WebSphere MQ Clustering New Features in MQ V7.1 / V7.5 -- Distributed

Jane Li (lihui@cn.ibm.com)
CDL MQ L2 support

23 Jan,2014

Agenda

- WMQ 7.1 enhancements
  - Performance improvement
  - Distributed error logs increased
  - Enhanced security model
  - Workload balancing by message group
  - Pub sub cluster controls
  - Cluster Monitoring and Rebalancing
- WMQ 7.5 enhancements
  - Split Transmit Queue

WMQV7.1 Clustering new feature

Performance improvement

- Fastbound amqrrmfa process
  - Simplifies access to queue manager data
  - Faster MQI calls
- Cluster repository cache maintenance and garbage collection

Repository manager error handling and recovery

- **V7.0 before behaviour:**

Really unexpected errors immediately terminate the repository manager (e.g. connection broken, rollback failure, etc,).

If command fails, report in error log, sleep for 10 minutes then rebuild the cache and **try once more, then terminate repository manager**

If command is processed correctly but commit is rolled back (resource problem) then **sleep 5 seconds and retry indefinitely**

If command queue disabled, sleep for 10 minutes then retry indefinitely

If repos manager is terminated, **queue manager reports error then continues, never to complain again**

- FFST generated at time of failure
- The only approved way of getting going again after the process has terminated is to restart the queue manager

V7.1 behaviour

1) Really unexpected errors immediately terminate the repository manager (e.g. connection broken, rollback failure, etc,). **See (5)**
2) If command fails, report in error log, sleep for 10 minutes, rebuild the cache and **retry for 120 hours (five days) every 10 minutes, then terminate. See (5)**
3) If command is processed correctly but commit is rolled back (resource problem) then sleep 5 seconds **and retry 2 more times then behave as (2)**
4) If command queue disabled, sleep for 10 minutes then retry indefinitely
5) If repos manager terminates, **queue manager terminates immediately**

If the new queue manager tuning parameter TolerateRepositoryFailure is set to TRUE then a failure of the repository manager will NOT terminate the queue manager

- FFST generated at time of failure and repeated every six hours while repository manager is retrying
- The only approved way of getting going again after the process has terminated is to restart the queue manager

Distributed error logs increased

- Default size of MQ error logs on distributed platforms increased from 256KB per log to 2MB
    - Override the change by setting the environment variable MQMAXERRORLOGSIZE, or setting ErrorLogSize in the QMErrorLog stanza in the qm.ini file.

Duplicate queue manager joining a cluster

- When a new queue manager joins a cluster using the same channel name as an existing cluster member the original member, AMQ9468 error is reported:
  - AMQ9468
  - Cluster receiver channel &3 has been configured by multiple queue managers
  - Queue manager &4 has joined a cluster using a cluster receiver channel with the same name as one that has already been defined by queue manager &5. All cluster receiver channels used within a cluster must be uniquely named. Only the last queue manager to join the cluster will use the named channel, Queue manager &5 will not successfully participate in the cluster while the newer queue manager is a member.

Slow repository cache maintenance and GC

- New messages written to the log when:
  - Maintenance/GC has already taken ten minutes and has yet to complete
  - Repeated at five minute intervals past that point until it completes (with the number of records processed reported to indicate progress)
  - A ‘completion’ message is issued if one or more ‘slow’ messages were issued
- AMQ9871
- Cluster maintenance has been running for 1111 minutes. Phase <insert one> has so far processed 2222 records
- AMQ9872
- Cluster maintenance has completed after 1111 minutes, 2222 records were processed
- The initial delay can be ‘tuned’
  - On distributed platforms, by setting the AMQ_CLUSTER_MONITOR_DELAY environment variable

Refresh start and ‘complete’ messages

- Prior to V7.1 a single message was issued at the point that the repository manager had generated all the messages as a result of a REFRESH CLUSTER command being processed.

- V7.1 now issues a start refresh informational message and clarifies what stage is reached when the previous ‘completed’ message is issued

- AMQ9875 REFRESH CLUSTER processing started for cluster.
- AMQ9442 Phase one of REFRESH CLUSTER has completed.

Large cache hardening operation

- AMQ9876
- Cluster management is about to compress a large number of cache records.

- AMQ9877
- Cluster cache compression has completed.
- A large cache compression has completed. This message corresponds to message AMQ9876 being previously reported.
- None.

Failure to restore cache on queue manager re-start

- Prior to V7.1, *“AMQ9422: Repository manager error, RC=1111”* reported.

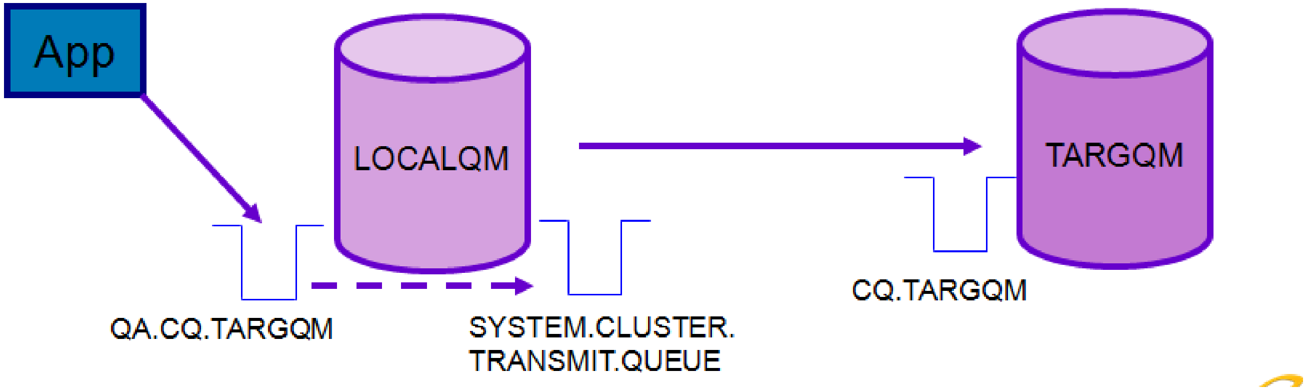
- V7.1 now issues a more specific error in the queue manager error log and issues an additional message to the console as part of strmqm
  - *“AMQ7286:An error occurred while restoring the cluster cache, see the error logs for details”* strmqm still completes
  - AMQ9873 An error occurred while restoring the cluster repository cache, reason=1111

To the console (message only):

- AMQ7286 An error occurred while restoring the cluster cache, see the error logs for details

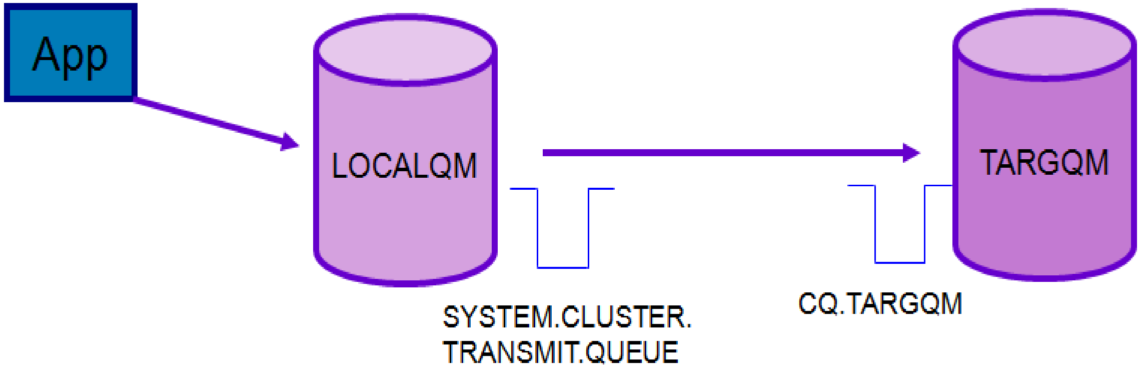
Using the new security model

- Pre WMQ 7.1:
  - Define local queue alias QA.CQ.TARGQM (targeting remote cluster queue CQ.TARGQM)...
    - **DEF QALIAS(QA.CQ.TARGQM) TARGQ(CQ.TARGQM)**
  - Give application permission to put to the local alias
    - **setmqaut -m LOCALQM -n QA.CQ.TARGQM -t queue -p myuser +put**
  - No access required to CQ.TARGQM and the cluster transmit queue



Using the new security model

- Now - using WMQ 7.1:
  - Give application permission to put to the real target queue name
    - **setmqaut -m LOCALQM -n CQ.TARGQM -t queue -p myuser +put**
  - No access required to the cluster transmit queue



How to back out to old security model

- New security stanza implemented in the qm.ini file.
- RemoteQueueAccessControl attribute:
  - To be specified under the security stanza, a new attribute/value pair is defined of the form:
    - RemoteQueueAccessControl = Xmitq
  - Only permissible value is Xmitq.
  - The default behaviour is to use the new security model.

