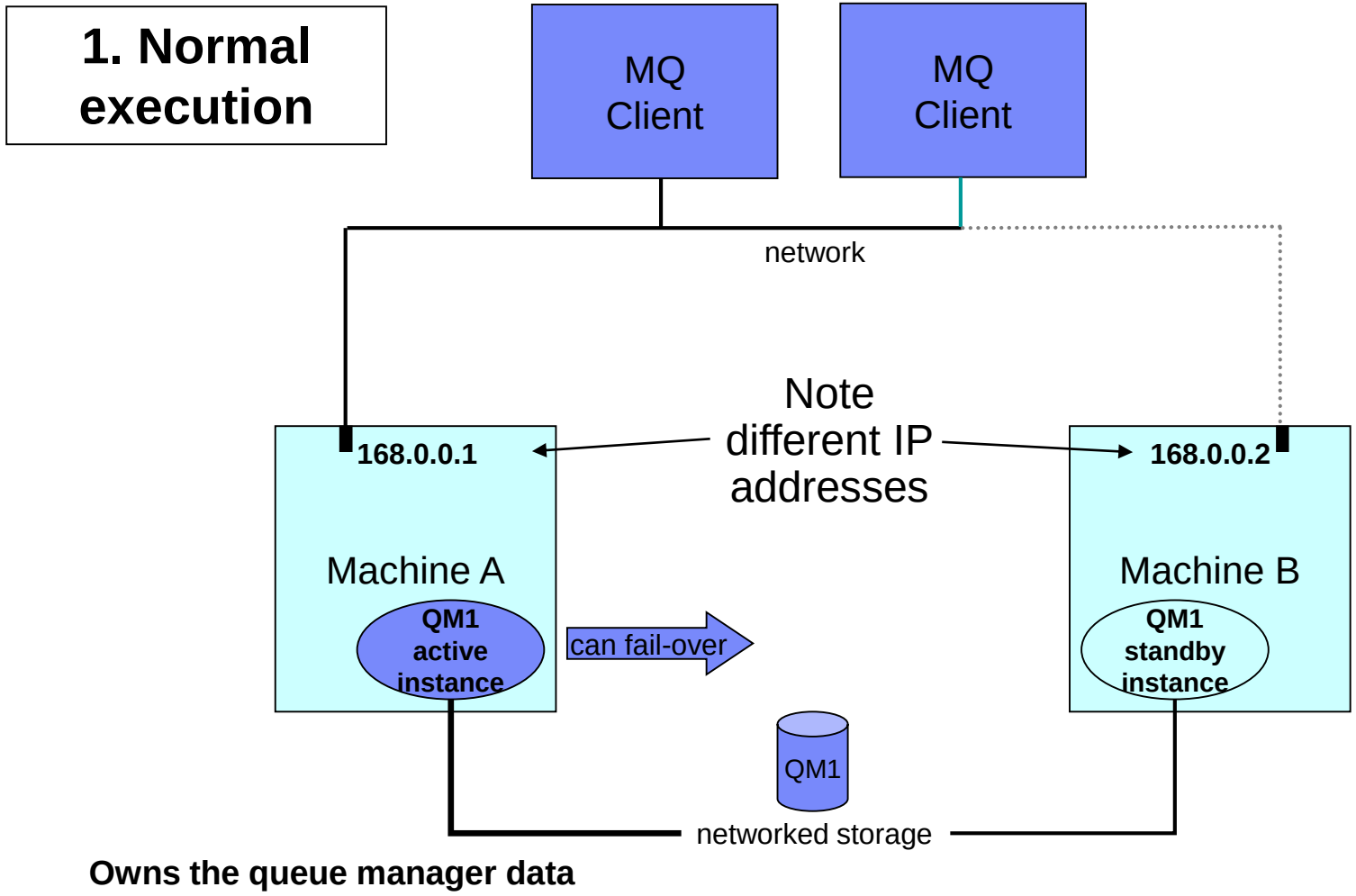
- **Instances are the SAME queue manager – only one set of data files**
  - Queue manager data is held in networked storage
  - Need to ensure storage is suitable. Use amqmfsck tool for this

- **Also available in PureSystems by using GPFS**

Setting up a multi-instance queue manager

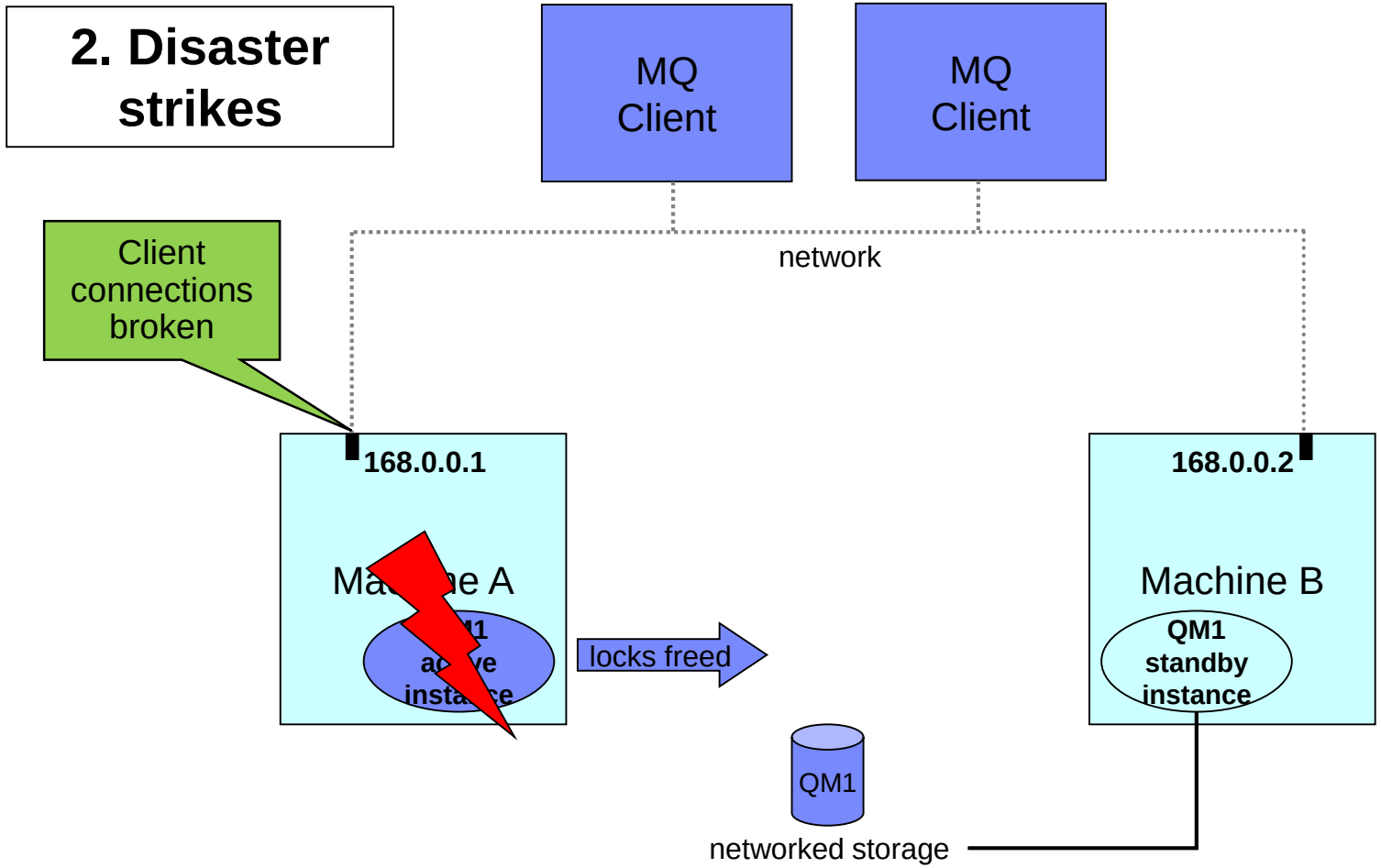
- **Install MQ on both machines**
- **Set up shared filesystems for QM data and logs**
- **Create the queue manager on machine1**
  - crtmqm –md /shared/qmdata –ld /shared/qmlog QM1
- **Define the queue manager on machine2 (can use dspmqinf to get details)**
  - addmqinf –vName=QM1 –vDirectory=QM1 –vPrefix=/var/mqm
    -vDataPath=/shared/qmdata/QM1
- **Start an instance on machine1 – it becomes active as it gets the lock first**
  - strmqm –x QM1
- **Start another instance on machine2 – it becomes standby**
  - strmqm –x QM1
- **That's it. If the queue manager instance on machine1 fails, the standby instance on machine2 takes over and becomes active**

