

MQ Appliance HA and DR Deep Dive

Matt Leming – lemingma@uk.ibm.com

MQ Development

Please Note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Agenda

- Introduction
- HA in the MQ Appliance
- DR in the MQ Appliance
- Combining HA and DR
- Communication considerations
- Performance

Introduction

Very high level overview and terminology

- **High availability:** providing *continuity of service* after a ‘component’ failure (e.g. disk, software bug, network switch, power supply)

- **Disaster recovery:** providing a means to *re-establish* service after a catastrophic failure (e.g. entire data center power loss, flood, fire)

- **Primary and secondary – which instance is *currently* ‘active’**
 - ▶ Has direct impact on commands you execute and behaviour
 - ▶ N.B. in disaster recovery ‘active’ doesn’t necessarily mean ‘running’

- **Main and recovery– which instance is *usually* ‘active’**
 - ▶ Doesn’t effect behaviour but useful concept in discussions and documentation

HA and DR – comparison of aims

High availability

- 100% data retention
- Automatic failover
- Short distances
 - ▶ meters or miles

Disaster recovery

- Some (minimal) data loss acceptable
- Manual fail over
- Long distances
 - ▶ out of region

In both cases we want:

- Minimal effect possible on performance
- As little impact on applications as possible

Timeline / Versions

- **Support for HA was a key feature of the IBM MQ Appliance from the first release and has been improved in every fix pack since**
 - ▶ Strongly recommend moving to latest release to get all enhancements
- **Support for DR was added in 8.0.0.4 and it uses some of the technology used for HA**
 - ▶ 8.0.0.4 released November 2015
- **This presentation is based on the capabilities of the 8.0.0.5 firmware for the IBM MQ Appliance**
 - ▶ Allows HA and DR support to be used together
 - ▶ 8.0.0.5 released May 2016

HA in the MQ Appliance

Setting up HA

Implementing HA is a simple with the MQ Appliance!

1. Connect two appliances together

2. On Appliance #1 issue the following command:

```
prepareha -s <some random text> -a <address of appliance2>
```

3. On Appliance #2 issue the following command:

```
crthagrp -s <the same random text> -a <address of appliance1>
```

4. Then create an HA queue manager:

```
crtmqm -sx HAQM1
```

■ **That's it!**

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



Physical layout



Heartbeat Connections
(1 Gb Ethernet)

eth13, eth17

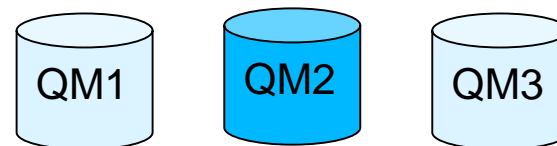
Replication Connection
(10 Gb Ethernet)

eth21





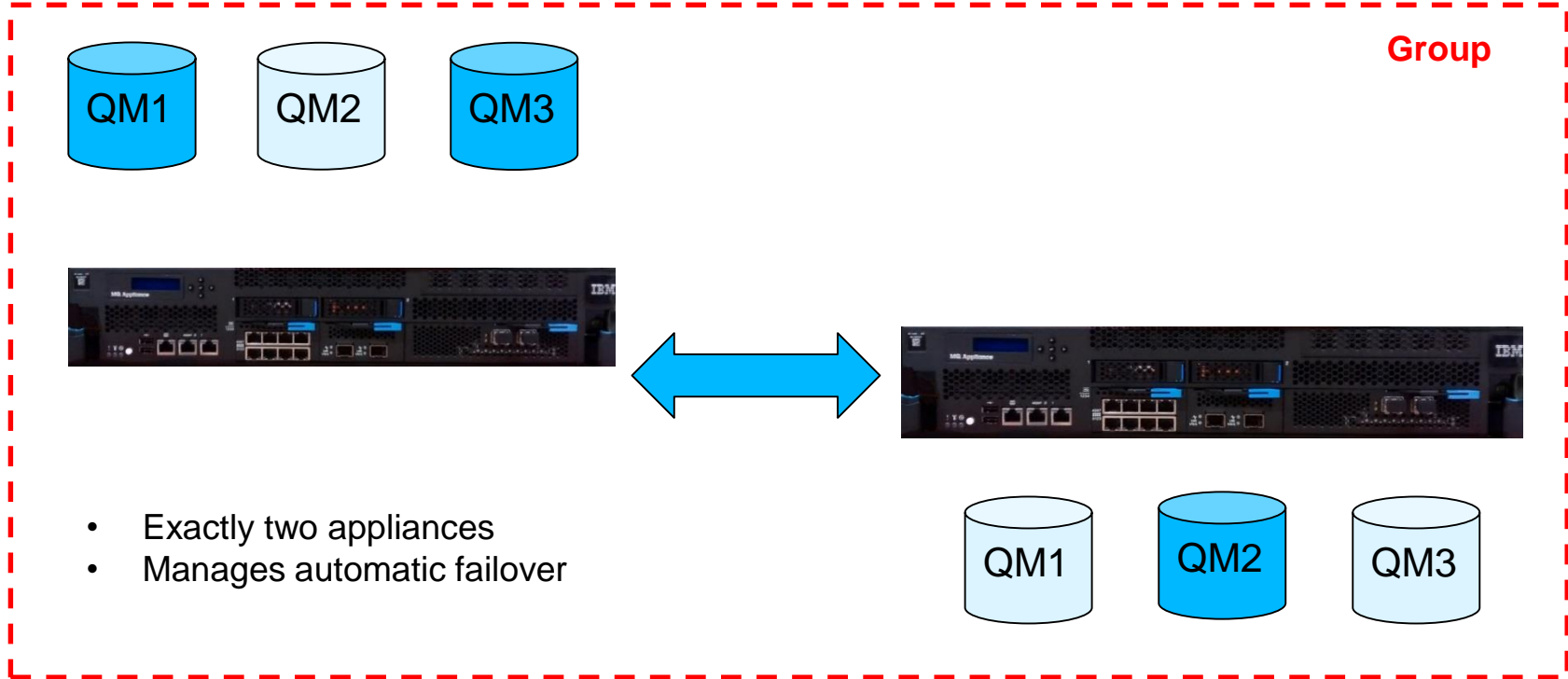
Fully synchronous replication



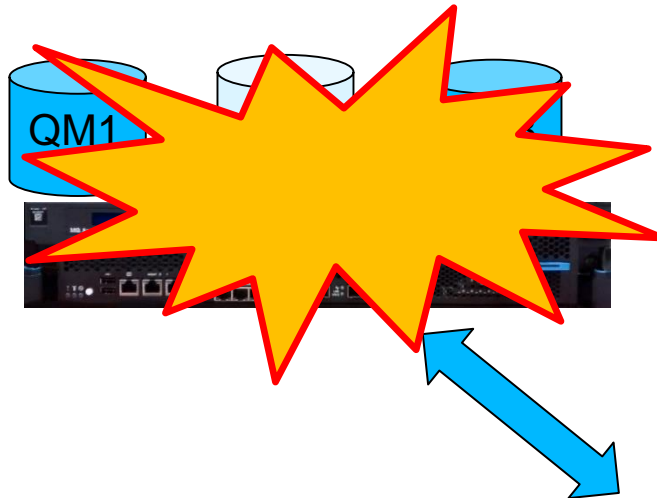
Key design points:

- No (persistent) message loss
- No external dependencies
- Transparent to application

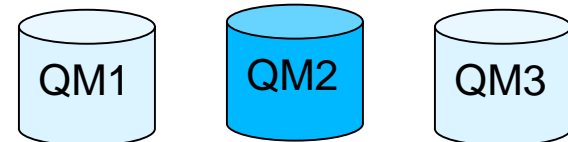
HA groups



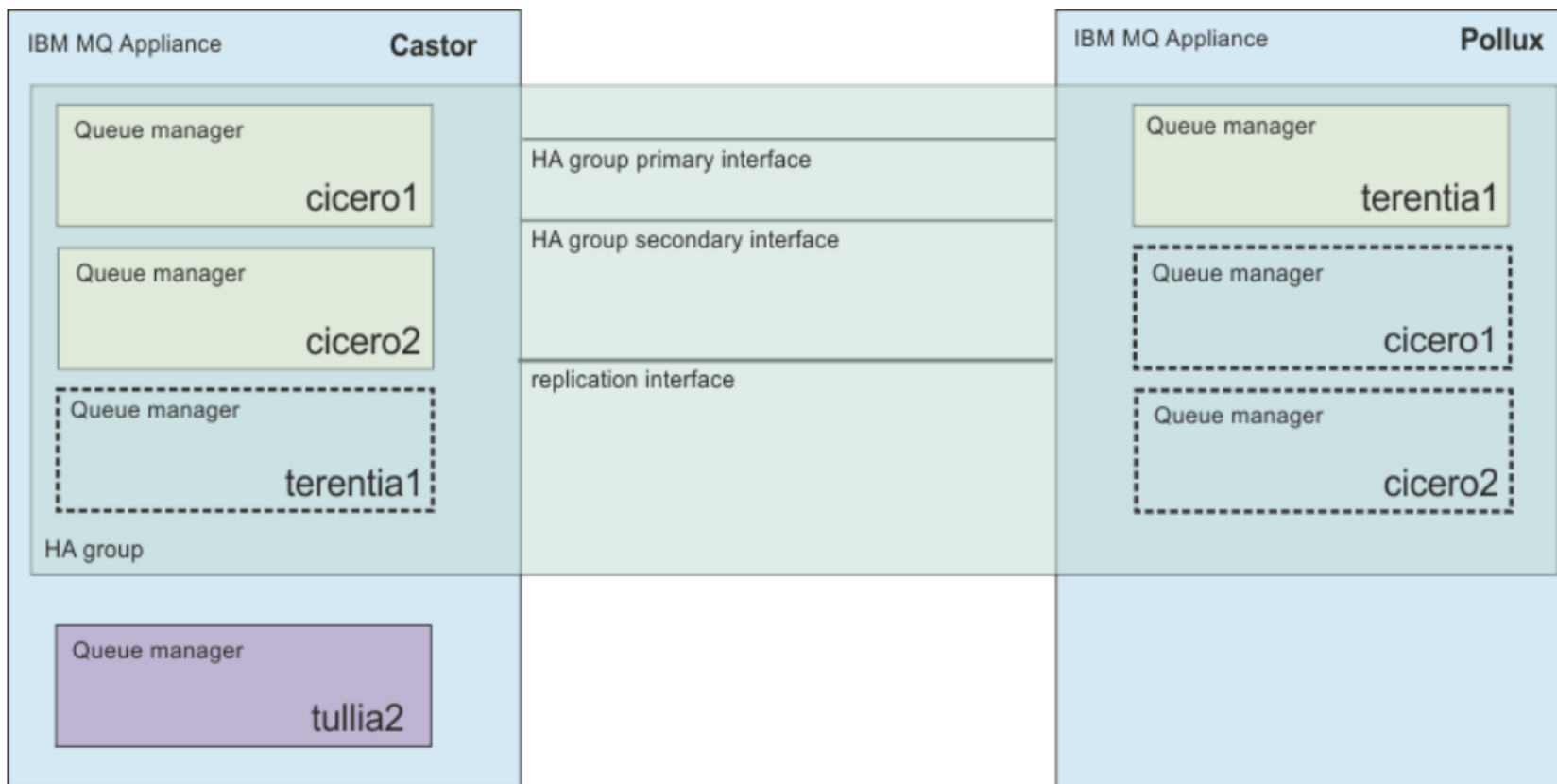
Disaster (small) strikes...



- Queue manager or entire appliance level failures
- Will restart on local system if possible



Designing a group



Notes: Designing a group

- This image is straight from the KnowledgeCentre and gives a good overview of the possible combinations of queue managers in an HA group.
- As long as IP addresses etc. for the three HA interfaces are correctly pre-configured, defining a group is as simple as executing the 'crthagrp/prepareha' commands.
 - ▶ Appliances can only be in exactly one group of exactly two appliances
- Queue managers may be added to the group at crtmqm time, or after creation using the 'sethagr' command
 - ▶ Queue managers can be active on either appliance (both appliances can simultaneously be running different active queue managers). Up to 16 active/passive instances per appliance are permitted.
- Unlimited (other than by storage capacity etc.) non-HA queue managers may also be present on either appliance.
 - ▶ This might be desirable for example if you have applications/queue managers with different QOS agreements, or Test and production environments on the same system.

