MQ High Availability and Disaster Recovery Implementation scenarios

Sandeep Chellingi
Head of Hybrid Cloud Integration
Prolifics
Agenda

- **MQ Availability**
  - Message Availability
  - Service Availability
  - HA vs DR

- **High Availability Scenarios**
  - HA Clusters
  - Multi-Instance Queue Manager
  - MQ Appliances
  - Replicated Data Queue Manager (RDQM)
  - MQ Containers

- **Disaster Recovery Scenarios**
  - MQ Appliances
  - Replication Data Queue Manager (RDQM)
  - MQ Containers

- **Comparison**
Message Vs Service Availability

- **Message Availability**
  - Messages stored on exactly one queue manager
  - To achieve message availability you need to recover messages as quickly as possible during outages
  - Multiple Message recovery options
    - High Availability using HA Clusters, Multi-Instance, Replicated Data queue Managers, MQ Appliances
    - MQ Containerization (Dockers, Kubernetes, public cloud container services)

- **Service Availability**
  - Availability of service, ensure queues are available
  - MQ clustering of queues across multiple queue managers
  - Horizontal scaling to improve service availability
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… 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.

- Detecting issues as they occur and switching between redundant components.

- Ideally ensuring that the switching technology itself is 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 nodes, and 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?
HA vs DR

- 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
**HA Clusters**

- HA Cluster examples Power HA on AIX, Veritas cluster, HP service guard and Red Hat cluster suite

- **HA clusters include the following features:**
  - Coordinates with multiple resources, such as an application server or database
  - Configuration options can include clusters comprising of more than two nodes
  - Seamless IP address switch between nodes during failover
  - Standby, takeover, One-Side takeover and mutual Takeover

- **Limitations of HA clusters:**
  - Additional product purchase, specific disk requirements and skills are required
  - Configuration of HA clusters is relatively complex
HA MQ Multi Instance Queue Manager

- MQ Multi-Instance Queue Manager
  - Active – Standby pair, MQ Manages Failover
  - Shared Network Storage managed by different sub System
  - Specific NFS storage requirements
  - Supported on Cloud
Why IBM MQ Appliance?

- **The scalability and security of IBM MQ Appliance**
  - Integrates seamlessly into MQ networks and clusters.
  - Familiar administration model for administrators with MQ & Datapower skills.

- **Fixed hardware specification allows IBM to tune the firmware**
  - Having fewer PVUs makes it easier to deploy and manage
  - Less performance tuning should be needed

- **Simplified ownership**
  - Self-contained: avoids dependencies on other resources/teams
  - Licensing: Simpler than calculating licensing costs (e.g. by PVU)
  - Security: Easier to assess for security compliance audit
MQ Appliance Vs Traditional MQ Server

- IBM MQ Appliance
  - Prebuilt for hub pattern – no apps on device
  - No additional software installation
    - No user exits in MQ
    - Monitoring agents must be remote
    - No malware or backdoors
  - High availability out-of-the-box
  - Pre-tuned for optimal performance
  - Single firmware update for whole appliance (rollback as single unit)

- IBM MQ on Traditional server
  - Do It Yourself hub or generic server – apps + middleware
  - Install any software
    - Build & maintain custom extensions
    - Can add local monitoring agents
  - HA cluster SW or network storage for HA
  - Custom tuning for OS and middleware
  - Discrete maintenance (OS, MQ, etc.)
HA IBM MQ Appliance 2002

- Replication-based HA IBM MQ Appliance
  - Automatic failover, plus manual failover for migration or maintenance
  - Independent failover for queue managers so both appliances can run workload
  - Optional IP address associated with an HA queue manager, automatically adopted by the active HA appliance—single logical endpoint for client apps
  - No persistent data loss on failure
  - No external storage, additional skills required
HA Floating IP address

- Optional IP address associated with an HA queue manager
- IP address automatically adopted by the active HA appliance
- Single logical end-point per queue manager for client applications
- No need for comma-separated list of IP addresses, CCDTs, or other routing
- Exploit aggregate interfaces for enhanced network availability
Disaster recovery for HA groups

- Support for both HA and DR
- DR appliance asynchronously updated from whichever HA appliance is active
- DR configured independently for each queue manager
  - One HA partner per appliance
  - One DR recovery appliance per queue manager
HA-DR MQ Appliances
Upgrading MQ appliances

- Appliance updates supplied as a simple single file; signed and secure.
  - Nothing else can be installed
- All system and MQ updates provided in one consumable package
- Rolling updates for HA and DR
- To install maintenance:
  - Download updates from Fix Central
  - Copy firmware image to the appliance
  - Initiate update and reboot
HA RDQM – IBM MQ Advanced

- Replication-based HA IBM MQ Advanced
- Linux only, MQ Advanced HA solution with no need for a shared file system or HA cluster
- Shared Nothing” approach, MQ manages failover
- Local block storage, synchronously replicated by MQ
- MQ configures the underlying resources to make setup and operations natural to an MQ user
- Three-way replication for quorum support
- Synchronous data replication for once and only transactional delivery of messages
- Active/passive queue managers with automatic takeover
HA RDQM – IBM MQ Advanced

- Active/passive queue managers with automatic takeover
- Per queue manager control to support active/active utilization of nodes
- Per queue manager IP address to provide simple application setup
- Supported on RHEL v7 x86-64 only
DR RDQM – IBM MQ Advanced