Multi-instance queue managers



Owns the queue manager data

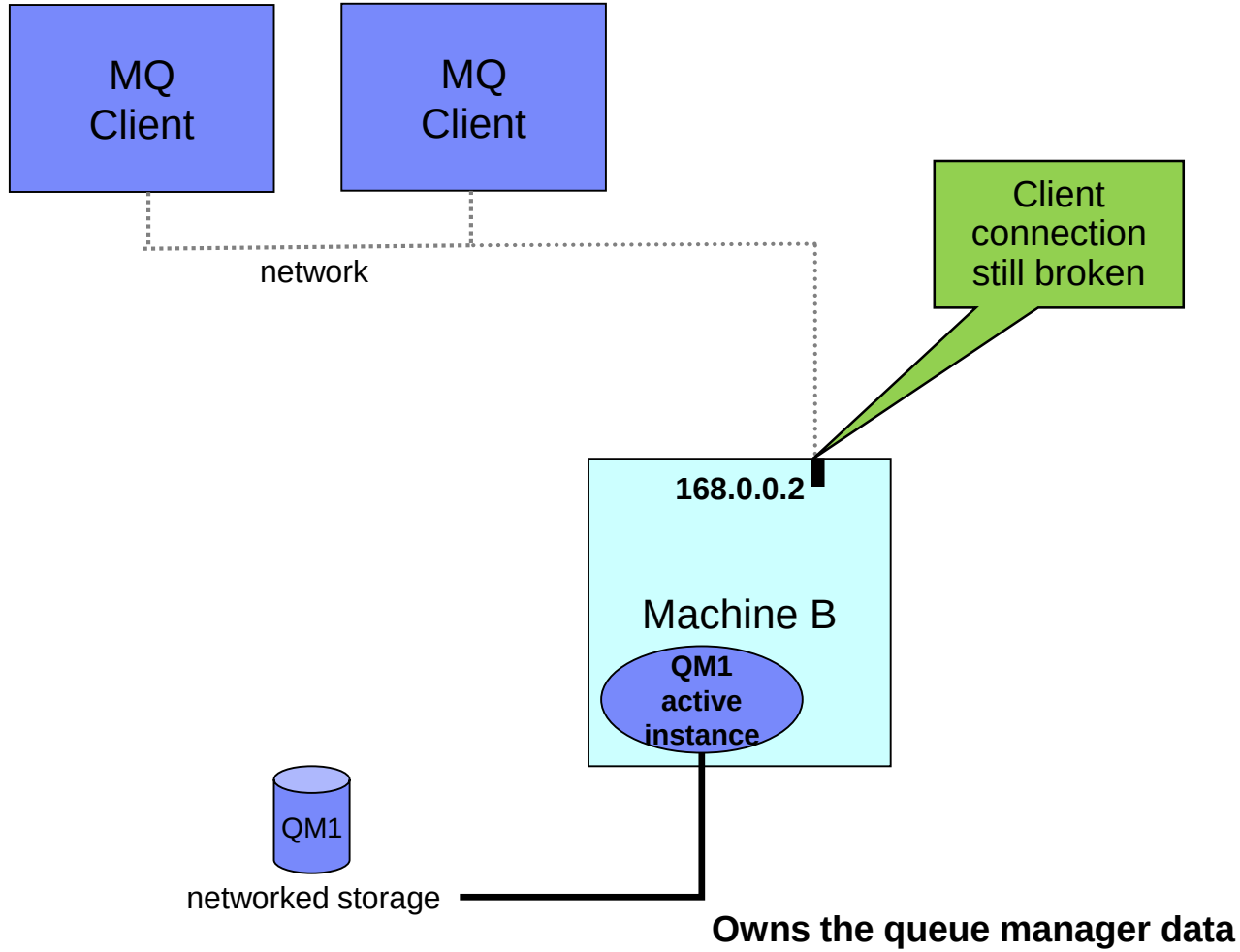
Multi-instance queue managers



Multi-instance queue managers

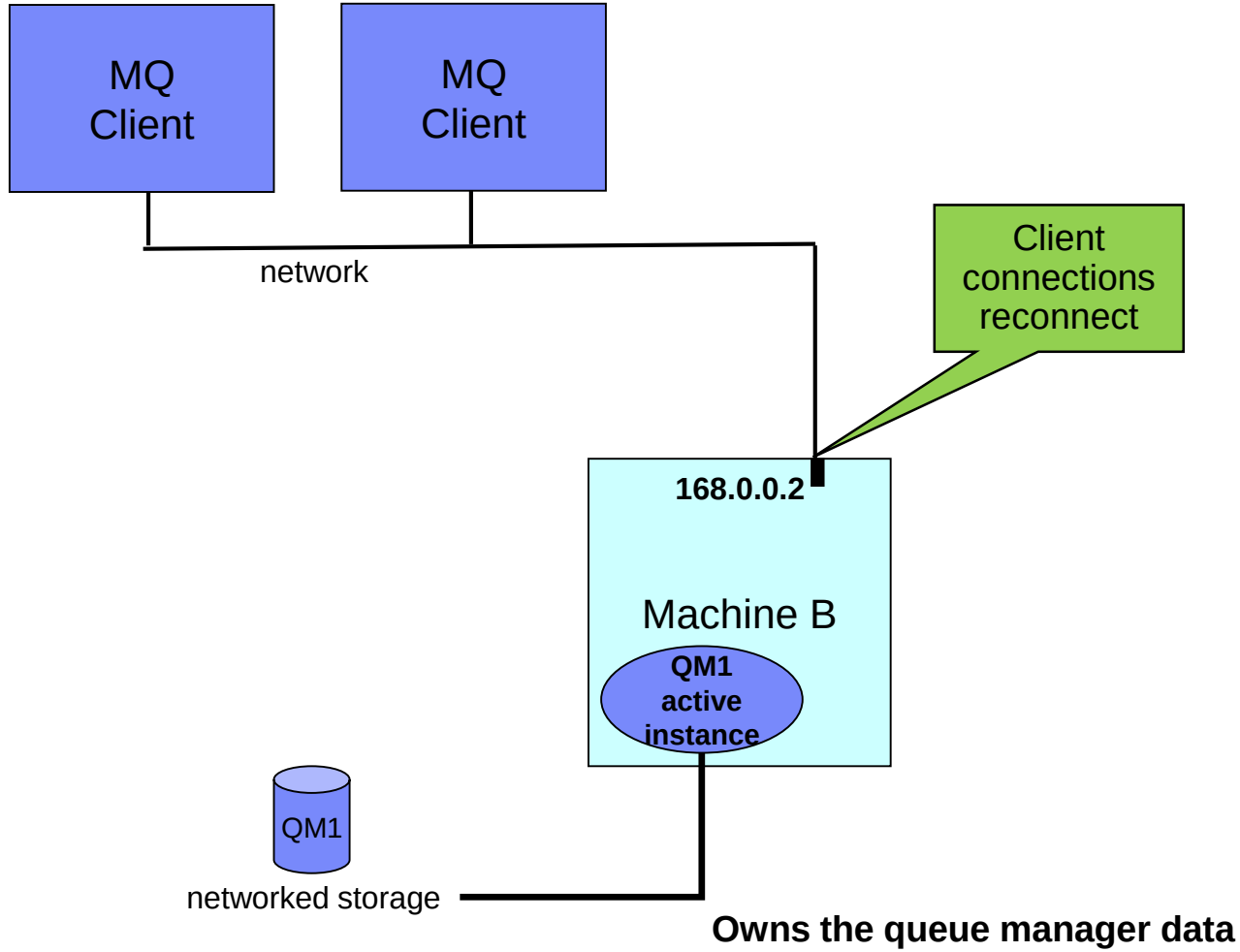
3. FAILOVER

Standby becomes active



Multi-instance queue managers

4. Recovery complete



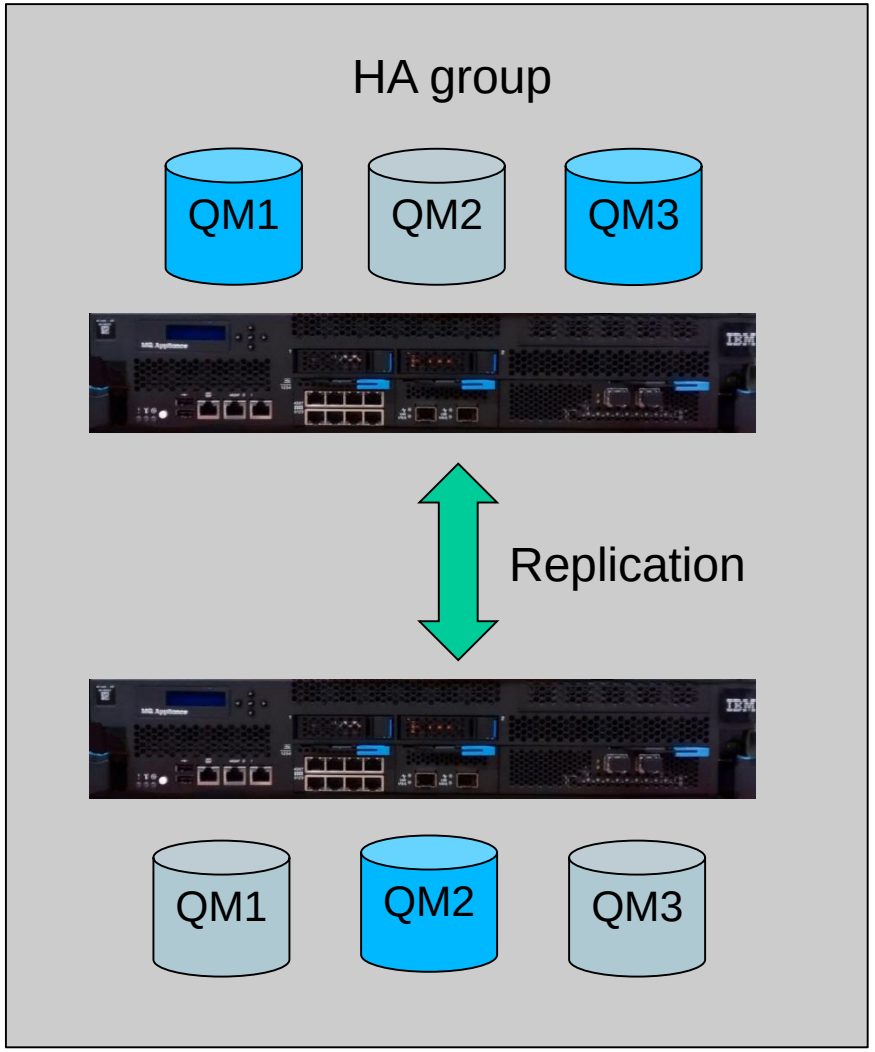
Dealing with multiple IP addresses

- **The IP address of the queue manager changes when it moves**
  - So MQ channel configuration needs list of possible endpoints
- **Connection name syntax extended to a comma-separated list**
  - CONNAME('168.0.0.1,168.0.0.2')
  - Needs 7.0.1 queue manager or client
- **Unless you use CCDT, external IPAT, or an intelligent router, or support pac MR01**
  - Basically anything that allows you to deal with multiple IP addresses

