Performance overhead of NHA:CRR

Objective

In MQ V9.4.2, IBM released the capability of replicating between a live group of NativeHA (NHA) QM and a recovery group. This new functionality is called Native HA Cross-Region Replication (NHA:CRR). This capability removes the need to rely on any storage-level data replication to be performed for HA and DR, which can often negatively impact the performance capability of your MQ system.

The MQ Performance team have previously published documents that illustrate NHA performance in the OpenShift environment. This document serves to illustrate the impact of enabling the additional asynchronous replication to the recovery group.

The recovery replication link being asynchronous, means that the live group does not wait for confirmation of receipt of the replicated updates to the recovery group before proceeding. Therefore, the most recent updates to the live group are not guaranteed to have been successfully replicated when a failover event is encountered, resulting in the loss of the most recent data. Data is typically both the addition of new messages, but also the removal of messages from queues and subscriptions.

The amount of data at risk will be governed by the messaging rate, message size, latency and bandwidth to DR site, connection resiliency, PV sizing and log configuration. This document will advise on what monitoring capabilities you can use to determine the current state of your messaging system.

Environment

OpenShift: 4.16.23
MQ Operator: 3.5
MQ: V9.4.2 / 9.4.2.0-r1
Test client: cphtestp (containerized MQI test application)

Recovery group hosted on xLinux hardware in the same datacentre as OCP cluster. 40ms latency will be injected between the live and recovery groups to replicate a more realistic DR environment.

Test configuration and results

MQI C based interfaces will be used by our containerized performance test harness cphtestp.

- Persistent
- 2K / 20K messages size
- TLS12 enabled for Live HA Group, Recovery HA Group and Live->HA replication
- 50GB PV allocated to Live group instances
- 100GB PV allocated to Recovery group instances
- Storage allocated on remote SAN infrastructure
- IMGLOGLN set to 12500MB (25% of PV)
- LZ4 Compression enabled, CompressionThreshold=64 within NHA groups and between the live and recovery groups
- OCP(Live) Network: 1Gb

- OCP(Live)->DR(Recovery) Network: 10Gb, 40 ms injected latency
- DR(Recovery) Network: 100Gb
- Core allocation to QM pod resources: 8 cores

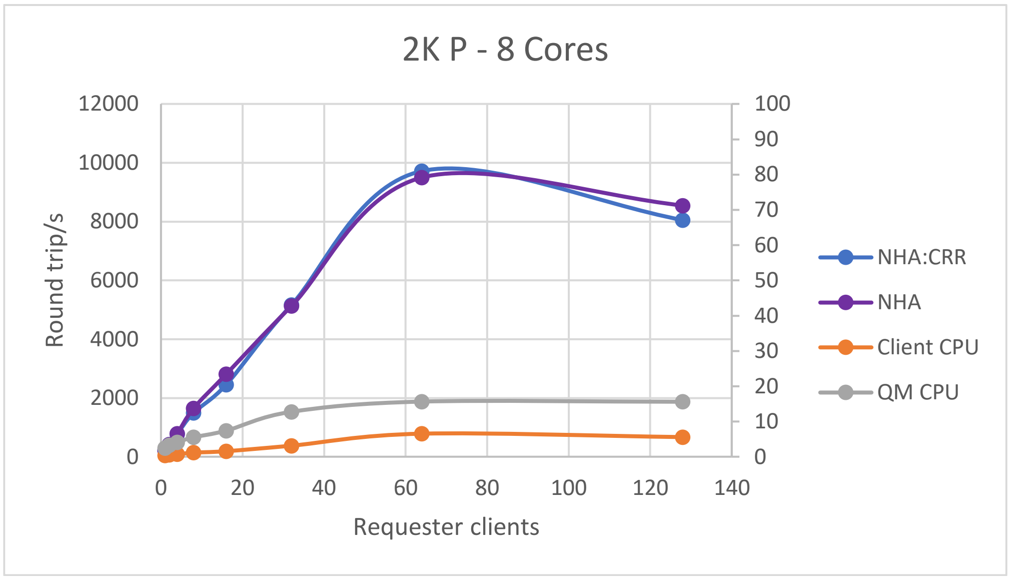

Figure 1 - 2K Persistent messaging

In the chart above for 2K Persistent messaging, the NHA and NHA:CRR performance is virtually identical, thus showing very little overhead of the asynchronous DR replication.