Workload Balancing by Group

- New MQOPEN option in the MQI
  - MQOO_BIND_ON_GROUP (0x00400000, 4194304)
- 'Out of order/application managed groups not supported, so:
  - Must specify MQPMO_LOGICAL_ORDER at put time
  - Must NOT manually specify group ID (leave as MQGI_NONE).
- Ungrouped messages, or messages which do not comply with above, will fall back to BIND_NOT_FIXED
  - Except where group options would give an error today (e.g. MQRC_INCOMPLETE_GROUP)
- Use usual grouping flags within the MD of messages:

Last message in group:	MQMF_LAST_MSG_IN_GROUP
All other messages in group:	MQMF_MSG_IN_GROUP

Workload Balancing by Group: Example

- MQOPEN QX@QM1

  (MQOO_BIND_ON_GROUP)

- MQPUT Msg1

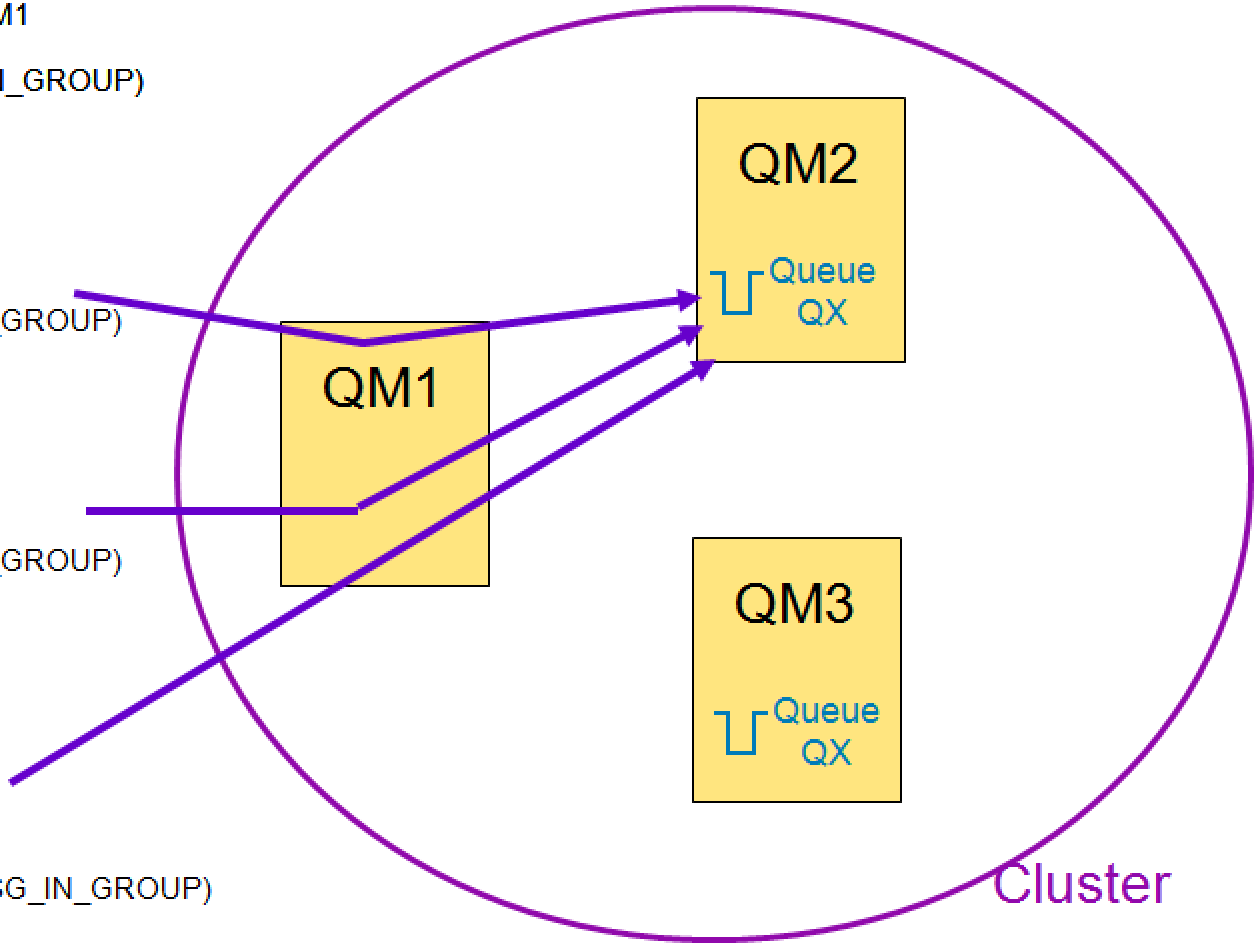
  (MQMF_MSG_IN_GROUP)

- MQPUT Msg2

  (MQMF_MSG_IN_GROUP)

- MQPUT Msg3

  (MQMF_LAST_MSG_IN_GROUP)



Workload Balancing by Group: Example

- MQOPEN QX@QM1

  (MQOO_BIND_ON_GROUP)

- MQPUT Msg4

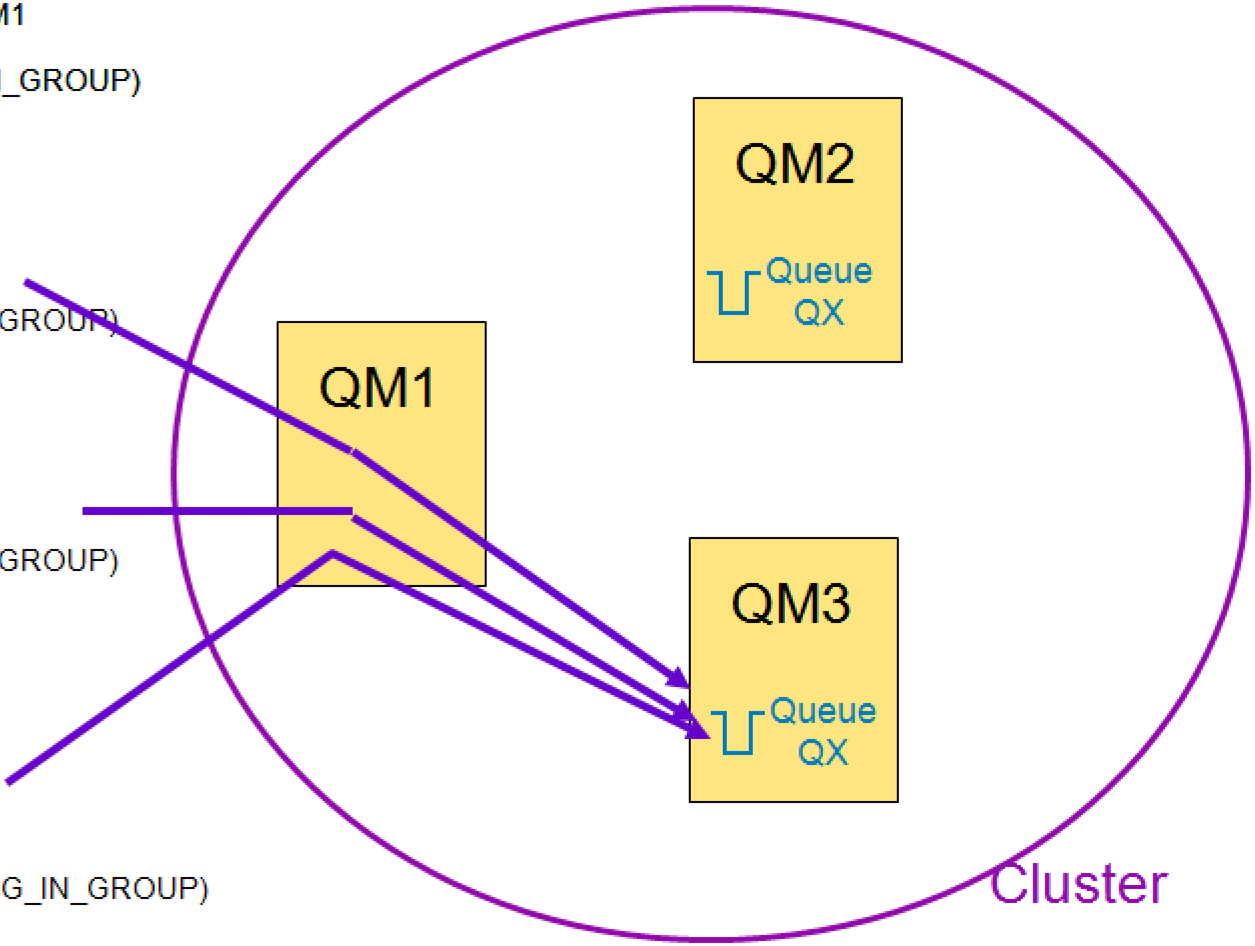
  (MQMF_MSG_IN_GROUP)