High availability: MQ Appliance

High availability for the MQ Appliance

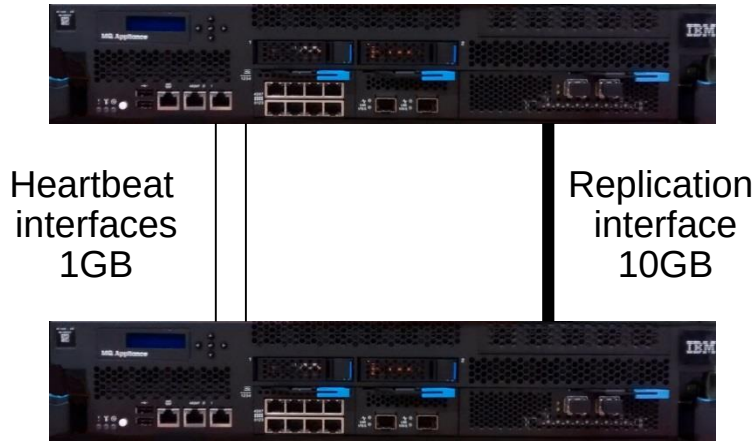
- **MQ Appliances can be deployed in HA groups**
  - Primary instance of queue manager runs on one
  - Secondary instance on the other
- **Queue manager data replicated across group**
  - Operations on primary queue manager automatically replicated to secondary
  - Appliances monitor one another and perform local restart/failover
- **Easier configuration than other HA solutions (no shared file system/shared disks)**
- **Supports manual failover, e.g. for rolling upgrades**
- **Replication is synchronous over ethernet, for 100% fidelity**
  - Routable but not intended for long distances
- **Available since original version of appliance**



Covered in more detail in
“MQ HA & DR Deep Dive” on Tuesday/Wednesday

Setting up HA for the MQ Appliance

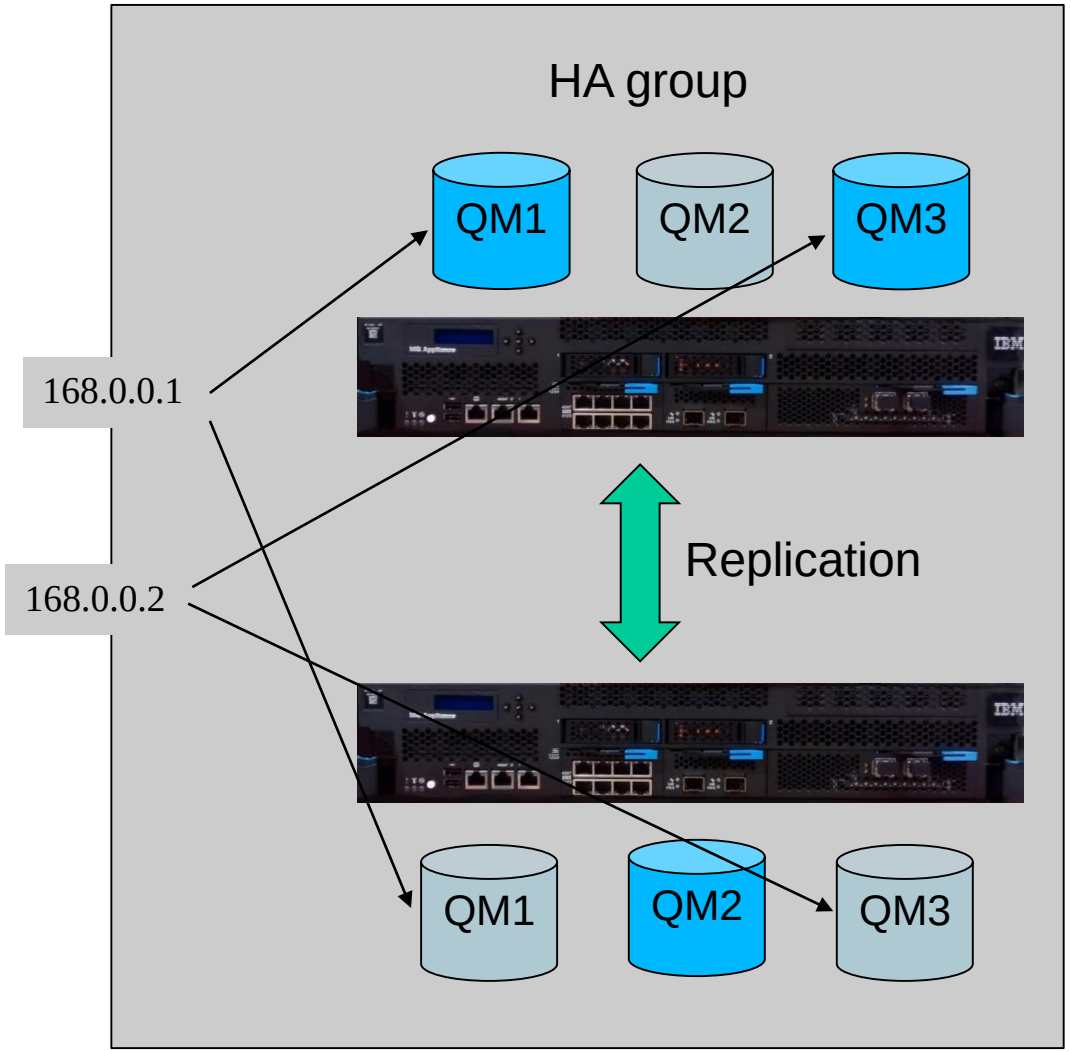
- **Simple to set up**
- **Connect appliances together**
  - Two cables for heartbeat interfaces
  - One cable for replication interface
- **On appliance 1 issue**
  - prepareha -s <some random text> -a <address of appliance2>
- **On appliance 2 issue**
  - crthagrp -s <the same random text> -a <address of appliance1>
- **Create an HA queue manager**
  - crtmqm **-sx** HAQM1



- **Note that there is no need to run strmqm.**
  Queue managers will start and keep running unless explicitly ended with endmqm

Floating IP address

- **Initial HA support on the appliance used a pair of IP addresses (one for each appliance)**
  - Other queue managers/clients connect using the same technologies as for multi-instance queue managers
- **The 9.0.1 firmware update included support for floating IP addresses**
  - I.e. a single IP address that can be used to reach a queue manager regardless of where it is running
  - Configured per queue manager
    - Different IP address for each queue manager