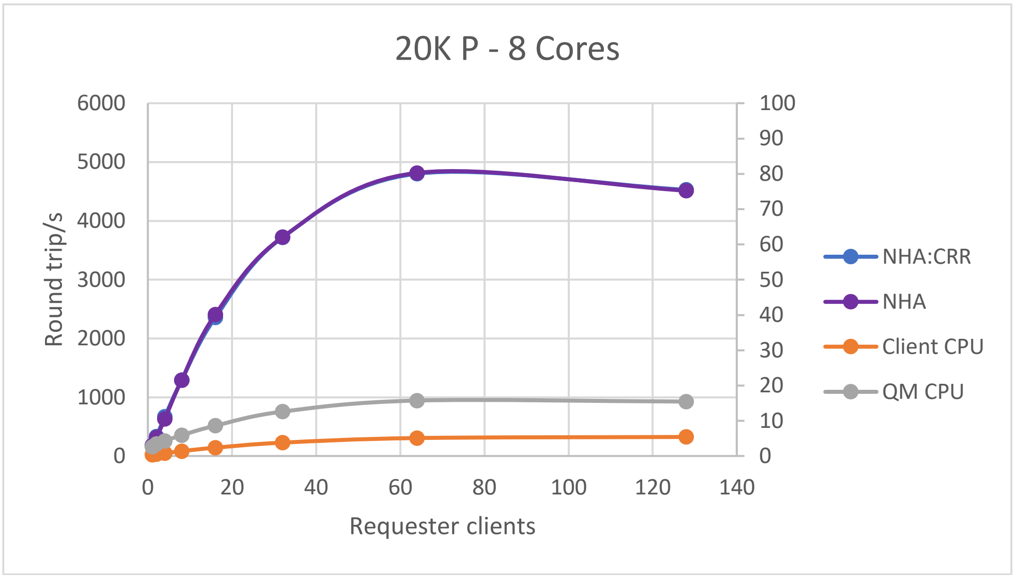

Figure 2 - 20K Persistent messaging

The 20K test again performs identically whether you have CRR enabled to the DR Recovery group or not, even with 40ms of injected latency. Hence, the blue NHA:CRR line is obscured by the purple NHA line above.

This shows that enabling NHA:CRR has very little impact when the replication infrastructure is able to support the messaging volume, and we have gained a full set of consistent data at our recovery site, albeit a little behind our live site. If the replication infrastructure cannot keep pace with the messaging workload at the live group, the recovery group may eventually require a rebase (i.e. a new recovery log to be sent to the recovery group). The next section will show you how to monitor your messaging deployment to ensure that your infrastructure can match your messaging DR requirements. i.e. how to monitor your Recovery Point Actual (RPA) against your Recovery Point Objective (RPO). RPA being the amount of data that you would lose compared with your maximum amount of data stated by your objective (RPO).

Monitor NHA:CRR

Users of NHA:CRR capabilities will want to know how much data they might lose should a failover event occur. In this section we will discuss techniques for monitoring the current RPA.

New statistics have been added to the MQ System topics that help with this task:

```
sh-5.1$ /opt/mqm/samp/bin/amqsrua -c NHAREPLICA -t RECOVERY -o +
Publication received PutDate:20250226 PutTime:15535589 Interval:10.000 seconds
paris                                        Log bytes sent 620335104 62033510/sec
paris                                        Backlog bytes 462823424
paris                                        Backlog average bytes 211347208
paris                                        Average network round trip time 40777 uSec
paris                                        Compressed log bytes sent 15062664
paris                                        Log data average compression time 11 uSec
paris                                        Log bytes decompressed 639122908
paris                                        Log data average decompression time 32 uSec
paris                                        Rebase count 13
paris                                        Recovery log sequence number <0:1157:41026:59250>
0x00000485a042e772
```

These statistics are collated at the live QM and can either be collected by using Prometheus or the supplied samples for monitoring the system topics (amqsrua). For initial performance testing of a new environment, we would recommend cphtestp (containerized MQI test harness) which will also collect and summarize the relevant values for you.

In the sample above you can see that the recovery group is currently 463MB behind the live group (Backlog bytes). The live group is replicating ~62MB/sec to the recovery group with the average latency measured at just above 40ms. The backlog value tends towards a pessimistic value due the fact that status updates are only sent every 10s from the recovery group to the live group.