- MQPUT Msg5

  (MQMF_MSG_IN_GROUP)

- MQPUT Msg6

  (MQMF_LAST_MSG_IN_GROUP)



Publish Subscribe Cluster controls (1)

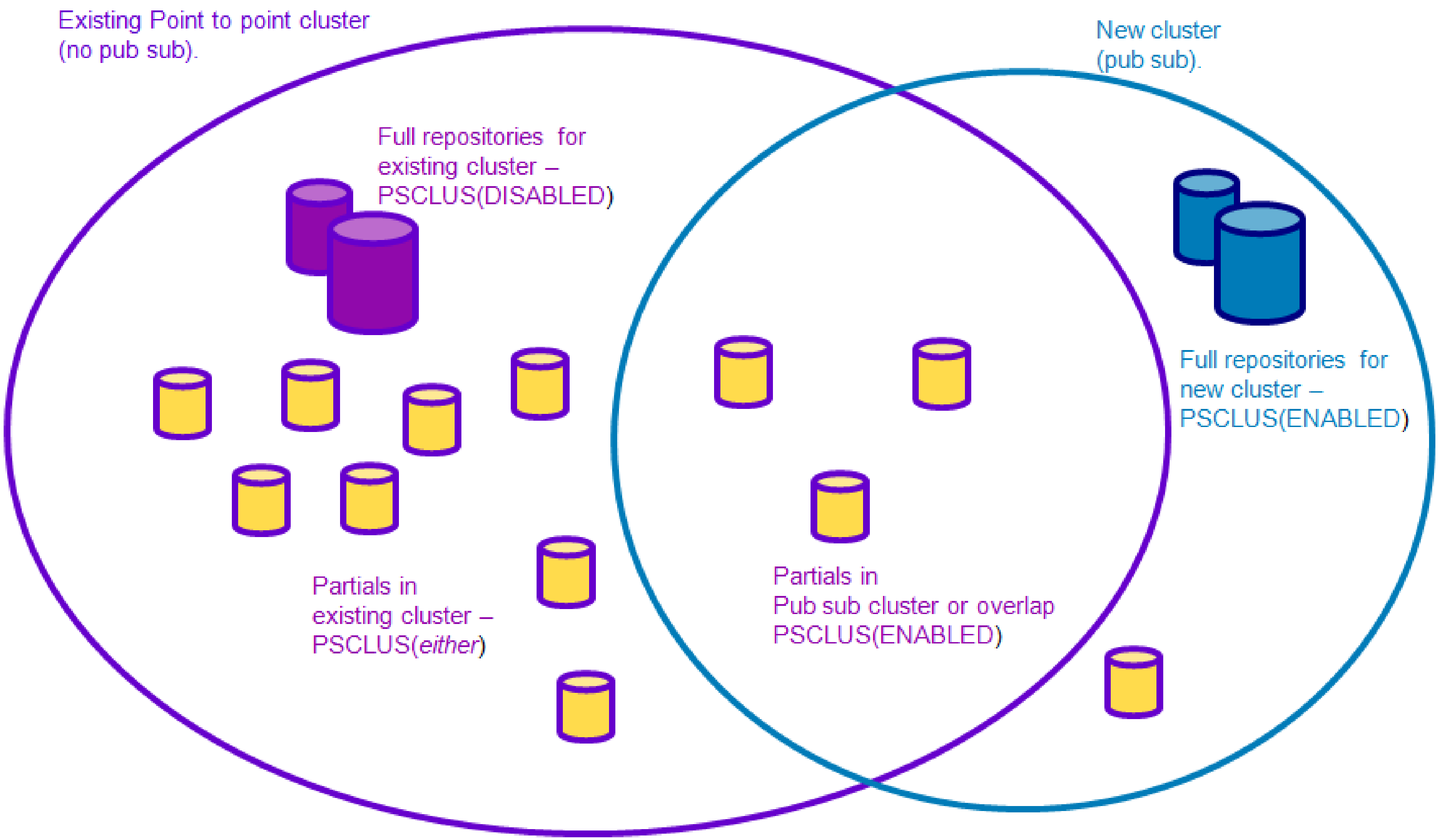
- New attribute at queue manager level controls whether or not this QM will participate in pub/sub clustering.
    - PSCLUS (ENABLED/DISABLED)

- Disables the definition of cluster topic objects, and the sending/receiving of proxy subscriptions.

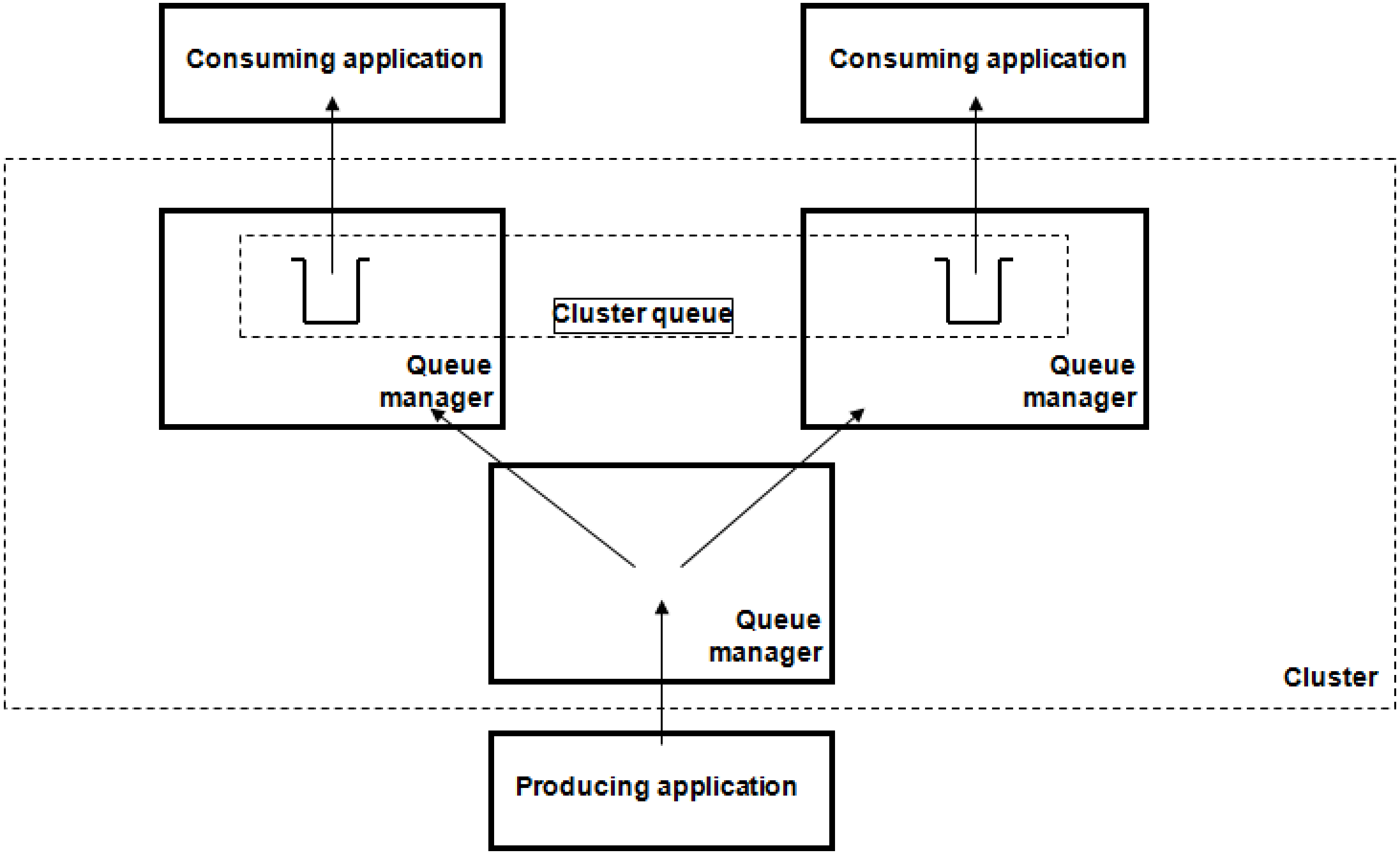
- Cannot be disabled if cluster topics already present (even defined elsewhere in the cluster).

