Disclaimer

- 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.
Broad Agenda

- General MQ implementation remarks
- MQ Client
- XA Transactions
- MQ v8 Design
- MQ v8 Beta Overview or Demo
- MQ v8 Beta Upcoming Features
- MQ and Disaster Recovery
IBM MQ for HP Non Stop Server Early/Beta program

- **Beta program for next version of MQ on HP Non Stop Server**
  - Opportunity to try your applications with the beta code
  - Update and discussion calls with the development team for beta participants
  - Provide feedback to the product team
  - Support for any questions
  - Advance information to help with your planning
  - Invite to any beta program workshops/education events

- **Joining the beta program**
  - Nomination from either your local IBM contact or the beta program manager
  - IBM asks you to accept standard beta program terms and conditions
  - Any questions on the beta program
    - Please ask the beta program manager....
    - Pete Murphy, Email: pete_murphy@uk.ibm.com or davidward@us.ibm.com
Multi-threading: Shades of meaning

- Most distributed platforms (i.e. Unices) have a feature called Posix Threads (pthreads)
- Multiple executable units within a single process
  - PS. Linux implements threads as light weight processes with the same pid.
- Each thread has its own stack and is not necessarily aware of other threads in the process
- Advantages:
  - Available on many platforms, even NonStop
  - In simpler cases, each thread can look like its own program
Multi-threading: Shades of meaning

- Some platforms implement their Posix threads (pthreads) in their OS kernel
- Often called “kernel-space” threads
- Advantages are that threads are dispatched by the kernel along side untheaded process.

- On NonStop, pthreads are NOT kernel supported.
- HP provides a PUT DLL to support threading within a process.
- Sometimes called “User-space” threads
Multi-threading: Shades of meaning

- **Pthreads programming is quite easy if ...**
  - Threads don’t need to synchronize with each other
  - But this isn’t typically the case

- **Pthreads synchronization APIs are complex**
  - Mutex
  - Condition Variable
  - Synchronizing access to Global variables
Multi-threading: Shades of meaning

- On NonStop, the term “Multi-threading” often means
  - A server program that uses no-wait I/O and AWAITIOX to process multiple requests in parallel

- On Unix, this is often called multi-plexing

- MQ 5.3’s Queue Server is an example of a “multi-threaded” program in this sense
MQ on NonStop: Multi Threading

- MQ v8 on NonStop makes heavy use of Posix threads
- MQ v5.3 on NonStop makes heavy use of no-waited “multi-threading”

- MQ v8 server is likely to evolve towards more use of no-waited I/O and less use of Posix threads.
MQ on NonStop: Multi Threading

- Most Linux/Unix/Windows versions support kernel level threading
- The NonStop operating system does not support kernel level threading
- The NonStop compilers provide two kinds of thread libraries: SPT (older version) PUT (newer version)
- Program can use either version, but not both
MQ on NonStop: Shared Memory

- MQ server makes heavy use of shared memory
- Why? Using shared memory for inter-process-communication is very fast
MQ on Linux

Queue Manager

Applications

- Application Code
- MQM Library

Shared Memory

- Queue Manager State
- Queue State
- Cluster Repository Cache
- Buffers

Queue Manager Processes

Filesystem
MQ v8 on Nonstop

Note: Applications can run in any CPU when using isolated bindings
MQ Client
Connecting NonStop to other MQ Servers

- **MQ Client is a linkable (native only) library**
  - C-lang
  - COBOL
  - pTAL

- **Connects to other MQ Servers within the network on any platform**

- **Supports (almost) all MQ API functions**

- **Supports TMF transactional work**
MQ Client – Connects to a remote Queue Manager

Applications

Application Program

MQIC Library

Network

Queue Manager

Linux
AIX
HPUX
Solaris
Windows
zOS

Queues
Channels

TMF

BEGINTRANSACTION
ENDTRANSACTION
ABORTTRANSACTION

MQIC
MQPUT

MQ TX signals

TMF Signaling

MQ TMF Gateway

Network
MQ Client and TMF

- Applications do BEGIN-/ENDTRANSACTION
- MQPUT and MQGET done within transaction context
- Expectation: COMMIT/ROLLBACK (ENDTRANSACTION/ABORTTRANSACTION) will include MQ related work (if SYNCPOINT used)
- How does this work when MQ Server on different platform not knowing about TMF?
Distributed Transactions – the non-TMF way