HA queue managers in MQ Console

HA Group Appliance #1

Name	Running TCP listener ports	Status	High Availability
HAQM1	1511	↑ Running	REPLICATED
HAQM2		↑ Running elsewhere	REPLICATED
QM1	1414	↑ Running	

Total: 3 Selected: 0 1 Last updated: 4:09:57 PM

HA Group Appliance #2

Name	Running TCP listener ports	Status	High Availability
HAQM1		↑ Running elsewhere	REPLICATED
HAQM2	1512	↑ Running	REPLICATED
QM2	1415	↑ Running	

Total: 3 Selected: 0 1 Last updated: 4:12:06 PM

HA queue managers in MQ Console: After failover

- Appliance #1 is now in *standby*
- All HA queue managers are now *running* on Appliance #2
- The console shows the high availability alert, and a menu to allow you to see the status and to suspend or resume the appliance in the HA group.

The screenshot shows the IBM MQ Console interface. At the top, there is a navigation bar with 'IBM MQ Console', 'Dashboard', and 'Appliance'. On the right, there is a 'High Availability' alert icon and the user 'admin'. Below the navigation bar, there is a 'Welcome' message and a '+ Add MQ Object Widget' button. The main content area displays a 'Queue Managers' widget with a table of queue managers. The table has columns for Name, Running TCP listener ports, Status, and High Availability. The status for HAQM1 and HAQM2 is 'Running', and for QM2 it is 'Running'. The High Availability column shows 'REPLICATED' for HAQM1 and HAQM2. A dropdown menu is open over the 'High Availability' column, showing options: 'This appliance: Online', 'Appliance 'MQAppl1': Standby', and 'Suspend this appliance...'. The table footer shows 'Total: 3 Selected: 0' and 'Last updated: 4:15:27 PM'.

Name	Running TCP listener ports	Status	High Availability
HAQM1	1511	Running	REPLICATED
HAQM2	1512	Running	REPLICATED
QM2	1415	Running	

HA failover (CLI view)

On Appliance #2:

- HAQM2 is running there, on its primary and preferred location
- HAQM1 is running on its primary – Appliance #1, so is secondary on Appliance #2

Before failover

```
M2000(mqcli)# status HAQM2
QM(HAQM2)                Status(Running)
CPU:                      0.00%
Memory:                   198MB
Queue manager file system: 118MB used, 3.0GB allocated [4%]
HA role:                  Primary
HA status:                Normal
HA control:               Enabled
HA preferred location:    This appliance
M2000(mqcli)# status HAQM1
QM(HAQM1)                Status(Running elsewhere)
HA role:                  Secondary
HA status:                Normal
HA control:               Enabled
HA preferred location:    Other appliance
M2000(mqcli)# _
```

On Appliance #2 – after failover:

- HAQM1 is now running on Appliance #2

After failover

```
M2000(mqcli)# status HAQM1
QM(HAQM1)                Status(Running)
CPU:                      0.09%
Memory:                   199MB
Queue manager file system: 118MB used, 3.0GB allocated [4%]
HA role:                  Primary
HA status:                Secondary appliance unavailable
HA control:               Enabled
HA preferred location:    Other appliance
M2000(mqcli)# _
```

Appliance HA commands

Command	Description
crthagr	Create a high availability group of appliances, run after prepareha
dsphagr	Display the status of the appliances in the HA group
dlthagr	Delete an HA group, can't have any HA queue managers at this point. Only entered on one appliance
makehaprimary	Specifies that an appliance is the 'winner' when resolving a partitioned situation in the HA group. Works at the queue manager level
prepareha	Prepare an appliance to be part of an HA group
sethagr	Pause and resume an appliance in an HA group, or add/remove a queue manager to/from the group
status	Per queue manager detailed information including replication status (e.g. percentage complete when re-syncing from lost connection)
sethappreferred	Set which appliance in the HA group the queue manager should run on, if that appliance is available. Might trigger a failover!
clearhappreferred	Mark a queue manager as having no preferred appliance

HA – requirements / restrictions

- **MUST ensure redundancy between heartbeat (1GB) links**
 - ▶ Shared nothing – power, switches, routers, cables
 - ▶ Minimises risk of partitioning ('split brain')
- **Less than 10ms latency between appliances (replication interface)**
 - ▶ For good application performance may find far lower required.
 - ▶ 1 or 2 ms a good target – practical at 'metro area' distances
- **Sufficient bandwidth for all message data being transferred through HA queue managers (replication interface)**
- **No native VLAN tagging or link aggregation on these connections**
- ***Ideal world is physical cabling between systems***
 - ▶ *In a single datacentre, co-located rack scenario, avoids all infrastructure concerns*

DR in the MQ Appliance

(Since 8.0.0.4)

Setting up disaster recovery

- DR has different goals (asynchronous, manual) so slightly different externals to HA but similar process
- 1. Connect two appliances together (only one, 10GB, connection needed)
- 2. On 'main' appliance, convert queue manager to disaster recovery primary:
 - ▶ `crtdrprimary -m <name> -r <standby> -i <ip address> -p <port>`
- 3. On 'recovery' appliance simply paste the text provided by the above
 - ▶ `crtdrsecondary <some provided parameters>`
- Synchronization begins immediately ('status' command shows progress)