- **Ideally** set on every queue manager if no pub sub to be used.

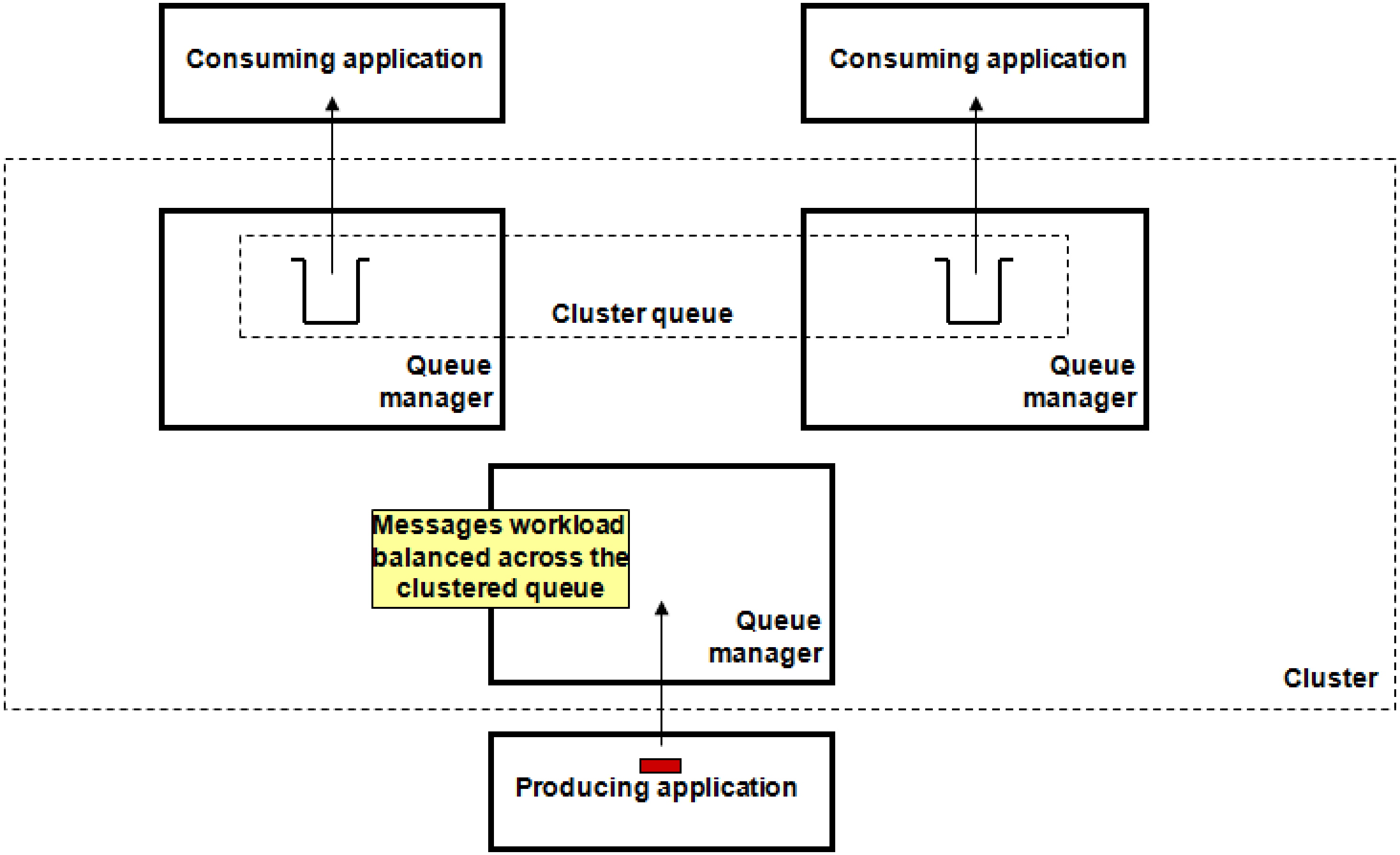
Publish Subscribe Cluster controls (2)



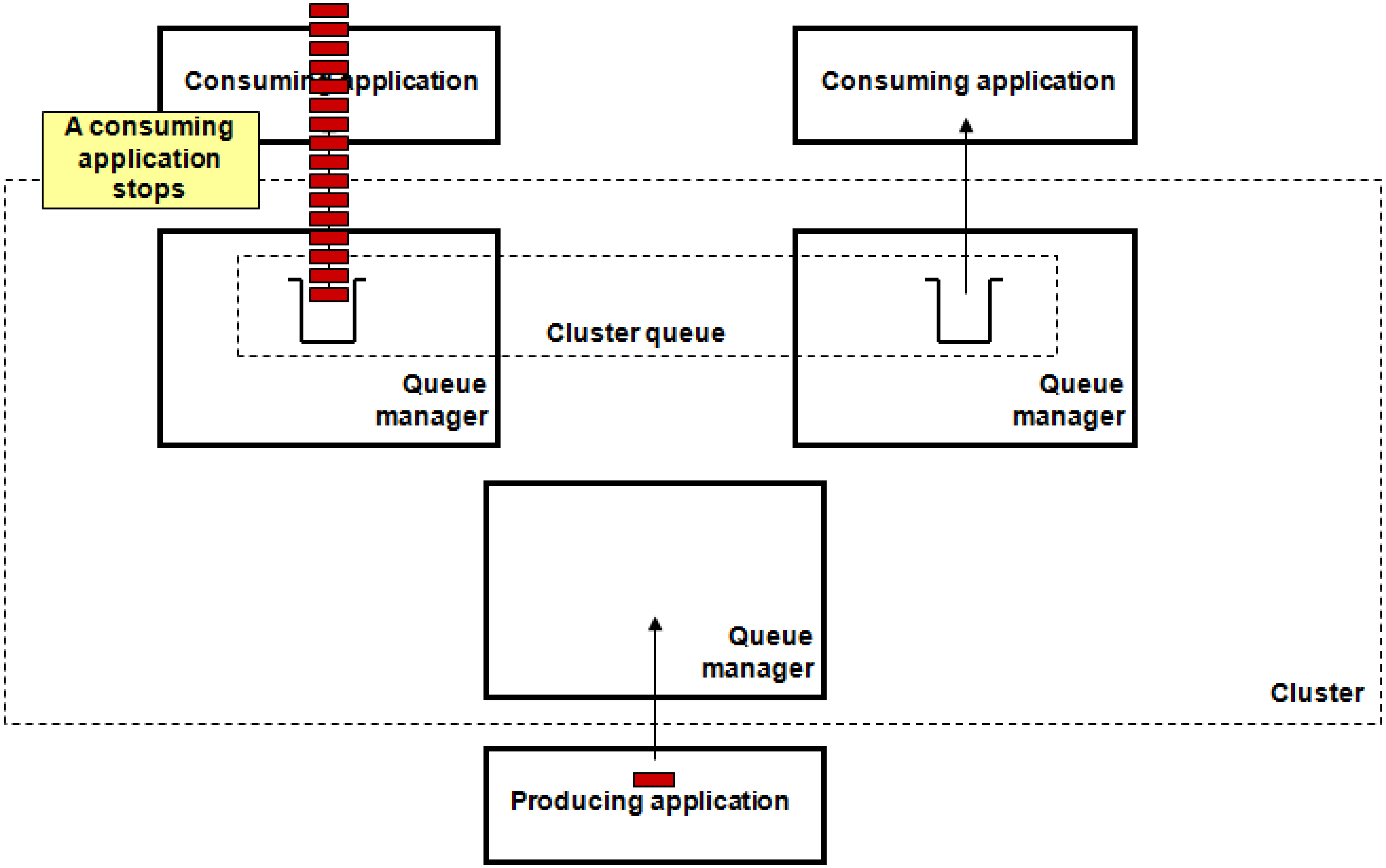
Cluster Monitoring and Rebalancing
A typical workload balancing scenario