- Data is replicated between Primary DR queue manager and Recovery DR queue Manager Nodes
- Replication of Data between two nodes is managed by DRBD
- Cannot Add existing queue manager in Disaster Recovery RDMQ
- A queue manager cannot be part of both HA RDQM and DR RDQM
- Primary Disaster Recovery Queue managers
Cloud

IaaS (Infrastructure-as-a-Service - VMs)
• Are good for large services/apps, but generally not ideal
• May be used more like physical machines, but with added flexibility

CaaS (Containers-as-a-Service - e.g. Kubernetes)
• Are good for micro-services/apps
• Potentially quite short-lived

PaaS (Platform-as-a-Service - e.g. Bluemix, Cloud Foundry)
• Are great for application code in general
• Handing off infrastructure worries to someone else

FaaS (Functions-as-a-Service - e.g. OpenWhisk, AWS Lambda)
• Could be used for occasional compute loads
• Will likely drive lots of short-lived connections, so may not perform well for some messaging workloads
• Most support JavaScript (could use the MQ Light API), but some can support Java, C# and more
Persistent Storage

Reliability of storage
• Replicated across failure domains / availability zones?
• Are disk writes cached?
• What’s the failure rate of disks?

Connecting to the right persistent storage
• When a queue manager’s compute resource is moved (ex: run a container in a different VM), then something needs to connect the queue manager to the correct storage.
• For Example, the correct block storage volume, or directory on networked file storage.

Identifying the right persistent storage
• A very basic cloud orchestration setup could result in multiple instances of “QM1”
Persistent Storage Considerations

Region: US West 1
us-west-1 (N. Calif)

- Availability Zone
  - us-west-1a
  - us-west-1b
  - us-west-1c

Region: US West 2
us-west-2 (Oregon)

- Availability Zone
  - us-west-2a
  - us-west-2b
  - us-west-2c
Persistent Storage Considerations – Local SSD’s

Region: US West 1
us-west-1 (N. Calif)

us-west-1a
server1
qm1
data

us-west-1b
server2
qm2
data

us-west-1c
server3
qm3
data

Region: US West 2
us-west-2 (Oregon)

us-west-2a

us-west-2b

us-west-2c
Persistent Storage Considerations – Local SSD’s

Region: US West 1
us-west-1 (N. Calif)

- us-west-1a
- us-west-1b
- us-west-1c

Region: US West 2
us-west-2 (Oregon)

- us-west-2a
- us-west-2b
- us-west-2c
Persistent Storage Considerations – Elastic block Storage
Persistent Storage Considerations – Elastic block Storage

- MQ multi-instance/cloud provider auto-restart/custom control e.g. Pacemaker
Persistent Storage Considerations – Elastic block Storage

Region: US West 1
us-west-1 (N. Calif)

- us-west-1a

Region: US West 2
us-west-2 (Oregon)

- us-west-2a
- us-west-2b
- us-west-2c

- MQ multi-instance/cloud provider auto-restart/custom control e.g. Pacemaker
Persistent Storage Considerations – Elastic block Storage

Region: US West 1
us-west-1 (N. Calif)

- MQ multi-instance/cloud provider auto-restart/custom control e.g. Pacemaker

Region: US West 2
us-west-2 (Oregon)
Persistent Storage Considerations – Elastic File Storage

Region: US West 1
us-west-1 (N. Calif)

us-west-1a
  server1
  qm1
  data

EFS

us-west-1b
  server1
  qm1'
  data

us-west-1c
  server1
  qm1'
  data

Region: US West 2
us-west-2 (Oregon)

us-west-2a

us-west-2b

us-west-2c
Persistent Storage Considerations – Elastic File Storage

Region: US West 1
us-west-1 (N. Calif)

- us-west-1a
- us-west-1b
- us-west-1c

EFS

Region: US West 2
us-west-2 (Oregon)

- us-west-2a
- us-west-2b
- us-west-2c

- Cloud provider auto-restart/custom control e.g. Pacemaker (not MQ multi-instance)
Persistent Storage Considerations – Elastic File Storage

Region: US West 1
us-west-1 (N. Calif)

Region: US West 2
us-west-2 (Oregon)
Persistent Storage Considerations – Elastic Block Storage

- DR rather than HA - asynchronous replication so some messages at risk of loss
Persistent Storage Considerations – Elastic Block Storage
Containers

- Containers provide a similar environment to a VM but lighter in weight
  - A virtual machine provides an abstraction of the physical hardware
  - A container abstracts the OS level, typically at the user level
- Linux containers
  - Containers all share the same OS kernel
  - Images are constructed from layered filesystems
  - Containers isolate applications from each other and the underlying infrastructure
MQ Containers HA

- Containers provide a similar environment to a VM but lighter in weight
- A virtual machine provides an abstraction of the physical hardware
- A container abstracts the OS level, typically at the user level

- Linux containers
- Containers all share the same OS kernel
- Images are constructed from layered filesystems
- Containers isolate applications from each other and the underlying infrastructure
Kubernetes Containers

- Docker
- IBM Containers
- Google Container Engine
- Azure Container Service
- Resource & Environment Management
- Installation of MQ
- Starting & Creating QMGRs
Questions & Answers