Disaster recovery: Physical connection



- Single 10GB Ethernet connection
- Eth20 interface

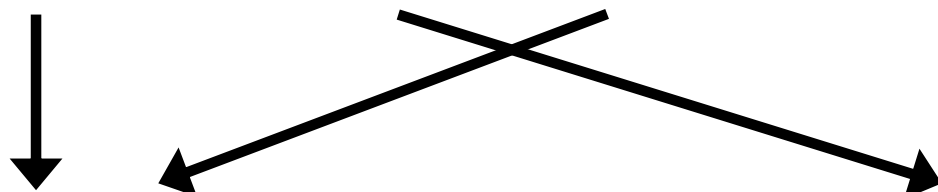


Disaster recovery: Flexible topologies

Production appliance



Production appliance



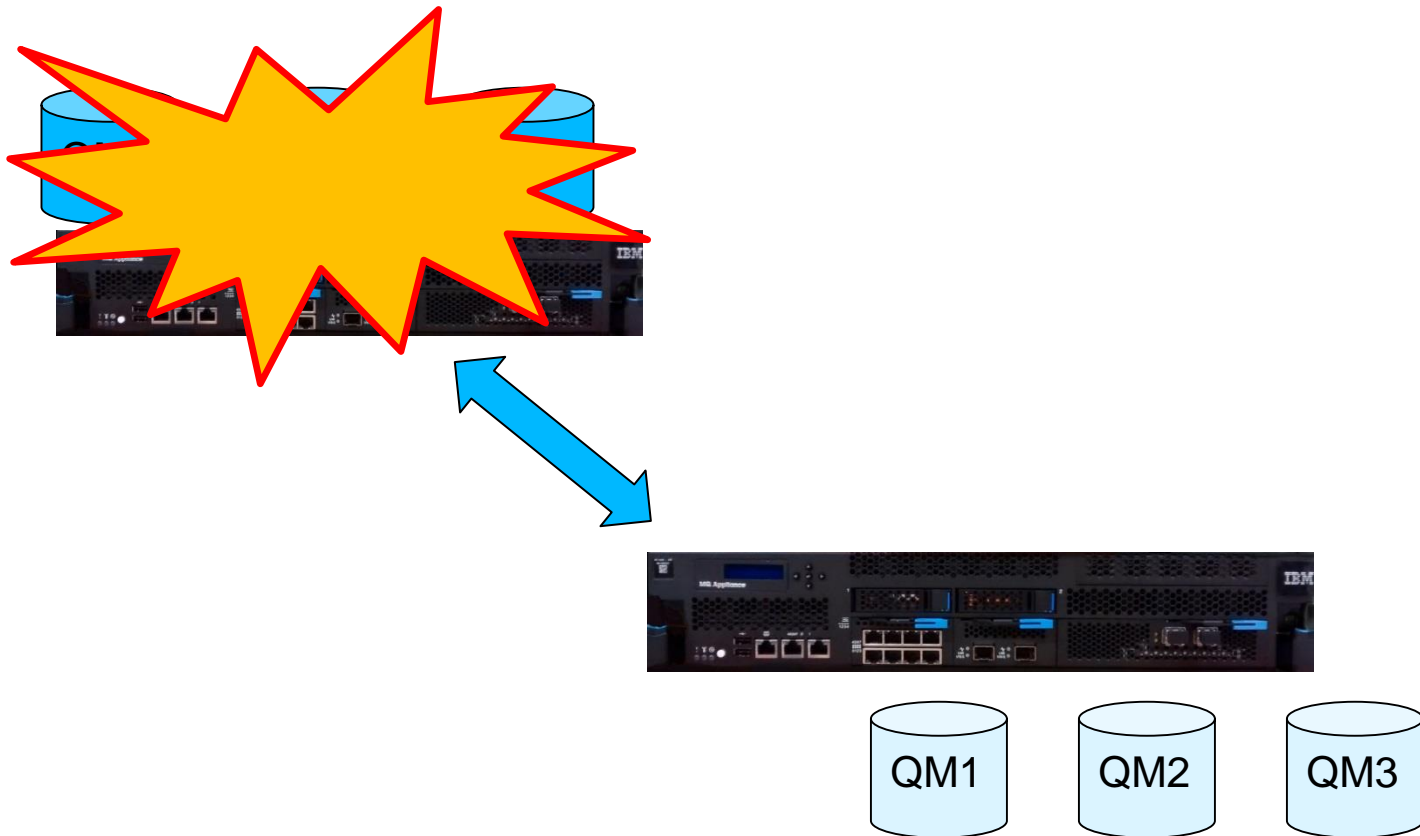
Offsite DR appliance



Mixed test/DR appliance

- **Because DR has no concept of a group, disaster recovery configurations can be even more flexible than HA**
 - ▶ though of course there is no automated management and failover
- **Each QM independently configures replication to a particular appliance.**
- **E.g. could configure single ‘DR’ site covering live appliances at multiple sites**

Disaster (large) strikes...



Recovering from disaster (1)

- On recovery appliance
- Check status

```
mqa (mqcli) # status mql
QM (mql)                               Status (Ended immediately)
DR role:                                Secondary
DR status:                               Remote appliance(s) unavailable
DR out of sync data:                     0KB
```

- Make queue manager primary

```
mqa (mqcli) # makedrprimary -m mql
The makedrprimary command succeeded.
mqa (mqcli) #
mqa (mqcli) #
mqa (mqcli) #
mqa (mqcli) # status mql
QM (mql)                               Status (Ended unexpectedly)
Queue manager file system:              134MB used, 3.0GB allocated [4%]
DR role:                                Primary
DR status:                               Remote appliance(s) unavailable
DR out of sync data:                     112KB
mqa (mqcli) #
```

Recovering from disaster (2)

- Start the queue manager

```
mqa(mqcli)# status mq1
QM(mq1)                               Status(Running)
CPU:                                    0.47%
Memory:                                 199MB
Queue manager file system:              134MB used, 3.0GB allocated [4%]
DR role:                                Primary
DR status:                               Remote appliance(s) unavailable
DR out of sync data:                    17212KB
```

- Clients can now reconnect, exactly the same as with HA case

Getting back to normal (1)

- Once the main site/appliance is available again you can switch back to normal running
- The exact way this is done depends on the state of the data on main and recovery appliances
- **Three possibilities**
 - ▶ Data is the same
 - ▶ Data is partitioned: data from recovery appliance is correct
 - ▶ Data is partitioned: data from main appliance is correct
- Use the 'status <qmname>' command to establish what the state is

Getting back to normal (2)