Under normal running…



When things go wrong…



Cluster Queue Monitoring Sample

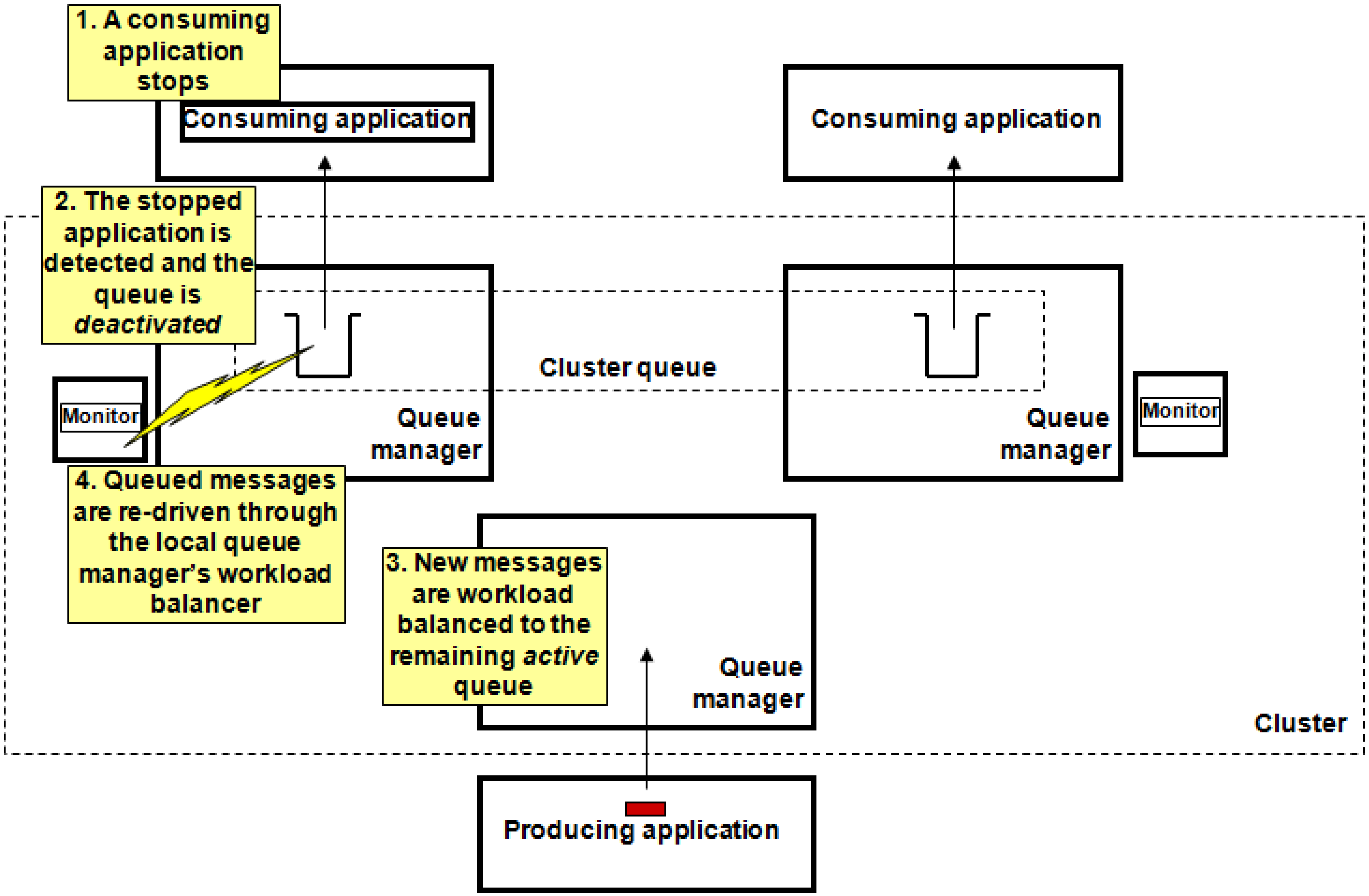
- A new tool **amqsclm** is provided to ensure messages are directed towards the instances of clustered queues that have consuming applications currently attached.
  - Move already queued messages from instances of the queue where no consumers are attached to instances of the queue with consumers.
  - The tool provides a monitoring executable to run against each queue manager in the cluster hosting queues.
  - Available on WMQ 7.0.1 fixpack 8, WMQ 7.1 later,

<mq install dir>/bin/amqsclm, source (amqsclma.c sample)

AMQSCLM Logic

- Single executable, set to run against each queue manager with one or more cluster queues to be monitored.
- The monitoring process polls the state of the queues on a defined interval:
  - If **no** consumers are attached:
    - CLWLPRTY of the queue is set to zero (if not already set).
    - The cluster is queried for any active (positive cluster priority) queues.
    - If they exist, any queued messages on this queue are got/put to the same queue. Cluster workload balancing will re-route the messages to the active instance(s) of the queue in the cluster.
  - If consumers **are** attached:
    - CLWLPRTY of the queue is set to one (if not already set).
- Defining the tool as an MQ service will ensure it is started with each queue manager

When the queues are monitored…



1. A consuming application stops

2. The stopped application is detected and the queue is *deactivated*

3. New messages are workload balanced to the remaining *active* queue

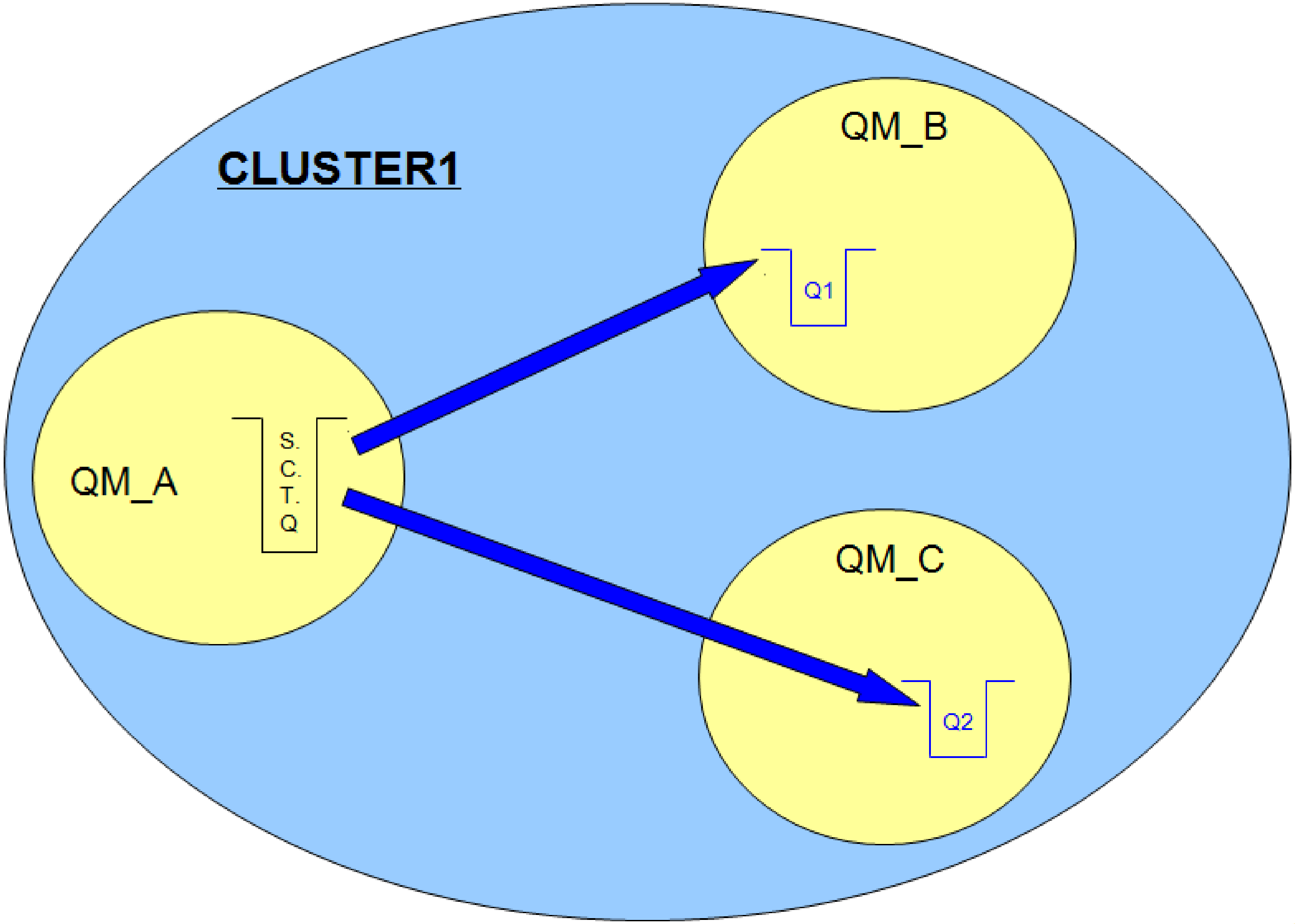
4. Queued messages are re-driven through the local queue manager's workload balancer