- XA - eXtended Architecture
- X/Open Group Standard for Distributed Transactions
- Defines a 2-Phase Commit Protocol (2PC)
Transactions on other Platforms

- Linux or Windows (as examples) do not have a transaction manager like TMF as part of the OS
- Databases on Linux or Windows come with their own integrated transaction manager
- So does MQ Server on those platforms
- The XA technology allows to use one common transaction for database and MQ I/O
Platform – Transactions

- **A Linux Queue Manager can be configured to be**
  - XA Transaction Manager
  - XA Resource Manager
  - Neither

- **MQ applications can**
  - Start and use local uncoordinated transactions
  - Start and use global coordinated transactions
  - Use a global coordinated transaction started by a foreign TM
Platform – Global Transactions with MQ as the coordinator

- Configure the Queue Manager as an XA Transaction Manager (TM)
- Configure DB2 as an XA Resource Manager (RM) coordinated by MQ

- MQBEGIN() Starts a new global transaction
- MQPUT() with SYNCPOINT Puts a message within the global transaction
- MQGET() with SYNCPOINT Gets a message within the global transaction
- *Do some DB2 SQL work …*
- MQCMIT() Commits the global transaction
  MQPUT, MQGET and DB2 are all committed

Alternatively

- MQBACK() Rolls back the global transaction
  MQPUT, MQGET and DB2 are all rolled back
Platform – Global Transactions with MQ as the subordinate

- Configure DB2 as an XA Transaction Manager (TM)
- Configure the Queue Manager as an XA Resource Manager (RM) coordinated by DB2

- **DB2 BEGIN TRANSACTION**
  - Tells DB2 start a new global transaction

- MQPUT() with SYNCPOINT
  - Puts a message within the global transaction

- MQGET() with SYNCPOINT
  - Gets a message within the global transaction

- **DB2 END TRANSACTION**
  - Tells DB2 to commit the global transaction
  - MQPUT, MQGET and DB2 are all committed

Alternatively

- **DB2 ROLLBACK**
  - Tells DB2 to rollback the global transaction
  - MQPUT, MQGET and DB2 are all rolled back
Platform – Global TMF Transactions with MQ

This is what NonStop customers expect MQ to support

- `BEGINTRANSACTION()`  - Asks TMF to start a new transaction
- `MQPUT()` with `SYNCPOINT`  - Puts a message within the TMF transaction
- `MQGET()` with `SYNCPOINT`  - Gets a message within the TMF transaction
- *Do some Guardian file or SQL work*
- `ENDTRANSACTION()`  - Asks TMF to commit the transaction
  - MQPUT, MQGET and file/db are all committed

Alternatively

- `ABORTTRANSACTION()`  - Asks TMF to rollback the global transaction
  - MQPUT, MQGET and file/db are all rolled back
MQ beta – TMF integration with MQ server

- TMF is the Transaction Coordinator/Manager
- Qmgr is a subordinate XA-RM
- MQ TMF Gateway maps TMF signals to XA calls
MQ on NonStop Products (Now and Future)

<table>
<thead>
<tr>
<th></th>
<th>IA64</th>
<th>x86</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Client</strong></td>
<td>MQ 8 Client</td>
<td>TBD</td>
</tr>
<tr>
<td></td>
<td>Released Sept 2014</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SupportPac MQC8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MQ 7.1 Client</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Released Jun 2013</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SupportPac MAT1</td>
<td></td>
</tr>
<tr>
<td><strong>Server</strong></td>
<td>Intended MQ 8 Server</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MQ 5.3.1 Server</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Current fixpack 5.3.1.10</td>
<td></td>
</tr>
</tbody>
</table>
MQ 8 Server beta

- IA64 platform (x86 later)
- Major upgrade to MQ 5.3.1 server
- Contains most MQ v6.x, v7.x and v8 features
- Notable exceptions
  - Advanced Message Security
  - LDAP authentication
  - MFT/File Transfer Edition
MQ 8 server - features carried over from 5.3.1