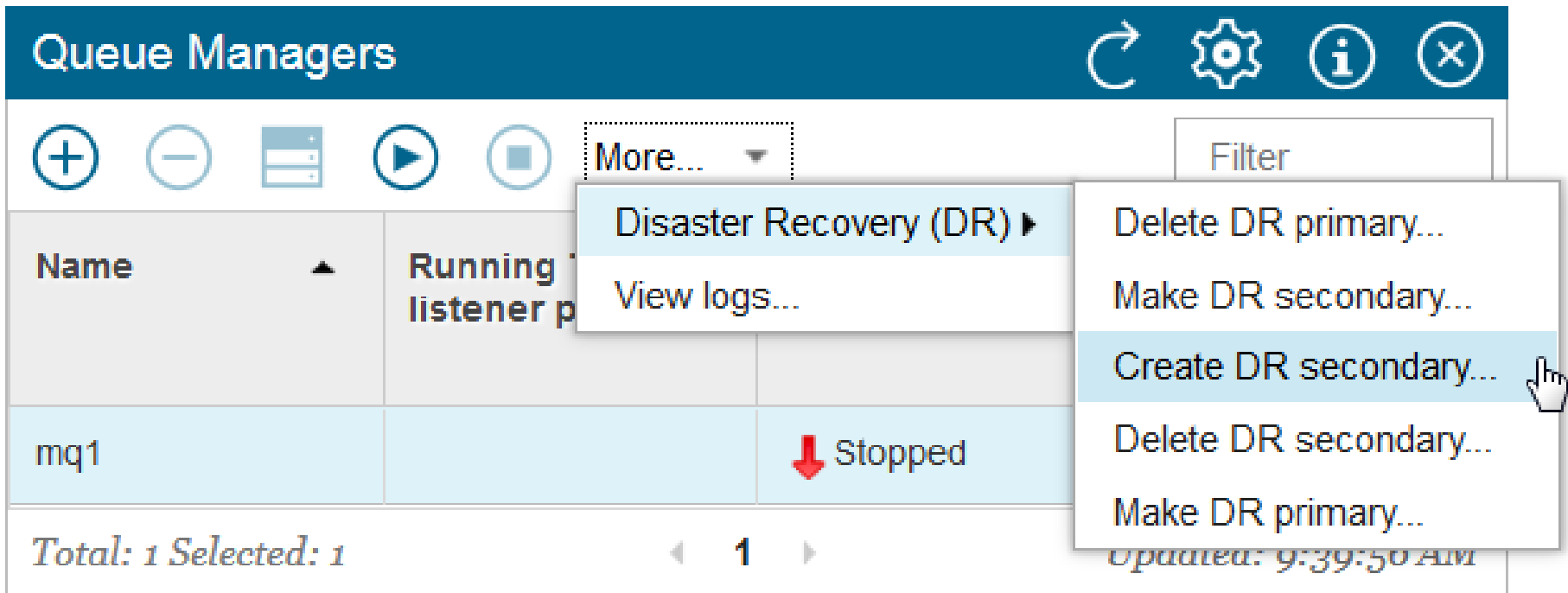
- Taking the case where you want to retain data from recovery appliance
- **Resolve the partitioned state**
 - ▶ On recovery appliance: **endmqm mq1**
 - ▶ On main appliance: **makedrsecondary -m mq1**
 - ▶ On recovery appliance: **makedrprimary -m mq1**

- **Synchronization occurs**

```
mqa(mqcli)# status mq1
QM(mq1)                               Status(Ended immediately)
DR role:                               Secondary
DR status:                             Synchronization in progress
DR synchronization progress:           8.3%
DR estimated synchronization time:     2016-09-09 06:01:23.751
```

- **Once synchronization is complete, move queue manager to main appliance**
 - ▶ On recovery appliance: **makedrsecondary -m mq1**
 - ▶ On main appliance: **makedrprimary -m mq1**
 - ▶ On main appliance: **strmqm mq1**

Managing DR from the MQ Console



The screenshot shows the 'Queue Managers' section of the MQ Console. A table lists the queue manager 'mq1' with a status of 'Stopped'. A context menu is open over the 'mq1' row, displaying options for Disaster Recovery (DR). The 'Create DR secondary...' option is highlighted by the mouse cursor. The table has columns for 'Name' and 'Running listener p...'. The status 'Stopped' is indicated by a red downward arrow. The bottom of the table shows 'Total: 1 Selected: 1' and a pagination control with the number '1'. A timestamp 'Updated: 9:39:50 AM' is visible at the bottom right of the interface.

Name	Running listener p...
mq1	↓ Stopped

Total: 1 Selected: 1

Updated: 9:39:50 AM

- Disaster Recovery (DR) ▶
 - Delete DR primary...
 - Make DR secondary...
 - Create DR secondary...**
 - Delete DR secondary...
 - Make DR primary...
- View logs...

- **NB: you need to have a queue manager selected to see this**
 - ▶ And it must be stopped (just like for commands)
- **Status is provided by bringing up the queue manager properties**

Replication, synchronization and snapshots (1)

- **There are two modes in which data can be sent from the primary instance to the secondary instance**
 1. Replication – when the two instances are connected, each individual write is replicated from the primary to the secondary in the order in which they are made on the primary
 - Just like when using HA
 2. Synchronization – when the connection is lost and then restored

- **Synchronization is used to get the secondary back in step as quickly as possible**
 - ▶ This means that the secondary is inconsistent until the synchronization completes and the queue manager would not be able to start

Replication, synchronization and snapshots (2)

- To resolve this issue, a 'snapshot' is taken of the queue manager whenever synchronization is started
- If connection lost again (or complete failure of the primary appliance) while synchronizing you can still issue `makedrprimary` command on standby to recover. However:
 - ▶ This will revert the queue manager to the state it was in at the beginning of synchronization
 - ▶ Can take a long time (hours for a large queue manager)
 - ▶ Updates made to the primary since original outage will be lost
- Space is reserved for this process whenever DR queue managers are configured
 - ▶ So may be surprised to see less disk available than you thought!

Appliance DR commands

Command	Description
<code>crtldrprimary</code>	Enables an existing queue manager for DR
<code>crtldrsecondary</code>	Creates a secondary version of a queue manager on a recovery appliance for DR purposes
<code>makedrprimary</code>	Switches a queue manager on an appliance to have the primary role in a DR configuration
<code>makedrsecondary</code>	Prevents a queue manager in a DR configuration from starting, and marks it as the secondary
<code>dltdrprimary</code>	Removes DR configuration from a queue manager that had the primary role in a DR configuration leaving it as either a stand-alone or HA queue manager
<code>dltdrsecondary</code>	Deletes a queue manager that had the secondary role in a DR configuration. Both <code>dltdr*</code> commands need to be run to fully remove a DR configuration

DR – requirements / restrictions

- The maximum latency for the replication link is 100 ms
- NB: there is no requirement for eth20 interfaces to be in same subset for DR only. The KC is wrong
- Native VLAN (trunked) and link aggregation are not supported on the replication interface