Limitations

- The monitoring tool is poll based.
  - The more queues to monitor the more overhead on the system - increased poll interval is recommended.
- Frequently connecting/disconnecting consuming applications will result in message churn and cluster repository overhead.
  - Tool is really suited to a set of long running consuming apps.
- Exclusive use of CLWLPRTY by the tool is required.
- If marooned messages are being relocated, any ‘BIND’ instructions are ignored.
- Monitoring is based purely on connected consuming applications, ‘slow’ consumers are not catered for.
- For simplicity the tool monitors queues within a single cluster.

WMQV7.5 Clustering new feature

The SYSTEM.CLUSTER.TRANSMIT.QUEUE

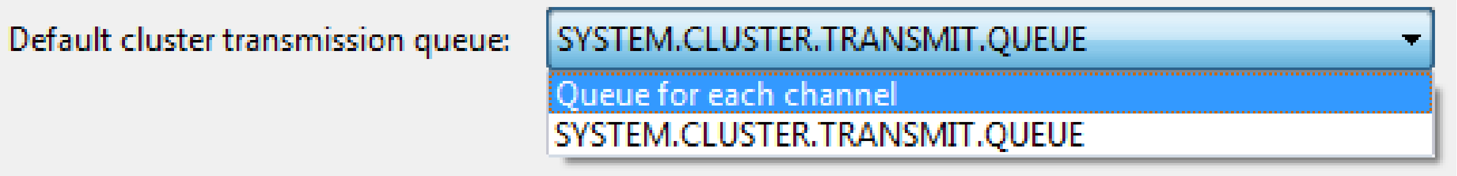


NEW: Split cluster transmit queue

- Much requested feature for various reasons…
- **Separation of Message Traffic**
  - With a single transmission queue there is potential for pending messages for cluster channel 'A' to interfere with messages pending for cluster channel 'B'
- **Management of messages**
  - Use of queue concepts such as MAXDEPTH not useful when using a single transmission queue for more than one channel.
- **Monitoring**
  - Tracking the number of messages processed by a cluster channel currently difficult/impossible using queue monitoring (some information available via Channel Status).
- **Not** about performance...

Split cluster transmit queue - automatic

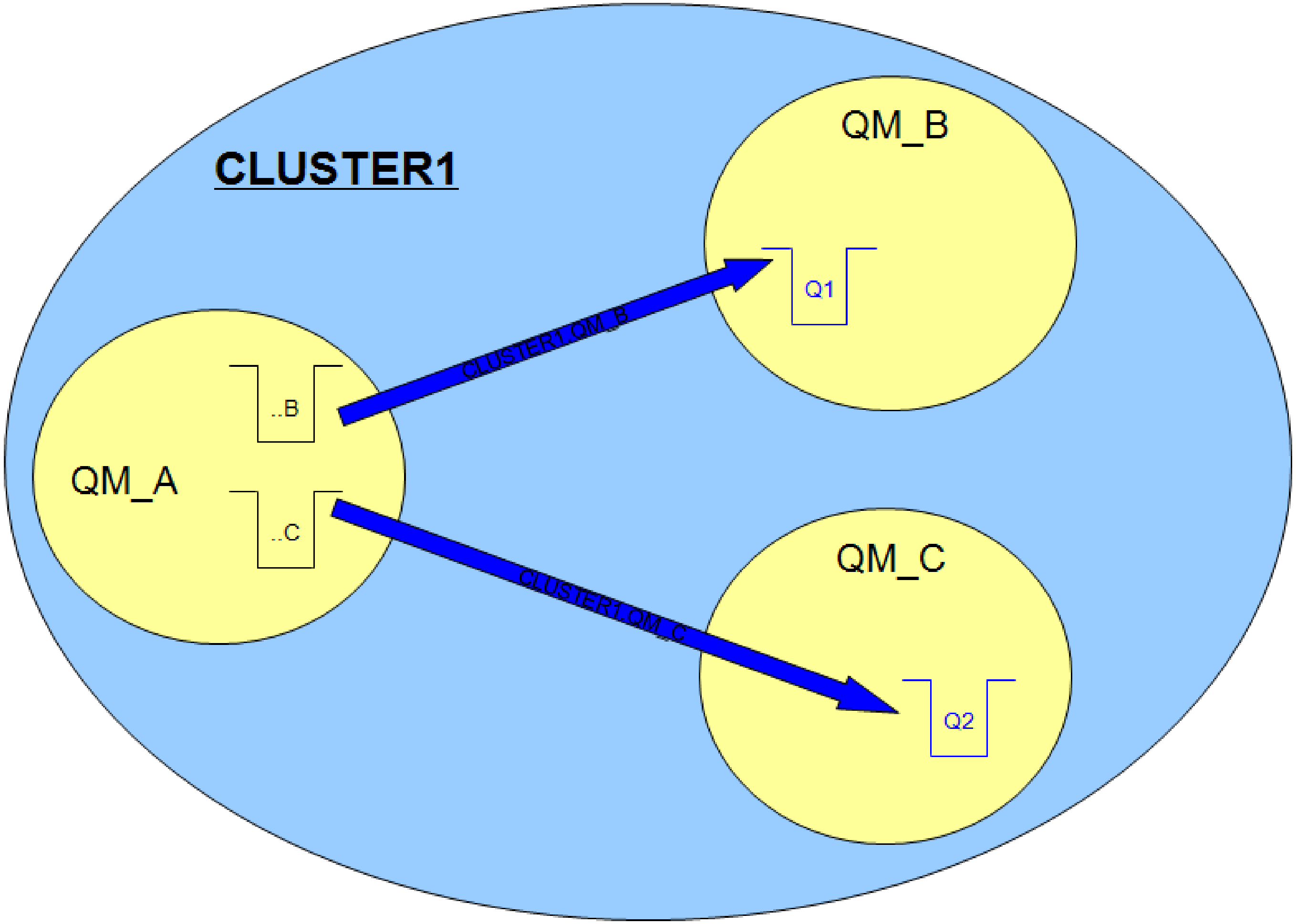
- New Queue Manager attribute which effects all cluster-sdr channels on the queue manager
    - **ALTER QMGR DEFCLXQ(SCTQ | CHANNEL)**



Default cluster transmission queue: SYSTEM.CLUSTER.TRANSMIT.QUEUE

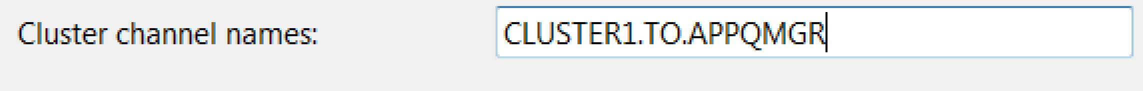
- Queue manager will automatically define a PERMANENT-DYNAMIC queue for each CLUSSDR channel.
    - Dynamic queues based upon new model queue **“SYSTEM.CLUSTER.TRANSMIT.MODEL”**
    - Well known queue names: **“SYSTEM.CLUSTER.TRANSMIT.<CHANNEL-NAME>”**