- Guardian native application support
- OSS unthreaded and multi-threaded application support
- Multiple MQ installations per NonStop system
- TMF integration
- EMS events *
- Java/JMS *
- MQGET SET_SIGNAL *
- SSL channels *
MQ 8 server - features *not* carried over from 5.3.1

- **Non-native (TNS) application support**
  - Non-native C, COBOL and TAL*** not supported
- **Standard Posix Threads (SPT) OSS application support** *
- **PATHWAY control over MQ processes** **

* SPT may be added for JDK 6 support
** PATHWAY support being reviewed
*** Native pTAL is supported
MQ 8 Server Beta 1

- Released Thursday March 19th
- H-Series and J-Series
- Requires at least J06.15 or H06.26
- Install package
  
  `mqsh-8.0-hpns-nse64-beta1.run`

- Requires OSS and TMF to be enabled
MQ 8 server Installation

- Packaged as a runnable OSS program file

    mqs-8.0-hpns-nse64.run

- Installed from an OSS shell prompt
- No separate instmqm script needed
- One OSS path and one Guardian path

    ./mqs-8.0-hpns-nse64.run

    -i <OSSpath>
    -g <Guardianpath>
MQ 8 server Installation

```
./mq8-8.0-hpns-nse64-beta2.run -i mq8beta2 -g data09.mq8beta2
#
# WebSphere MQ Server 8.0 for HP NonStop Server
#
# Fixpack       : 8.0.0.2
# Architecture  : nse64
# Build         : p800-L150420-085849
#
# MQ Install Path : /home/david/mq8beta2
#                  $DATA09.MQ8BETA2 (/G/data09/mq8beta2)
#
# MQ owner      : MQM.DAVE 44,11
#
# System Name   : MARVIN
# RVU           : J06.18.01
# UNAME         : NONSTOP_KERNEL marvin J06 18 NSE-AB
#
# Fri Apr 24 13:03:40 EDT 2015
#---------------------------------------------------------------

Creating OSS 'opt' tree and Guardian sub-volume ...

100% [================================] 827/827 files  543MB ET 01:05

Setting OSS 'opt' tree attributes [OK]
Setting OSS 'opt' tree permissions [OK]
Setting Guardian file attributes [OK]
Setup misc 'opt' tree files [OK]
Creating OSS 'var' tree [OK]
MQ install successful [01:43 time elapsed] [OK]
```
MQ Installation Awareness

- MQ v8 is very different from MQ 5.3 in this area

- MQ 5.3 required **MQNSKOPTPATH/MQNSKVARPATH** environment variables and TACL params for
  - MQ commands
  - Application programs

- MQ v8 does not need any environment variables or TACL params (normally)
  - MQ commands (crtmqm, strmqm, runmqsc etc)
  - Application programs
Co-existence

- MQ 8 server can be installed multiple times on the same Nonstop system

- MQ 8 server can be installed on the same Nonstop system as:
  - MQ 5.3 server
  - MQ 7.1 client
  - MQ 8 client

- Each installation needs unique OSS and Guardian install locations
Running MQ administration tools

- MQ 8 tools do not require environment variables
  - MQNSKOPTPATH and MQNSKVARPATH are ignored

- Either run the tool directly or put the `opt/mqm/bin` directory in your PATH

  ```bash
  <mqinstall>/opt/mqm/bin/dspmqver
  
  or
  
  export PATH=<mqinstall>/opt/mqm/bin:$PATH
dspmqver
  ```
Building MQ applications

**Guardian**

```bash
CCOMP /in mqprogc/ mqprog;
    runnable, extensions, nolist,
    ssv0 "$system.system",
    ssv1 "$data04.mq8",
    eld(-L$data04.mq8 -lmqm)
```

**OSS**

```bash
c89 -Wextensions
    -I<mqinstall>/opt/mqm/inc
    -L<mqinstall>/opt/mqm/lib -lmqm
    -o mqprogram
    mqprogram.c
```
Running MQ Applications

- An application that has been linked against the MQ 8 libraries …
  - Does not need any MQ or _RLD_LIB_PATH environment variables or defines.
  - Will use the MQ installation that it was (last) linked with.