High availability: HA clusters

HA clusters

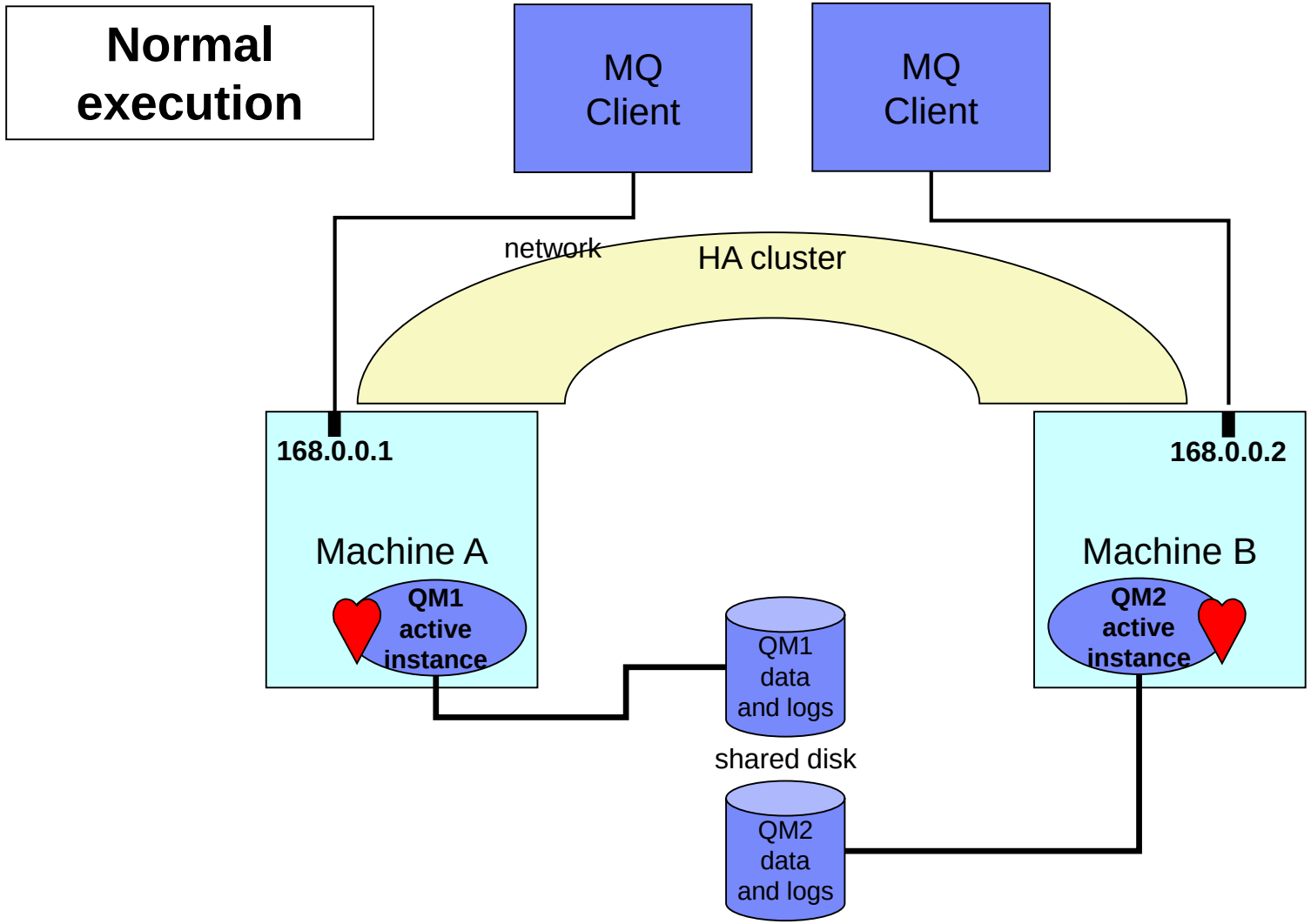
- **MQ traditionally made highly available using an HA cluster**
  - IBM PowerHA for AIX (formerly HACMP), Veritas Cluster Server, Microsoft Cluster Server, HP Serviceguard, …