An ever-increasing number of backlog bytes would suggest that the replication infrastructure cannot keep up with the messaging workload at the live group and a rebase may soon be required. Messages to the QM error file will report on rebase activity.

If you want more frequent status updates supplied to the live group, these can be achieved using the following qm.ini settings (set on the Live group QM config, but should be set on both in case the roles are ever reversed):

```
    NativeHARecoveryGroup:
      StatusInterval=10000
      StatusCommitThreshold=0
```

The StatusInterval units are milliseconds, and the default is 10s but can be lowered to a minimum of 5 seconds (5000). The StatusCommitThreshold defaults to 0 (i.e. just apply StatusInterval) but can be set to any value; this value is measured in bytes and determines how many committed bytes at the recovery group are required before sending the recovery log status to the live group.

In the 20K test run earlier, the maximum backlog bytes reported in the default configuration was ~1200MB. After setting the StatusCommitThreshold to 1MB (1048576), the maximum reported backlog bytes of the recovery group was 14MB.

The queue manager is replicating the recovery logs from the live site and replaying them in the recovery site. The further behind the recovery site is, due to a replication backlog, the greater the RPA in the event of a failover. Additionally, to minimize the replication overhead of a log rebase, it is optimal for the live site to maintain enough recovery log data for the recovery site to replay. Therefore, for an increased backlog in replicated data, you may require larger amounts of storage available to both live and recovery groups to ensure that this is possible. If the recovery group becomes too far behind the live group, and the live group has already reused some of its recovery log, a log rebase will be required to update the recovery group to the current state. In this instance, the backlog bytes may not be reduced until the rebase is complete. Whilst a rebase is taking place, the RPA will likely be increasing further as the live and recovery groups conciliate their data.

Conclusion

The new capabilities of NHA:CRR provide further availability to your messaging infrastructure by supporting asynchronous replication to a disaster recovery (DR) site remote from the location of your Live HA group (already providing High Availability).

The asynchronous nature of this replication provides little impact to the performance of your existing HA live group. You will need to ensure that you have sufficient bandwidth and resources at your recovery site to ensure that it you can meet your Recovery Point Objectives. Monitor your system during deployment testing to ensure your requirements are met. You should also continue to monitor your production systems so that you are aware of any outages and impacts that may result in failing to meet your RPO.

If the recovery group cannot be kept consistent using the replicated logs, it may be required to rebase the recovery group from the live group. A set of consistent data is always maintained at the recovery group, so that in case of failure of the primary site during a rebase, there will always be a consistent set of logs at the recovery site from which could be used to start processing workload at the recovery site (which now forms the live group).

MQ can now offer a native solution to your HA and DR requirements, simplifying deployment and reducing cost whilst maintaining a performant solution for your messaging requirements.

Links

cphtestp
https://github.com/ibm-messaging/cphtestp

Native HA Performance on OpenShift
https://github.com/ibm-messaging/mqperf/blob/gh-pages/openshift/OCP4.12-NativeHA.pdf

NHA:CRR Documentation
https://www.ibm.com/docs/en/ibm-mq/9.4?topic=containers-native-ha-cross-region-replication

NHA:CRR Advanced tuning
https://www.ibm.com/docs/en/ibm-mq/9.4?topic=operator-advanced-tuning-native-ha-crr

Monitoring system resource using amqsrua
https://www.ibm.com/docs/en/ibm-mq/9.4?topic=stmat-monitoring-system-resource-usage-by-using-amqsrua-command

Metrics published on the system topics
https://www.ibm.com/docs/en/ibm-mq/9.4?topic=trace-metrics-published-system-topics

Appendix A

Hardware specification for Worker Nodes and Clients :

System	ThinkSystem SR630
OS	Worker nodes use CoreOS, Clients use RHEL 9.4
CPU	2x16 Core 2.8Ghz Xeon Gold 6242
RAM	96GB RAM RDIMM TruDDR4 2933MHz
RAID	930-16i 4GB Flash PCI 12Gb RAID Adapter
Disks	800GB SSD (2x400GB in RAID-0 array) SS530 Performance SAS 12Gbp/s
SAN Connectivity	Dual Port HBA 16Gb
10GbE Network	Dual Port 10GbE Broadcom Network Adapter
100GbE Network	Dual Port 100GbE Mellanox ConnectX-4 Network Adapter

https://lenovopress.com/lp1049-thinksystem-sr630-server-xeon-sp-gen2