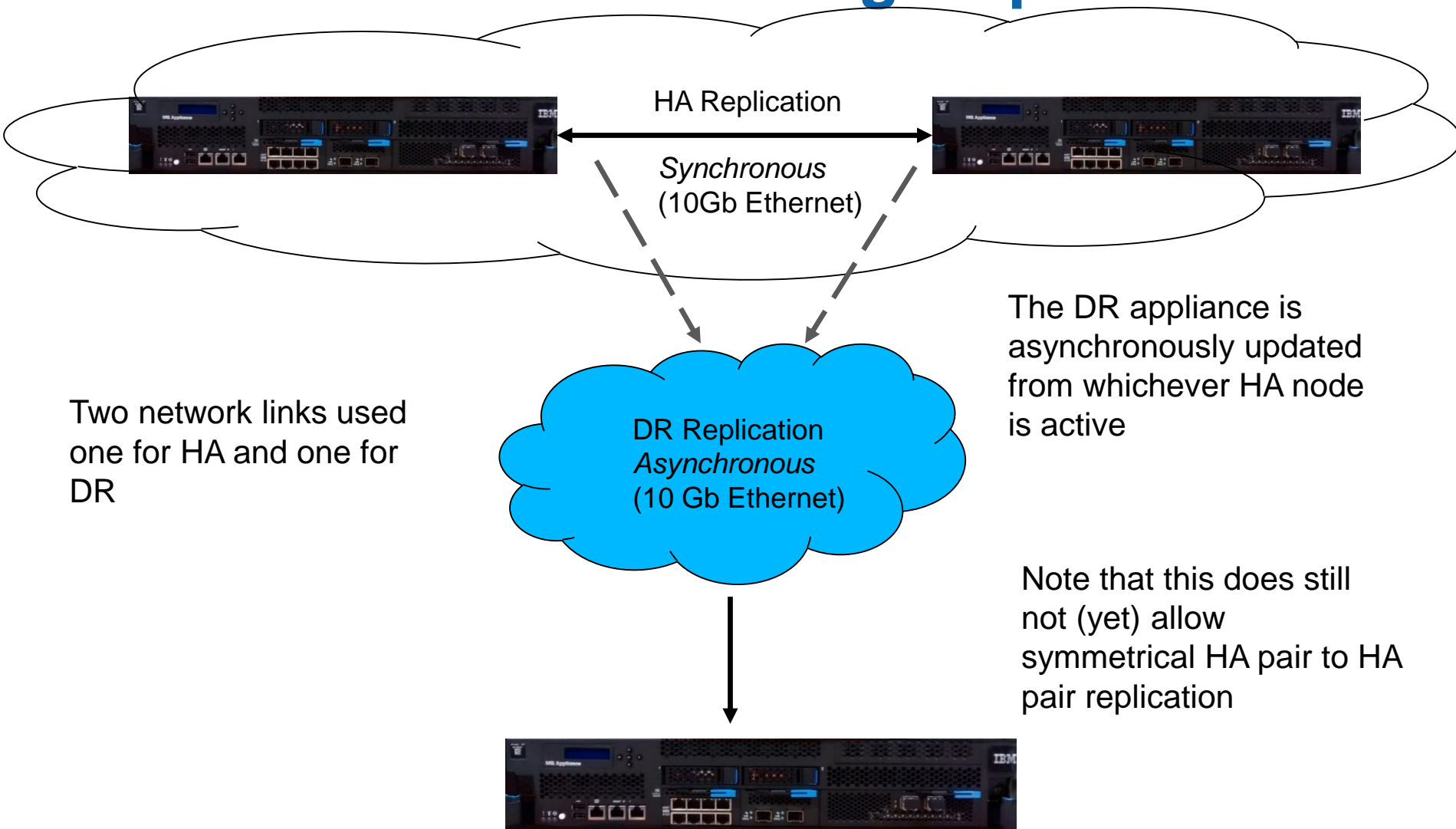
Combining HA and DR

8.0.0.5

What was available in 8.0.0.4?

- **You could have either**
 - ▶ HA between a pair of appliances
 - ▶ DR between two or more appliances
- **But not HA and DR at the same time**
 - ▶ Even between different queue managers
- **Furthermore at 8.0.0.4 the only available hardware (M2000) came with two 10GB Ethernet ports**
 - ▶ Both of which would be required if HA and DR were available
 - ▶ Leaving only the 6 * 1GB ports for messaging traffic
 - Link aggregation could be used to improve bandwidth/availability

And in 8.0.0.5: DR for HA groups



Two network links used
one for HA and one for
DR

The DR appliance is
asynchronously updated
from whichever HA node
is active

Note that this does still
not (yet) allow
symmetrical HA pair to HA
pair replication

And if you have the M2001 hardware update...

- You get 4 * 10GB ports
 - ▶ Which allows two to be used for application workloads



Setting up DR for HA groups

- **Appliance and queue managers must already be set up for HA**
 - ▶ Same commands as shown earlier
- **Stop HA queue manager on the appliance that it is running on**
- **On the same appliance make the queue manager the DR primary:**
 - ▶ `crtdrprimary -m <name> -r <standby> -i <ip address> -p <port> -f <floatingIP>`
- ***On recovery appliance paste the command provided from crtdrprimary command***
 - ▶ `crtdrsecondary <some provided parameters>`
- **Note the new floating IP address (-f flag)**
 - ▶ *This allows the recovery appliance to replicate data from the queue manager regardless of which appliance in the HA pair the queue manager is running on*

DR for HA groups – requirements / restrictions

- As per separate HA/DR, plus
- The floating IP address must be in the same subnet as the static DR replication interface on the HA appliances (eth20)
- If removing an appliance from an HA group you need to remove the DR configuration for all queue managers that are part of this group first
 - ▶ I.e issue 'dltdrprimary' before issuing 'sethagr -e'
 - ▶ Suspending a queue manager from a group 'sethagr -s' is fine

Communication considerations

Channel reconnection

- **The same approach is used regardless of the HA/DR combination being used**
 - ▶ Appliance HA/DR looks externally just like the multi-instance queue manager function that was added in MQ 7.0.1
- **Client applications, and other queue managers, reconnect to the secondary instance after failure by configuring multiple IP addresses for the channels**
 - ▶ Either explicitly in CONNAME (comma separated list)
 - ▶ Or by defining a CCDT with multiple endpoints
 - ▶ Or using a preconnect exit
- **Don't forget that cluster receivers define their own 'multiple homes'**

Client reconnect implications

- Again, the same considerations as with multi-instance queue managers
- When a queue manager ‘fails over’ using HA or DR, effectively from the point of view of an application or remote queue manager, this queue manager has been restarted.
 - ▶ Ordinarily, application would receive MQRC_CONNECTION_BROKEN
- This can typically be hidden from applications using 7.0.1 or higher client libraries by using ‘client auto-reconnect’ feature, but there are some limitations to be aware of
 - ▶ Failure during initial connect will still result in ‘MQCONN’ failing and the application having to retry
 - ▶ Browse cursors are reset
 - ▶ In process units of work are rolled back
 - ▶ XA transactions are not supported
- This can allow existing applications to exploit HA with no change, or minimal change, but consult documentation

Security

- **Most security data - for example certificate stores and authority records - is replicated alongside queue manager in HA or DR configuration**
- **However: users and groups are NOT replicated between appliances**
 - ▶ Because not all configuration has to be identical – you may have queue managers on either device and associated users which you do not wish replicated
 - ▶ **So... strongly consider LDAP for messaging users on replicated queue managers**
- **Group configuration/heartbeat/management is ‘secure by default’**
 - ▶ Prevents e.g. another device configuring itself replication target
- **But currently no encryption on replication link – possibly acceptable for single data centre HA, needs careful consideration for DR**
 - ▶ If necessary, using AMS will ensure all message data encrypted both at rest and across replication