- **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 issues

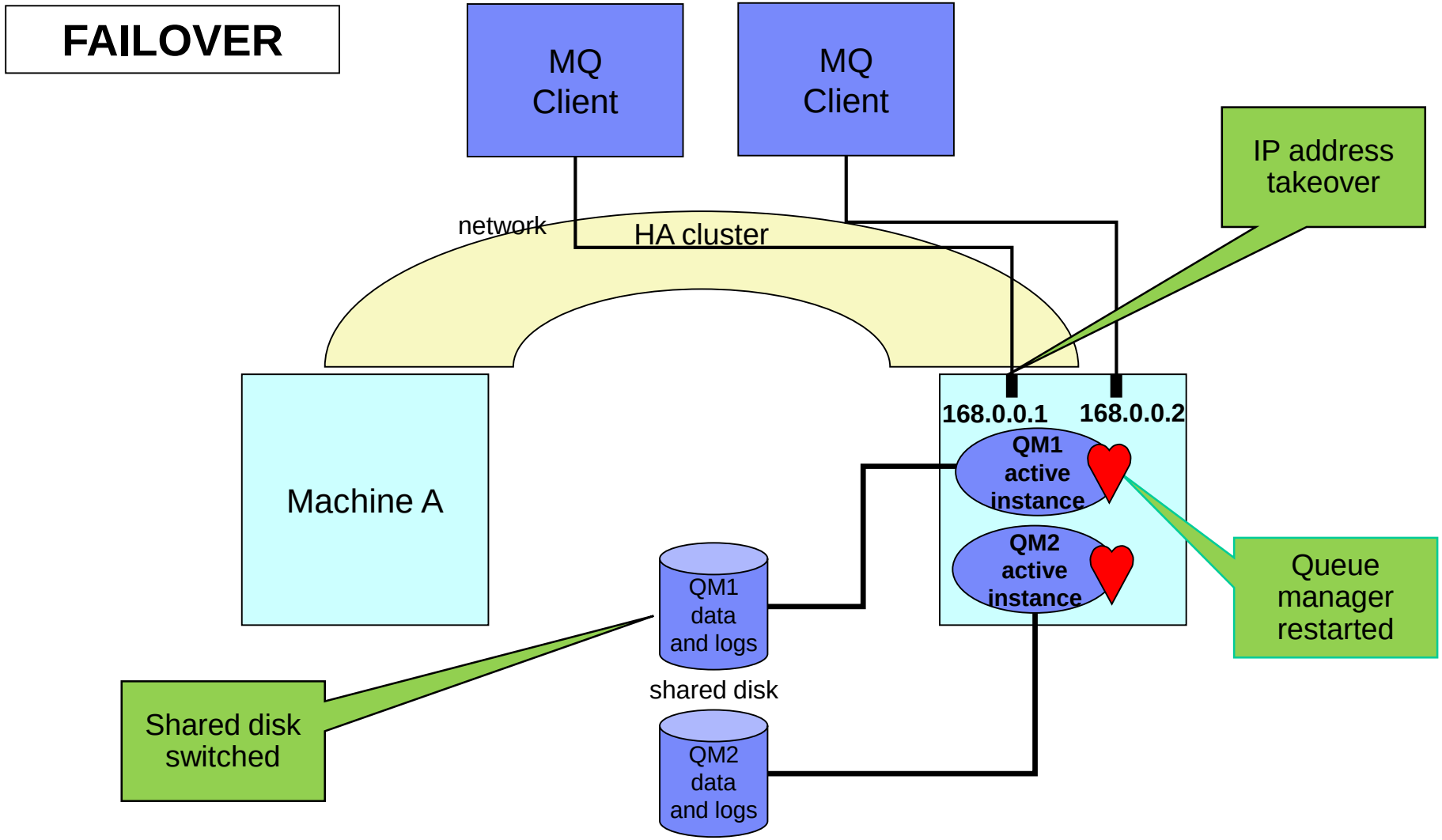
HA clusters – high level process

- **MQ installed on machines in cluster**
- **Queue manager data and logs are placed on a shared disk**
  - Disk is switched between machines during failover
- **The queue manager given 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
- **Scripts are provided to start, monitor and stop the queue manager**
  - The HA cluster runs these as appropriate
- **Setup instructions and sample scripts provided in KC**
  - Similar to what used to be in support pac MC91, which has been withdrawn as of MQ 7.0.1 as the process is now simpler because of introduction of multi-instance queue managers

MQ in an HA cluster – active/active



MQ in an HA cluster – active/active



Multi-instance QM or HA cluster?

- **Multi-instance queue manager**
  - Integrated into the MQ product
  - Faster failover than HA cluster
    - Delay before queue manager restart is much shorter
  - Runtime performance of networked storage can be an issue
  - System administrator responsible for restarting the standby instance after failover
  - IP address changes which can require some care

- **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)
  - Extra product purchase and skills required
  - No IP address change

High availability: client considerations

HA applications – MQ connectivity

- **If an application loses connection to a queue manager, what does it do?**
  - End abnormally
  - Handle the failure and retry the connection
  - Reconnect automatically thanks to application container
    - For example message driven beans running in application server
  - Use MQ automatic client reconnection

Automatic client reconnection

- **MQ client automatically reconnects when connection broken**
  - MQI C clients and standalone JMS clients
  - JMS in app servers (EJB/Servlet, MDB) can't auto-reconnect
    - Not needed for MDB anyway as resource adapter does that
    - EJB/Servlets need to retry
- **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)

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

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)

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
- **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!**