- Re-link or set _RLD_LIB_PATH to use a different MQ installation

- MQ 5.3 applications can be run against MQ 8

  ```
  export _RLD_FIRST_LIB_PATH=<mqinstall>/opt/mqm/lib
  
  add define = _RLD_FIRST_LIB_PATH, class search,
  
  subvol0 $data04.mq8
  ```

- Note: known problem described later (RLD warnings)
Limitations

- **The following features are not supported in beta 1**
  - Java/JMS bindings
  - SSL channels
  - runmqsc FIXCOMMAND (FC and !)
  - MQ EMS events
  - MQGET SET SIGNAL
  - crtmqm and runmqlsr selection of TCPIP process name (-g)

- **Non-native TNS C, COBOL and TAL applications are not supported**
Limitations – Single CPU

- MQ v8 beta uses a single-CPU design
- The CPU used to start the queue manager is called the HOME CPU of the queue manager.
  
e.g. run –cpu=3 <mqinstall>/opt/mqm/bin/strmqm QMGR

- Most MQ admin tooling must be run in the HOME CPU
  
e.g. run –cpu=3 <mqinstall>/opt/mqm/bin/runmqsc QMGR

- MQ (standard-bound) applications can run in any CPU
  
e.g. run –cpu=3 mqprogram
  
  mqprog /cpu 3/

- Using the wrong CPU will result in MQCONN 2059 reason code or “Queue Manager not available”
Known problems (beta 1)

- Creating a queue manager with periods in its name will result in TMF gateway start failure FDC
  - Workaround: avoid queue manager names containing non-alphanumeric characters
  - Fixed in beta 2

- Deadlock or MQCONN failure reason 2059 when simultaneous applications call MQCONN
  - Being investigated  Fixed in beta 2

- amqzmgr0 may be left running with runmqlsr after endmqm
  - Workaround: Use “endmqlsr –m QMGR” to stop the listener

- RLD warnings when running MQ v8 admin tools in an OSS window with _RLD_FIRST_LIB_PATH environment variable
  - Workaround: use separate OSS logins for MQ admin tools and MQ applications
Compatibility

- MQ 5.3 native application programs can run without change
  - OSS unthreaded
  - OSS PUT multi-threaded
  - Guardian (native)

- Native C-language (c89 and c99)
- Native C++ (version 3 only if using IMQI classes)
- Native COBOL
Security - CHLAUTH

- **Set rules (via MQSC, PCF or Explorer) to permit/deny inbound connections**
  - Inbound clients
  - Inbound message channels

- **CHLAUTH Rules can**
  - Allow a connection
  - Allow a connection and assign an MCAUSER
  - Block a connection
  - Block privileged access
  - Control SSL Peer Name matching

- **CHLAUTH Rules can use any of the following identifying data**
  - IP address
  - SSL/TLS DN
  - Client userid
  - Remote queue manager name
Security - CHLAUTH

- Create CHLAUTH rules using
  - MQSC
  - PCF
  - MQ Explorer
Security - CONNAUTH

- The ability for an application to provide a user ID and password when connecting to MQ

- ALTER QMGR CONNAUTH(USE.PW)
- DEFINE AUTHINFO(USE.PW)
  - AUTHTYPE(xxxxxxxx)
  - CHKCLNT(REQUIRED)
  - CHKLOCL(OPTIONAL)
- REFRESH SECURITY TYPE(CONNAUTH)
Administration

- MQSC DISPLAY QSTATUS
- MQSC LISTENERs and SERVICEs
- MQSC FILTER
- Accounting Reports
- Statistics Reports
- TraceRoute Messages
MQ 8 server – MQI enhancements

- Publish-subscribe
- Message Selectors
- Message Properties
- Asynchronous Consume
MQI – Publish/Subscribe

- Previously provided by the MQ 5.3.1 Pubsub Broker (strmqbrk/endmqbrk)
- Now supported directly by the queue manager
- runmqsc support for TOPICs as well as QUEUES
- Publish done using MQPUT to a TOPIC
- Subscribe done using new MQSUB call; then receive publications using an MQGET call
- MQOPEN extended to support TOPICs
MQI Asynchronous Consume

- A message-driven function or routine called by the Queue Manager when a message is ready to be delivered
- No MQGET needed and no buffer need be provided by the applications
- Fewer resources allocated waiting for a message to arrive
- Enabled using the MQCB and MQCTL calls
Client Conversation Sharing