Performance

HA performance – no latency

10 Application Request Responder (2KB Persistent)

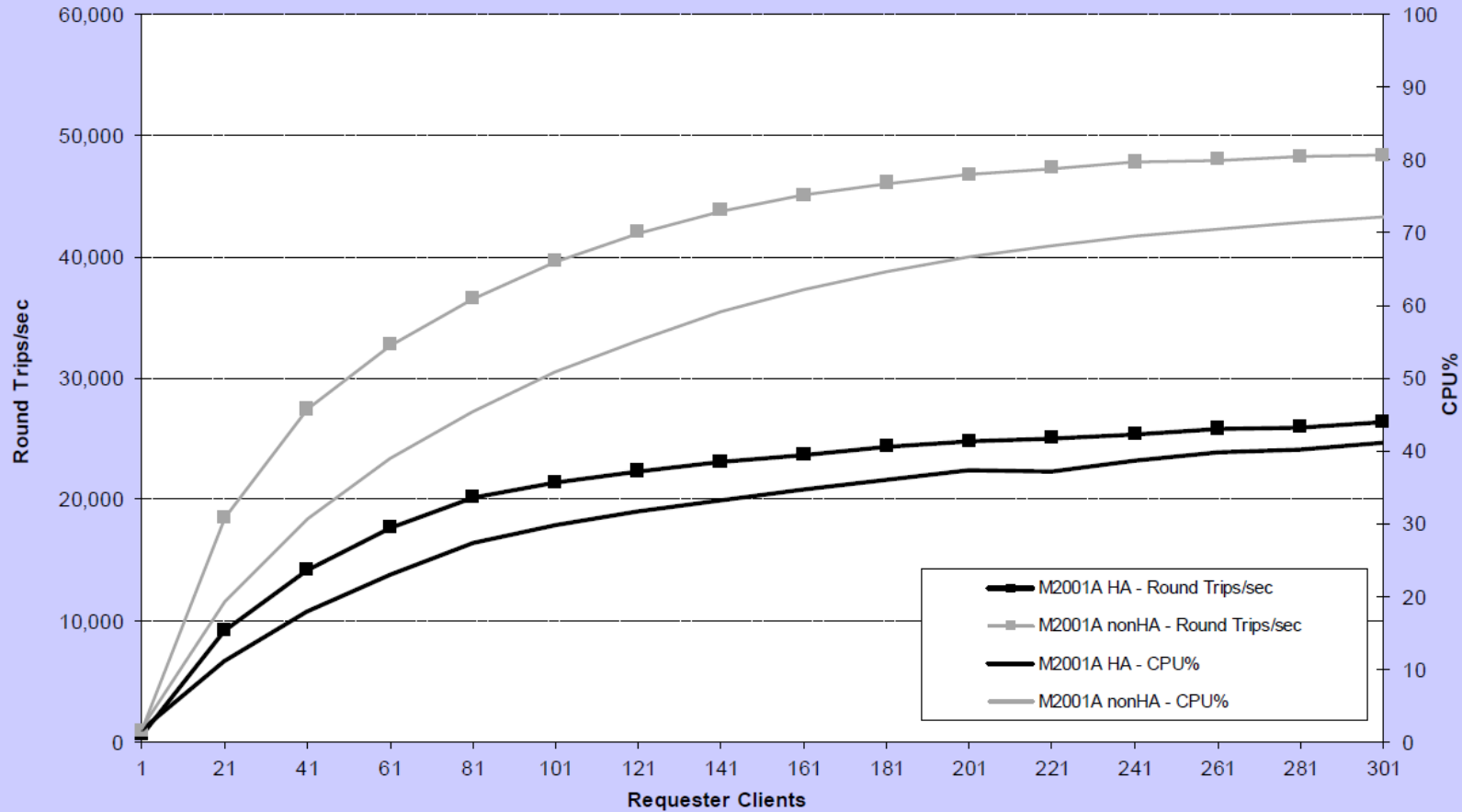


FIGURE 2 – PERFORMANCE RESULTS FOR 2KB PERSISTENT MESSAGING

HA performance – 2ms latency

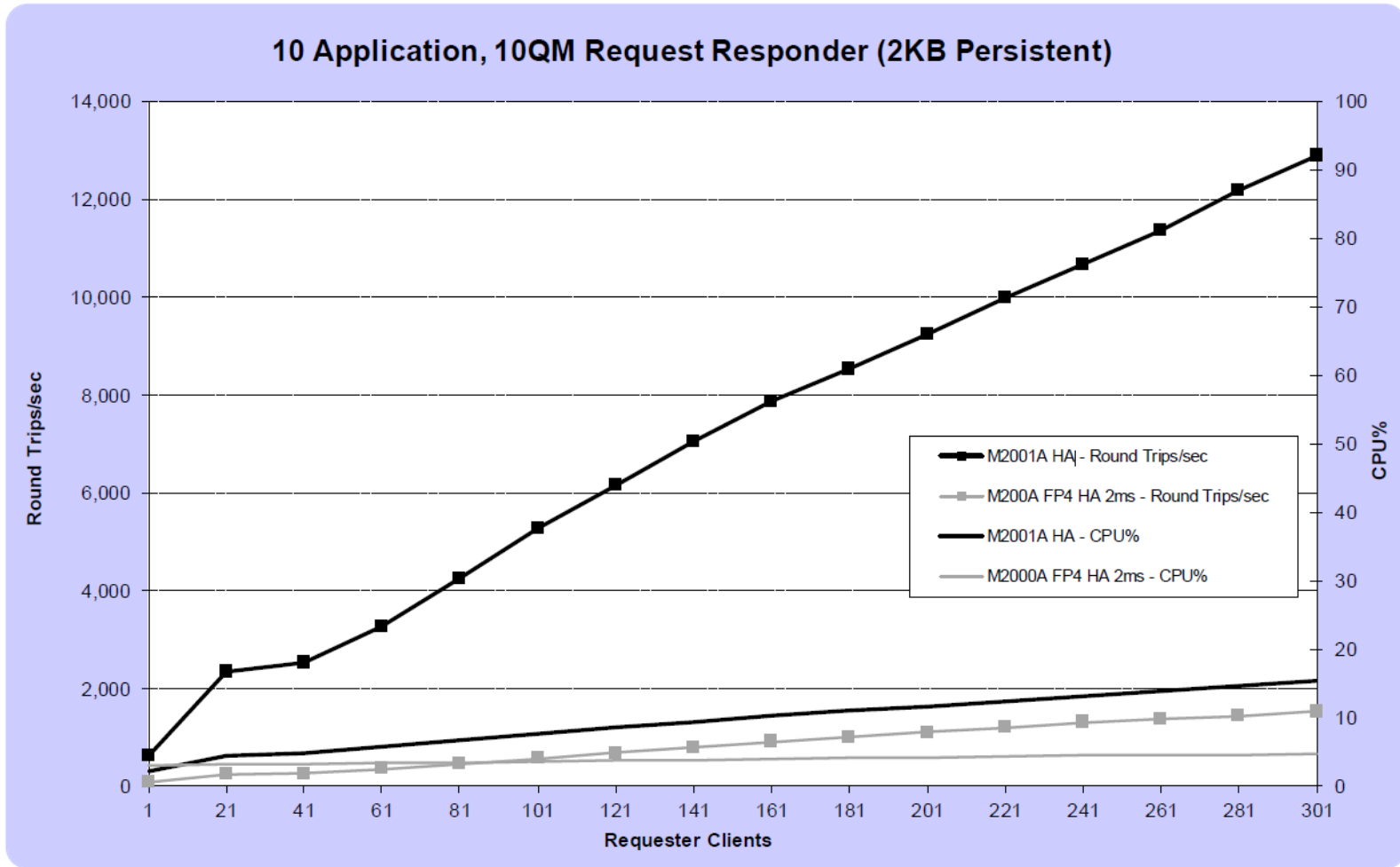


FIGURE 4 - PERFORMANCE RESULTS FOR 2KB, 10QM PERSISTENT MESSAGING WITH/WITHOUT 2MS LATENCY

DR performance

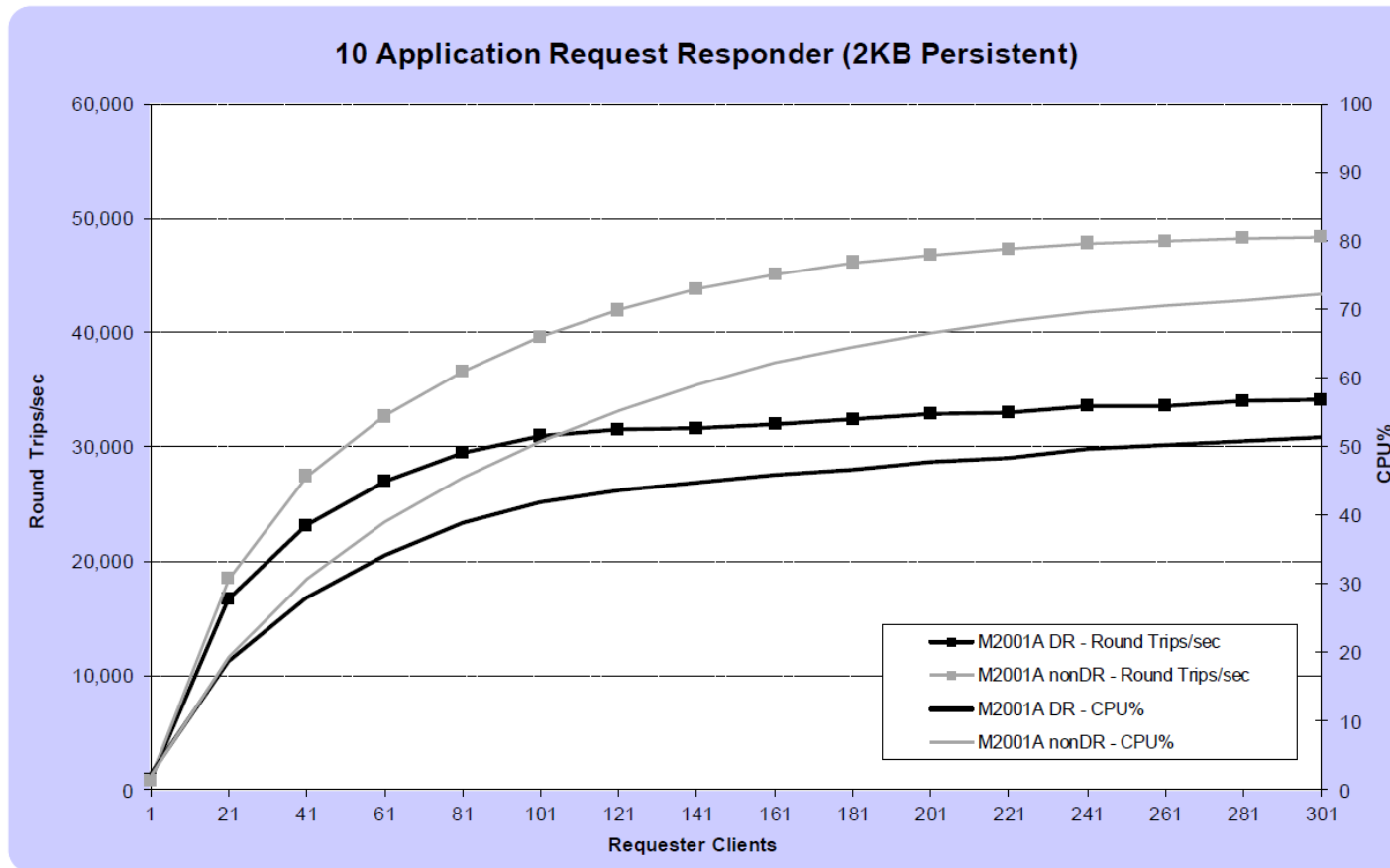


FIGURE 5 – PERFORMANCE RESULTS FOR 2KB PERSISTENT MESSAGING

- **NB: if running over a link with 50ms latency DR is within 90% of DR numbers above – so much more tolerant of latency**

Performance notes

- **More information available here:**
 - ▶ <ftp://public.dhe.ibm.com/software/integration/support/supportpacs/individual/mpa2-2.0.pdf>
- **Note that these graphs are for M2001 hardware**
- **M2000 information is available at**
 - ▶ <ftp://public.dhe.ibm.com/software/integration/support/supportpacs/individual/mpa2.pdf>

Summary

- Introduction
- HA in the Appliance
- DR in the Appliance
- Combining HA and DR
- Communication considerations
- Performance

Questions & Answers