Splitting out the S.C.T.Q. per channel



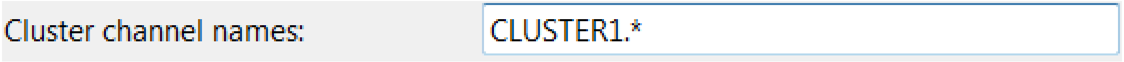
Split cluster transmit queue - manual

- Administrator manually defines a transmission queue and using a new queue attribute defines the CLUSSDR channel(s) which will use this queue as their transmission queue.
  - **DEFINE QLOCAL(APPQMGR.CLUSTER1.XMITQ) CLHLNAME(CLUSTER1.TO.APPQMGR) USAGE(XMITQ)**



Cluster channel names: CLUSTER1.TO.APPQMGR

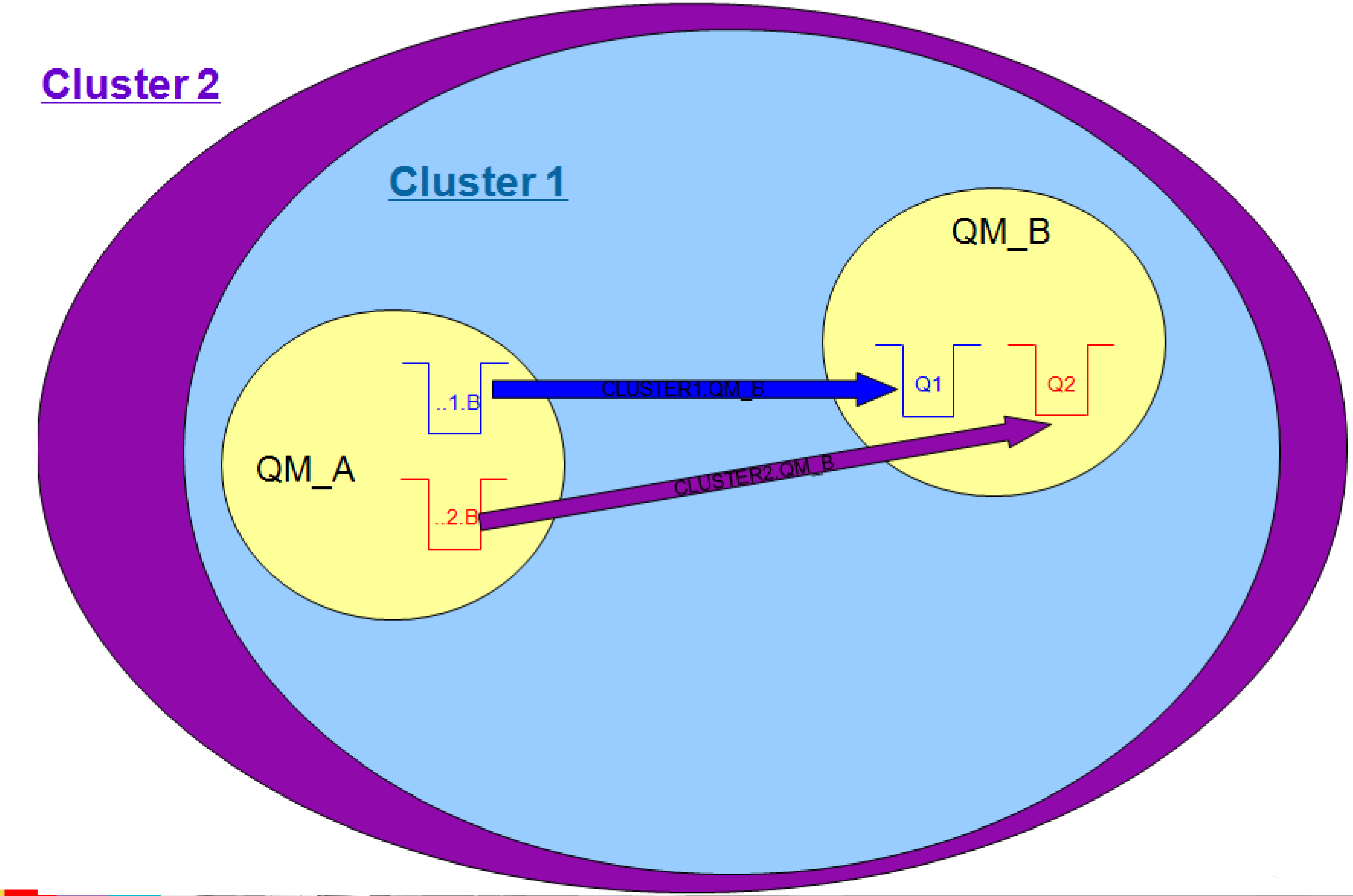
- The CHLNAME can include a wild-card at the start or end of to allow a single queue to be used for multiple channels. In this example, assuming a naming convention where channel names all start with the name of the cluster, all channels for CLUSTER1 use the transmission queue CLUSTER1.XMITQ.
  - **DEFINE QLOCAL(CLUSTER1.XMITQ) CHLNAME(CLUSTER1.*) USAGE(XMITQ)**



Cluster channel names: CLUSTER1.*

  - Multiple queues can be defined to cover all, or a subset of the cluster channels.
- Can also be combined with the automatic option
  - Manual queue definition takes precedence.

Splitting out by cluster (or application)