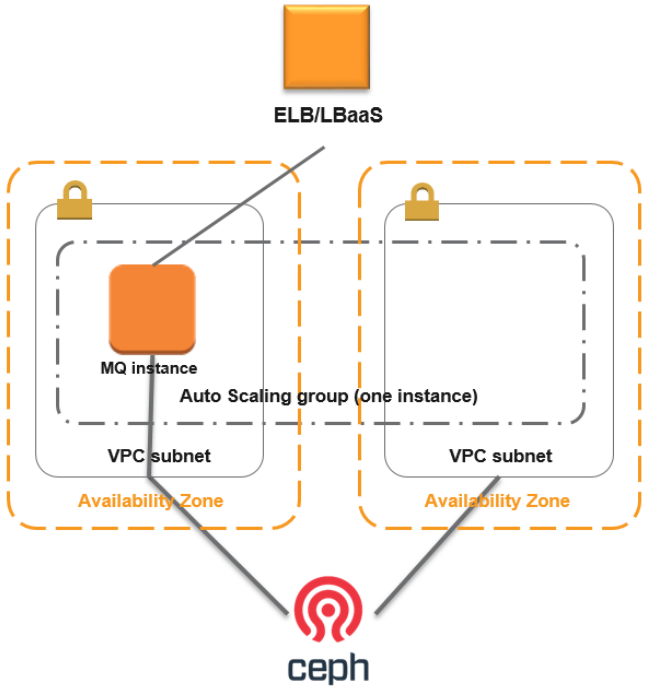
High availability: other options

Virtual images

- 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

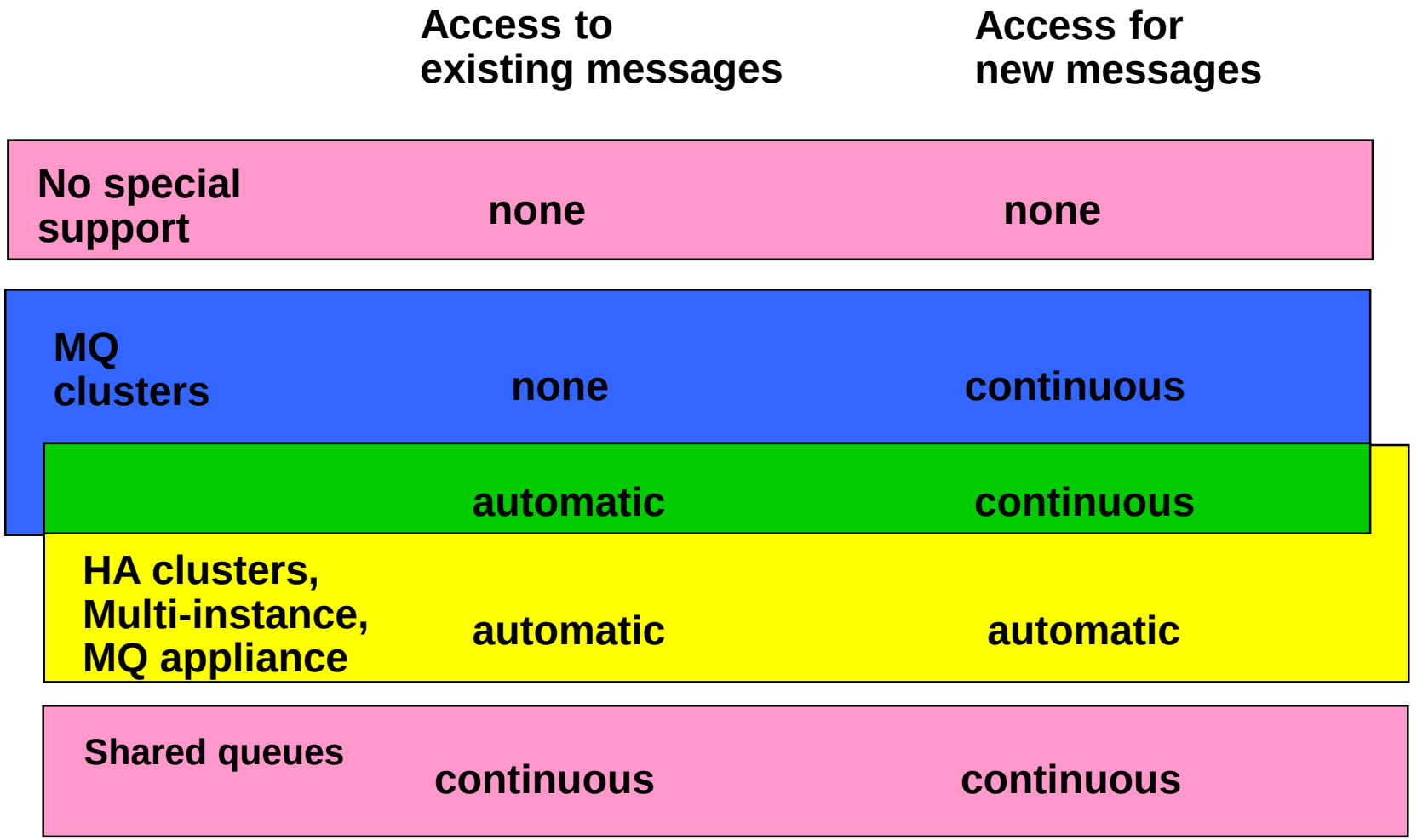
HA and DR for MQ in clouds

- **Fundamentally the same options as discussed earlier**
- **Orchestration tools to help provide fail over**
- **Different filesystems with different characteristics**
  - https://github.com/ibm-messaging/mq-aws/tree/master/drbd
  - https://www.ibm.com/developerworks/community/blogs/messaging/entry/mq_aws_efs
  - https://www.ibm.com/developerworks/community/blogs/messaging/entry/POC_MQ_HA_using_Ceph