- Also known as client connection multiplexing
- Multiple client connections can flow down the same TCPIP socket
- Resource usage on queue manager is reduced
  - Fewer threads
  - Much lower memory footprint
- New SHARECNV channel attribute
## MQ V6 Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPV6 Channels</td>
<td>MQ v6.x</td>
</tr>
<tr>
<td>Cluster Workload Balancing</td>
<td>MQ v6.x</td>
</tr>
<tr>
<td>MQSC DISPLAY QSTATUS (enhanced in v6 ??)</td>
<td>MQ v6.x</td>
</tr>
<tr>
<td>MQSC LISTENER</td>
<td>MQ v6.x</td>
</tr>
<tr>
<td>MQSC SERVICES</td>
<td>MQ v6.x</td>
</tr>
<tr>
<td>MQSC FILTER</td>
<td>MQ v6.x</td>
</tr>
<tr>
<td>Accounting Reports</td>
<td>MQ v6.x</td>
</tr>
<tr>
<td>Statistics Reports</td>
<td>MQ v6.x</td>
</tr>
<tr>
<td>TraceRoute Messages</td>
<td>MQ v6.x</td>
</tr>
</tbody>
</table>
### MQ V7.0, 7.0.1, 7.1, 7.5 Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publish-Subscribe MQI</td>
<td>MQ v7.0</td>
</tr>
<tr>
<td>Message Selectors</td>
<td>MQ v7.0</td>
</tr>
<tr>
<td>Message Properties</td>
<td>MQ v7.0</td>
</tr>
<tr>
<td>Async Consume</td>
<td>MQ v7.0</td>
</tr>
<tr>
<td>Conversation Sharing</td>
<td>MQ v7.0</td>
</tr>
<tr>
<td>Multi-Instance Queue Managers</td>
<td>MQ v7.0.1</td>
</tr>
<tr>
<td>Command &amp; Configuration Events</td>
<td>MQ v7.0.1</td>
</tr>
<tr>
<td>Pubsub Routing Exit</td>
<td>MQ v7.0.1</td>
</tr>
<tr>
<td>Channel access control with CHLAUTH</td>
<td>MQ 7.1</td>
</tr>
<tr>
<td>setmqaut on non-local objects</td>
<td>MQ 7.1</td>
</tr>
<tr>
<td>Channel USEDLQ attribute</td>
<td>MQ 7.1</td>
</tr>
<tr>
<td>dmpmqcfg</td>
<td>MQ 7.1</td>
</tr>
<tr>
<td>Multiple Cluster Transmit Queues</td>
<td>MQ v7.5</td>
</tr>
<tr>
<td>Advanced Message Security</td>
<td>MQ v7.5</td>
</tr>
</tbody>
</table>
## MQ V8 Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic Host Routing for Pubsub Clusters</td>
<td>MQ V8</td>
</tr>
<tr>
<td>JMS 2.0</td>
<td>MQ V8</td>
</tr>
<tr>
<td>LDAP or Local OS authentication</td>
<td>MQ V8</td>
</tr>
<tr>
<td>CHLAUTH domain name support</td>
<td>MQ V8</td>
</tr>
<tr>
<td>QLOAD utility</td>
<td>MQ V8</td>
</tr>
</tbody>
</table>
IBM MQ for HP Non Stop Server Early/Beta program

- Beta program for next version of MQ on HP Non Stop Server
  - Opportunity to try your applications with the beta code
  - Update and discussion calls with the development team for beta participants
  - Provide feedback to the product team
  - Support for any questions
  - Advance information to help with your planning
  - Invite to any beta program workshops/education events

- Joining the beta program
  - Nomination from either your local IBM contact or the beta program manager
  - IBM asks you to accept standard beta program terms and conditions
  - Any questions on the beta program
    - Please ask the beta program manager ....
    - Pete Murphy, Email: pete_murphy@uk.ibm.com or davidward@us.ibm.com
More information

- MQ v8 Knowledge Center
- MQ v8 PDF documents
  ftp://public.dhe.ibm.com/software/integration/wmq/docs/V8.0/PDFs/
- davidward@us.ibm.com