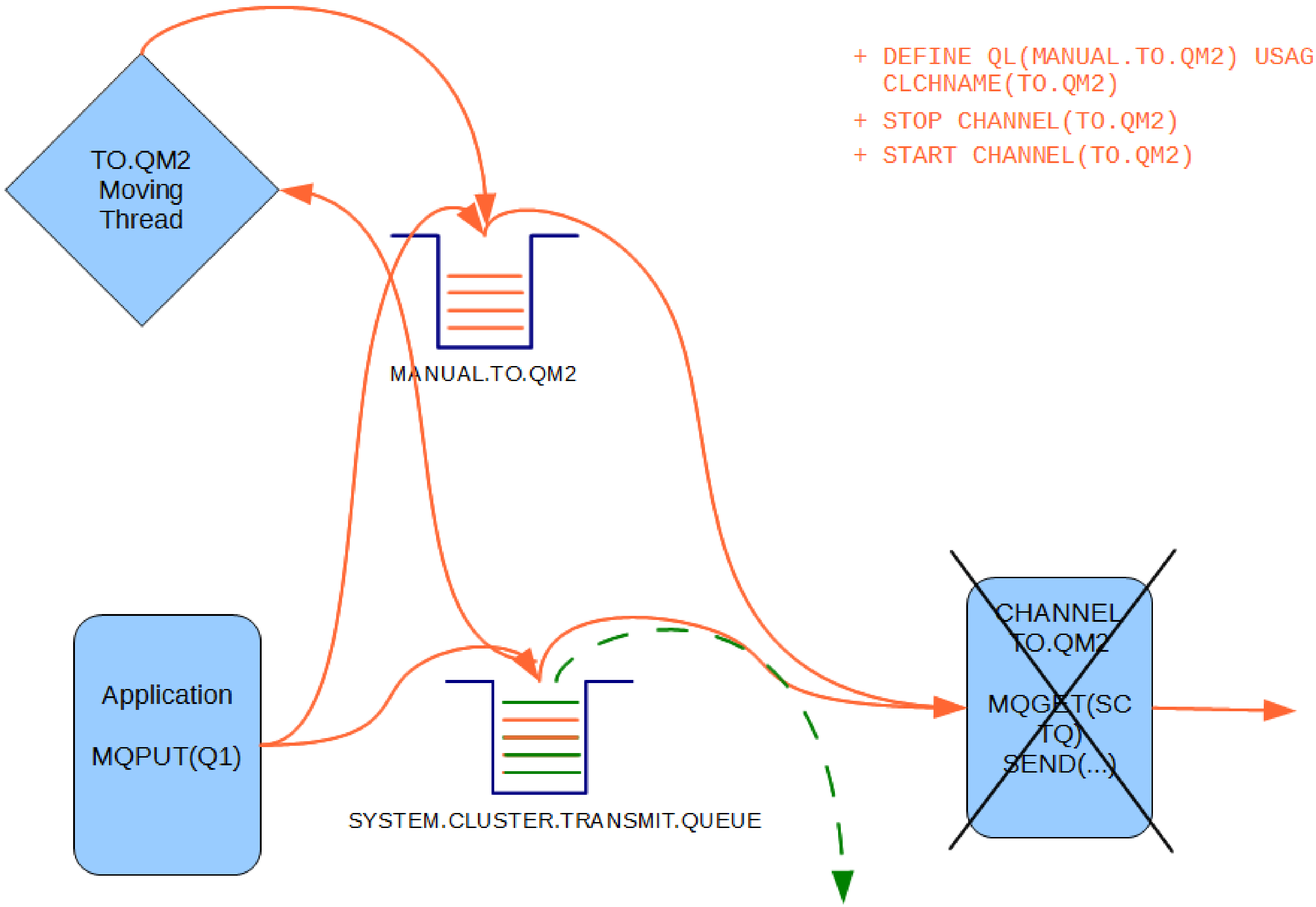


Switching of Transmission Queue

- When a CLUSSDR channel is started, the channel queries from the queue manager
  - The name of the current transmission queue for the channel
  - The name of the transmission queue according to the current configuration
- If these values are different, then after the channel has resolved it's in-doubt batch it will initiate the process of switching transmission queue before starting it's next batch of messages.
- The switching process runs asynchronously to the channel
- At the start of switching
  - Applications continue putting messages to the old transmission queue
  - The channel reads messages from the new transmission queue
  - A background thread run's moving messages from the old transmission queue to the new transmission queue
  - This stage could run for a sustained period of time
- When the background thread finds there are no more messages on the old transmission queue it atomically directs all further put's to the new transmission queue and switching is complete

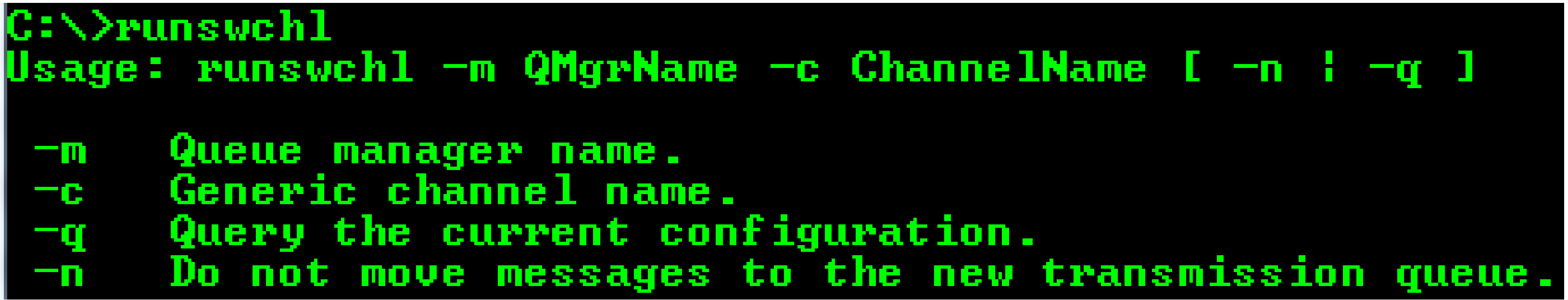
Automatic Switching – Channel start

```
+ DEFINE QL(MANUAL.TO.QM2) USAGE(XMITQ)
  CLCHNAME(TO.QM2)
+ STOP CHANNEL(TO.QM2)
+ START CHANNEL(TO.QM2)
```



Manual Switching - runswchl

- In order to allow the transmission queue used by a CLUSSDR channel to be switched without starting the channel a new command runswchl has been defined.
- runswchl -m QMgrName -c ChannelName [-n | -q]



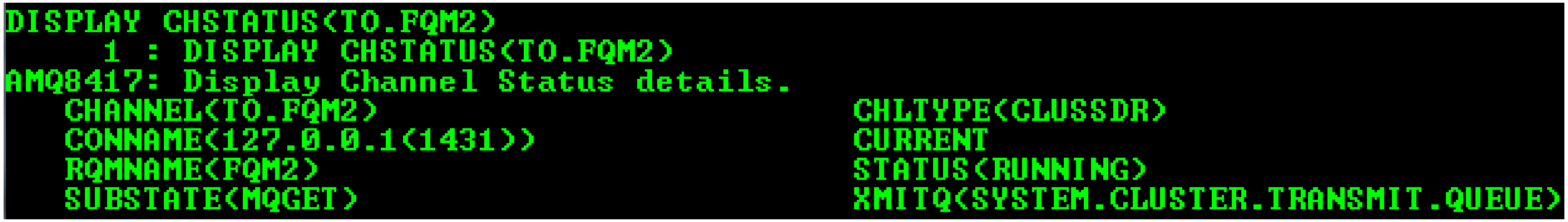
```
C:\>runswchl
Usage: runswchl -m QMgrName -c ChannelName [ -n | -q ]

 -m    Queue manager name.
 -c    Generic channel name.
 -q    Query the current configuration.
 -n    Do not move messages to the new transmission queue.
```

- Uses the same technology as used by a channel when it starts, however messages are moved synchronously and the command will end when all messages have been moved.
- The '-q' flag runs in query (or rehearsal) mode, so will not actually switch the transmission queue.
- The '-n' flag will perform the switch of transmission queue but will not move the messages. Any messages already put to the old transmission queue will be marooned and will require manual intervention in order for them to be sent on to their destination.
-

Reversing Back

- Remove all CLCHNAME options from any Transmission Queues
- Revert to default DEFCLXQ option (SCTQ)
- Restart channels for effect



```
DISPLAY CHSTATUS(TO.FQM2)
     1 : DISPLAY CHSTATUS(TO.FQM2)
AMQ8417: Display Channel Status details.
   CHANNEL(TO.FQM2)                        CHLTYPE(CLUSSDR)
   CONNAME(127.0.0.1(1431))                CURRENT
   RQMNAME(FQM2)                           STATUS(RUNNING)
   SUBSTATE(MQGET)                         XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

A Known problem in Splitting Cluster transmit queue

- APAR IC98353 (not closed yet)

- Problem scenario:

- when a partial repository at version 6, 7 or 7.1 attempts to join a cluster which has full repositories at version 7.5 using split cluster transmit queues, the clussdr channel from the second full repository qmgr to the old version partial qmgr can’t start normally, which shows “RETRYING” or “NOT FOUND” status.

- Monitor APAR status, applying ifix when it available.

.

http://www-947.ibm.com/support/entry/portal/product/websphere/websphere_mq?productContext=24824631

Note: You can’t find it at this point since APAR is not published yet, but you can try to search it few weeks later using keyword “IC98353” in search box from above MQ support portal website.

Additional WebSphere Product Resources

- MQDev community/blog ss https://www.ibm.com/developerworks/community/blogs/messaging/?lang=en_us
- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at: http://www.ibm.com/software/websphere/support/supp_tech.html
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/
- Join the Global WebSphere Community: http://www.websphereusergroup.org
- Access key product show-me demos and tutorials by visiting IBM Education Assistant: http://www.ibm.com/software/info/education/assistant
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically: http://www.ibm.com/software/websphere/support/d2w.html

References

- Troubleshooting WebSphere MQ clusters
  http://www-01.ibm.com/support/docview.wss?uid=swg21620882
- WebSphere MQ Clustering - Cluster hints and tips
  http://www-01.ibm.com/support/docview.wss?uid=swg21229905&acss=wmq030812#1
- Ask the Experts Replay: MQ Clustering Gotchas! Avoiding Cluster Administration Pitfalls
  http://www-01.ibm.com/support/docview.wss?uid=swg27039756
- Considerations in Designing and Maintaining a WebSphere MQ Clustering Environment
  http://www-01.ibm.com/support/docview.wss?uid=swg27017264
- Cluster setup and basic usage of clustered queues and topics in WebSphere MQ 7
  http://www-01.ibm.com/support/docview.wss?uid=swg27038687
- WebSphere MQ queue manager clustering
  http://www-01.ibm.com/support/docview.wss?uid=swg27009247
- WebSphere MQ V7 Clustering
  http://www-01.ibm.com/support/docview.wss?uid=swg27024558
- Implementation Considerations for Multi-Instance Queue Managers in WebSphere MQ Cluster Environment
  http://www-01.ibm.com/support/docview.wss?uid=swg27018127

Questions and Answers