High availability: summary of main options

Summary

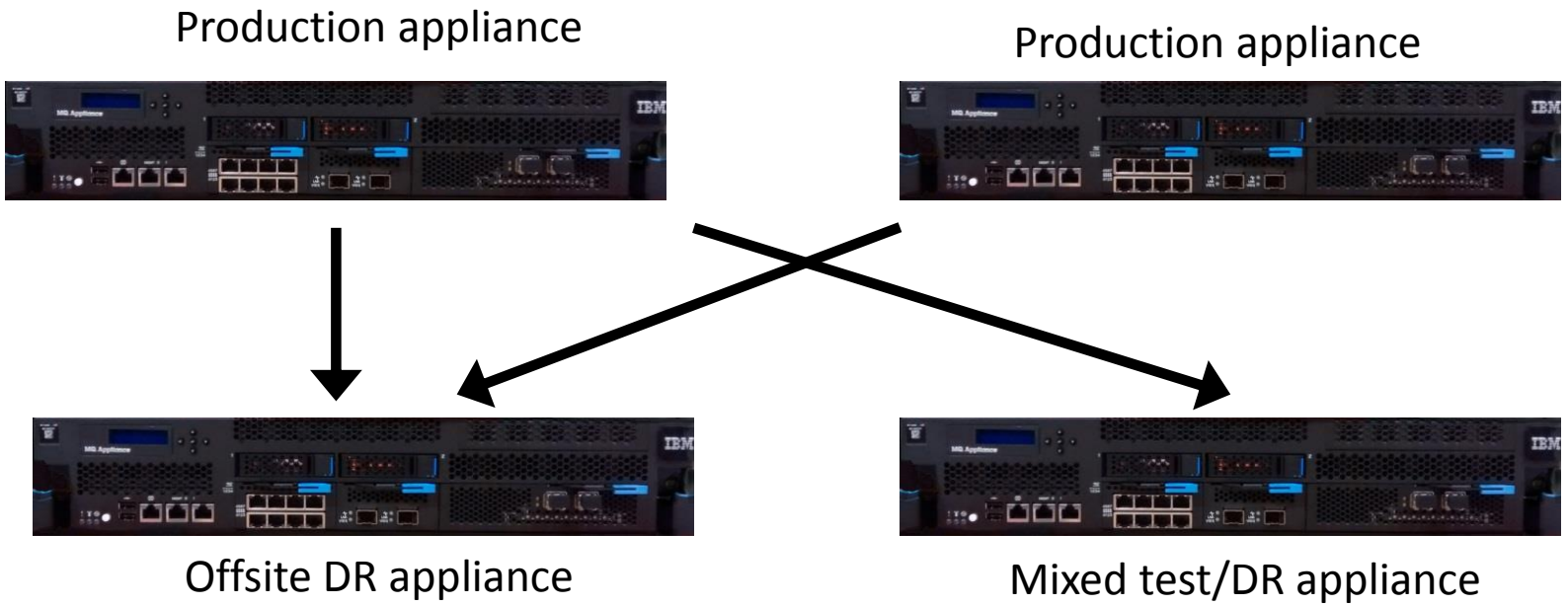


Disaster recovery

Disaster recovery: MQ Appliance

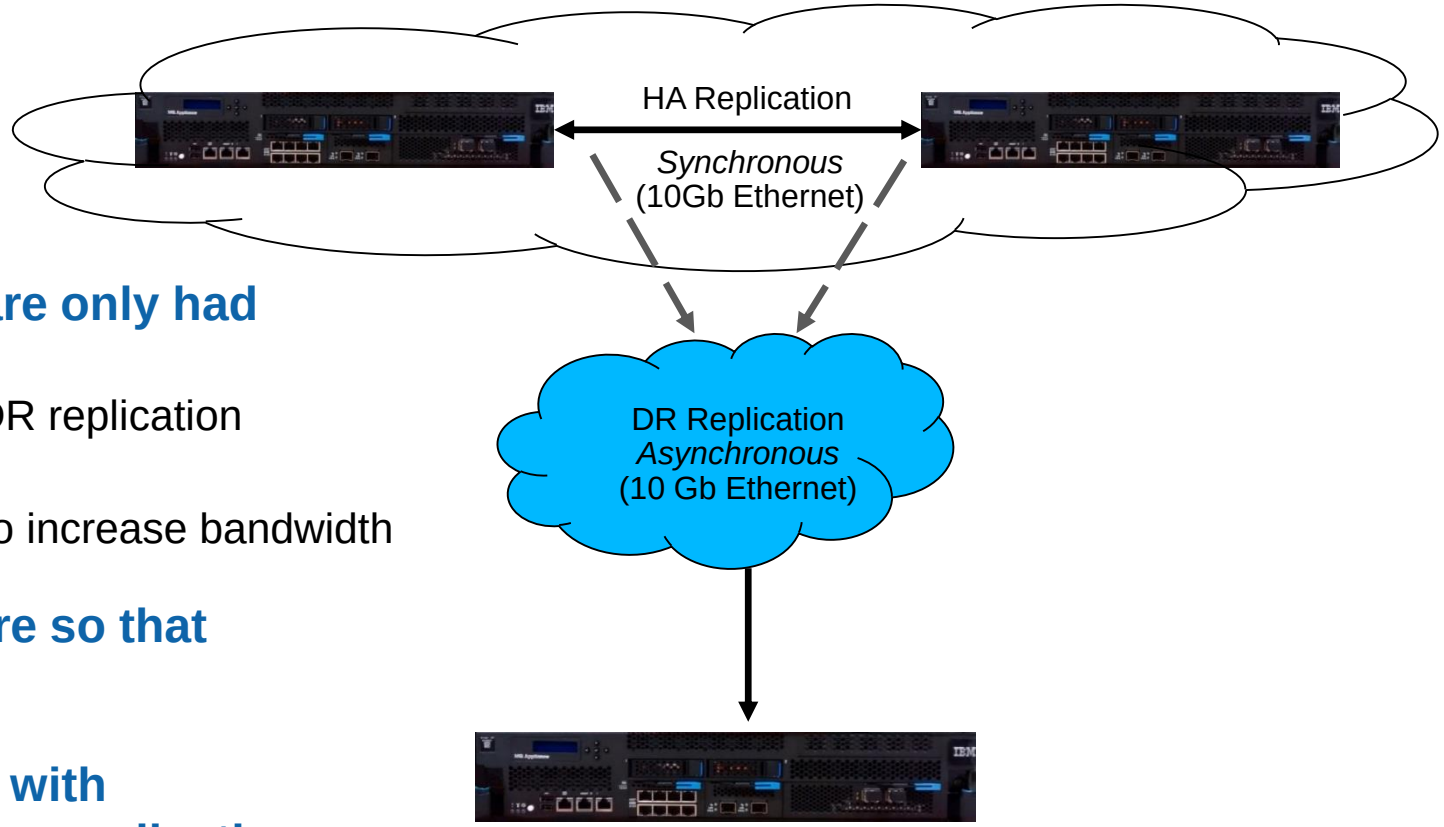
DR for the MQ Appliance

- 8.0.0.4 firmware update added DR support to the MQ Appliance
- Asynchronous replication of queue manager data, much more tolerant to latency than HA
- No concept of DR group, DR is added for specific queue managers
- Recovery/failback action requires manual steps (simply issuing a couple of commands)



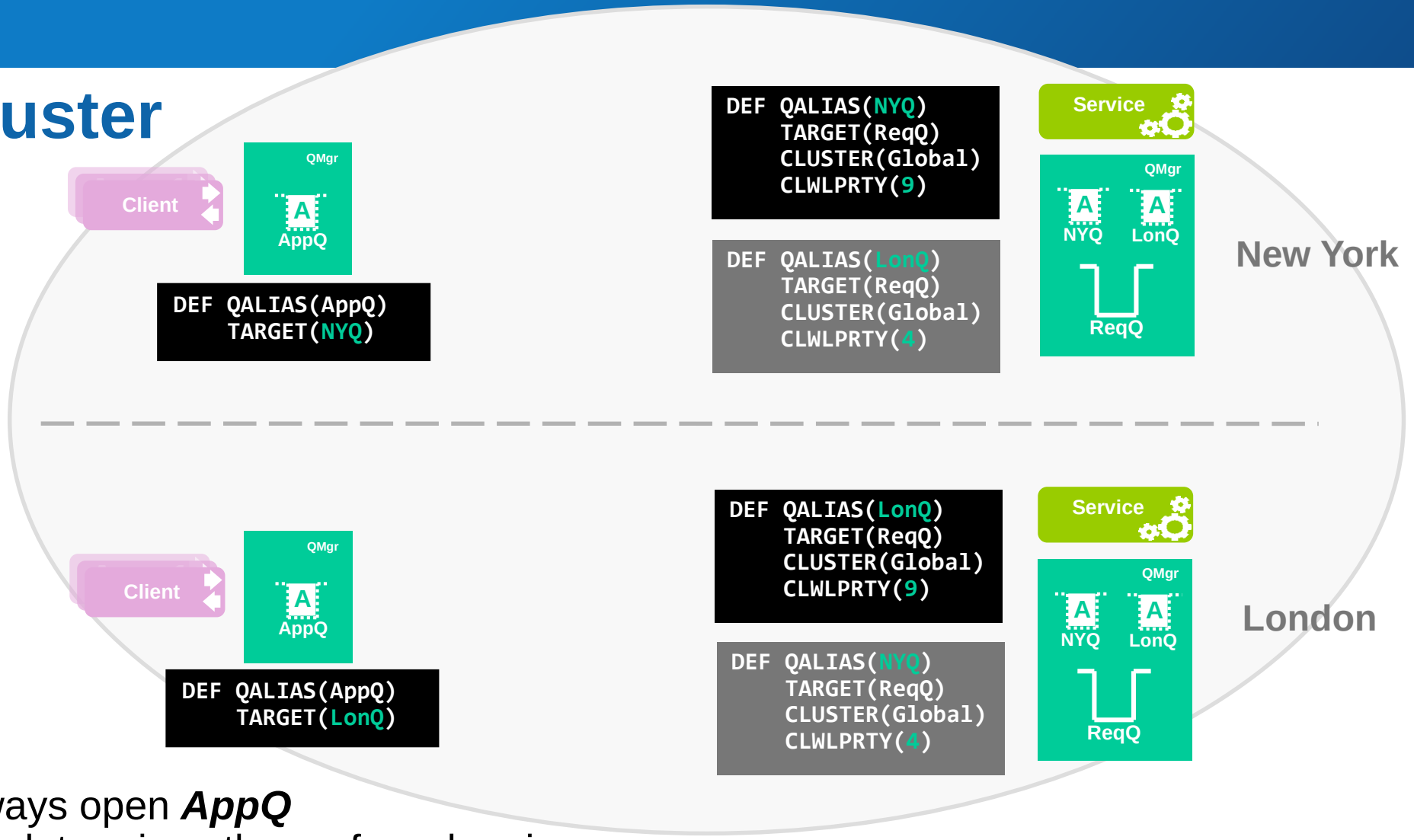
Disaster recovery for HA groups

- **DR support in 8.0.0.4 had one major restriction**
- **You could have**
  - HA between a pair of appliances
  - DR between two or more appliances
  - But not both
- **Furthermore, the available M2000 hardware only had two 10GB ethernet ports**
  - One needed for HA replication, the other for DR replication
  - Leaving 1GB interfaces for messaging traffic
    - Although link aggregation could be used to increase bandwidth
- **8.0.0.5 lifted this restriction in the firmware so that HA and DR could be used together**
- **M2001 hardware refresh also came along with 4 * 10 GB ports removing the constraint on application bandwidth**



Active/Active with MQ clustering

One cluster



- Clients always open ***AppQ***
- Local alias determines the preferred region
- Cluster workload priority is used to target geographically local cluster aliases
- Use of CLWLPRTY enables automatic failover (based on status of channel)
  - CLWLRANK can be used for manual failover (not based on status of channel)

Summary

Recap

- Overview
- MQ high availability
- NOTES: MQ disaster recovery

Questions & Answers