

Session 13928

Can I consolidate my Queue managers and Brokers?

Morag Hughson – hughson@uk.ibm.com
Dave Gorman – Dave_Gorman@uk.ibm.com



Abstract



N
O
T
E
S

- WebSphere MQ queue managers and brokers are designed to be shared resources, utilised by multiple different applications. It is not uncommon for us to see users providing a brand new queue manager or broker for every new application. This session intends to answer that question once and for all, and also strives to answer the question, when DO I need a new queue manager/broker? As well as the high level question of when you need a complete new queue manager/broker it will also look into when you need the resources contained within such as queues, channels, logs and execution groups.

Consolidate

consolidate

con·sol·i·date [*kuh n-sol-i-deyt*] *verb*,
con·sol·i·dat·ed, **con·sol·i·dat·ing**, *adjective*
verb (used with object)



1. to bring together (separate parts) into a single or unified whole; unite; combine: *They consolidated their three companies.*
2. to discard the unused or unwanted items of and organize the remaining: *She consolidated her home library.*
3. to make solid or firm; solidify; strengthen: *to consolidate gains.*
4. *Military* . to strengthen by rearranging the position of ground combat troops after a successful attack.

Complete your sessions evaluation online at SHARE.org/BostonEval

Consolidate

N

- Just to ensure we are all on the same page, we show the definition of “consolidate” here. This definition is taken from Dictionary.com.

O

- We note that to consolidate you start with separate parts. For the purposes of this presentation, these separate parts are queue managers and/or brokers.

T

E

S

Questions for today

- When can I consolidate my queue managers/brokers?
- When do I need a new queue manager/broker?



Complete your sessions evaluation

Questions for today - Notes

N

- We really have two questions we want to answer here.
 - When can I consolidate my queue managers?
 - When do I need a new queue manager?

O

- Let's first think about why we end up with multiple queue managers. There are a good number of reasons, and perhaps you can think of more than just those on the next slide. Some of the reasons are good reasons for another queue manager, some are not such good reasons, and some are foisted on us whether we think they are good or not!

T

- We're also going to look at answering the question, when DO I need a new queue manager/broker? Some of the good reasons to have another queue manager are of course relevant for that.

E

S

Multiple Queue Managers/Brokers

- Capacity
 - Although some users do not use their queue managers/brokers enough!
 - Peak periods
- High availability
 - Keeping the service provided by your applications available even with an outage of one machine
- Localisation
 - Queue managers/brokers near the applications
 - Government regulation that data can't leave the country
- Isolation
 - One application can't impact another application
 - Maintenance windows, outages
 - Sometimes forced through regulatory requirements such as PCI-DSS
 - Dedicated environments for each client
 - Different MQ/MB requirements – Multi-version helps
- Span-of-control
 - I want my own queue manager/broker!

Complete your sessions evaluation online at SHARE.org/BostonEval

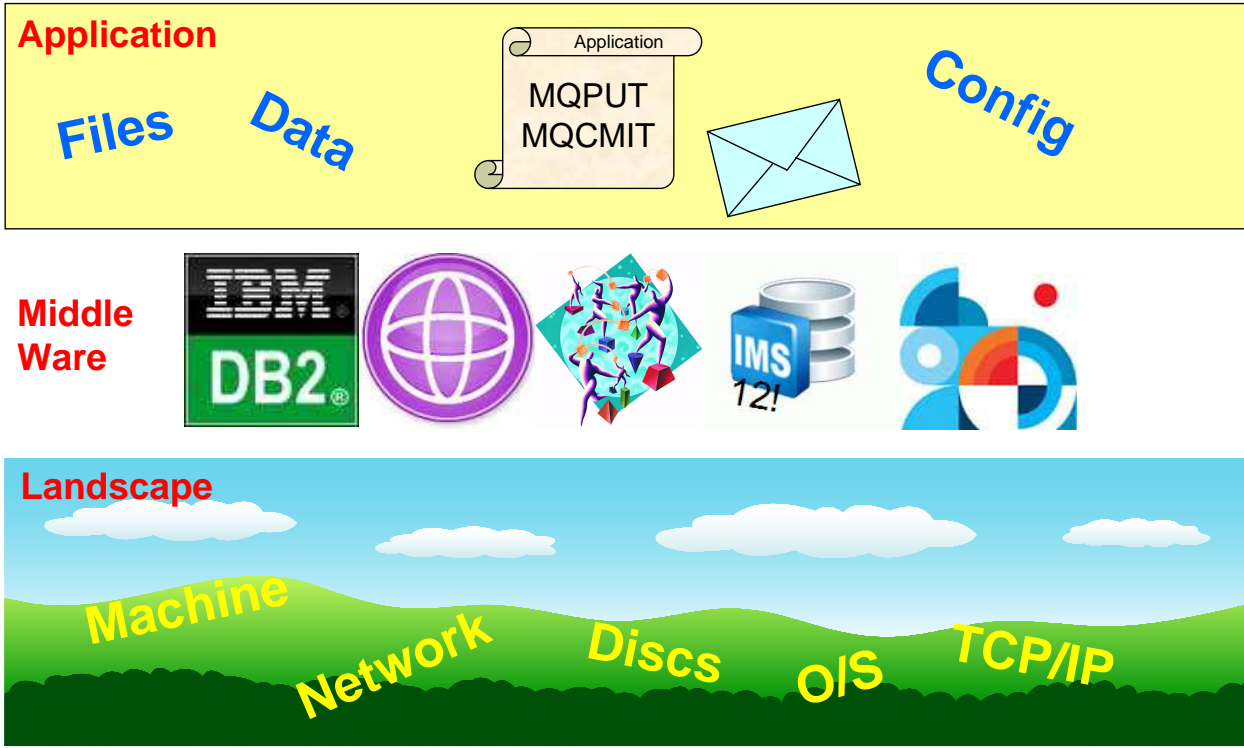
Multiple Queue Managers/Brokers - Notes

N
O
T
E
S

- First there is perhaps the most obvious reason and that is capacity. One queue manager simply cannot go on forever holding the queues for more and more applications. Eventually the resources on the machine will be exhausted and you need to have more queue managers to cope. It is worth noting at this point that some users of WebSphere MQ have gone to the other extreme though, and are not loading up their queue manager's enough.
- Of course, peak period capacity requirements, that is having enough capacity to cope with the workload at peak times, can mean that during off-peak times the capacity is much less utilised, leading to queue managers which are often idle.
- Another very good reason is for availability. Having more than one single point means that there is redundancy in the system, and if one queue manager or machine is unavailable, other queue managers in the system are able to carry on running your work.
- Localisation is also an often quoted reason for having more queue managers. That is, ensuring that there is a queue manager geographically close to where the applications run.
- Isolation of one application from another is sometimes a reason for multiple queue managers, whether ensuring things like a full DLQ from one application doesn't impact another, to any regulatory requirements for separation.
- And finally, another common, but perhaps not so well advised reason is the parochial-ness that occurs in some departments who want to own and run their own queue manager even when there is a department who's job it is to do this for them. Although of course, when the span-of-control differs, this can be advised.

Queue Managers

A shared resource

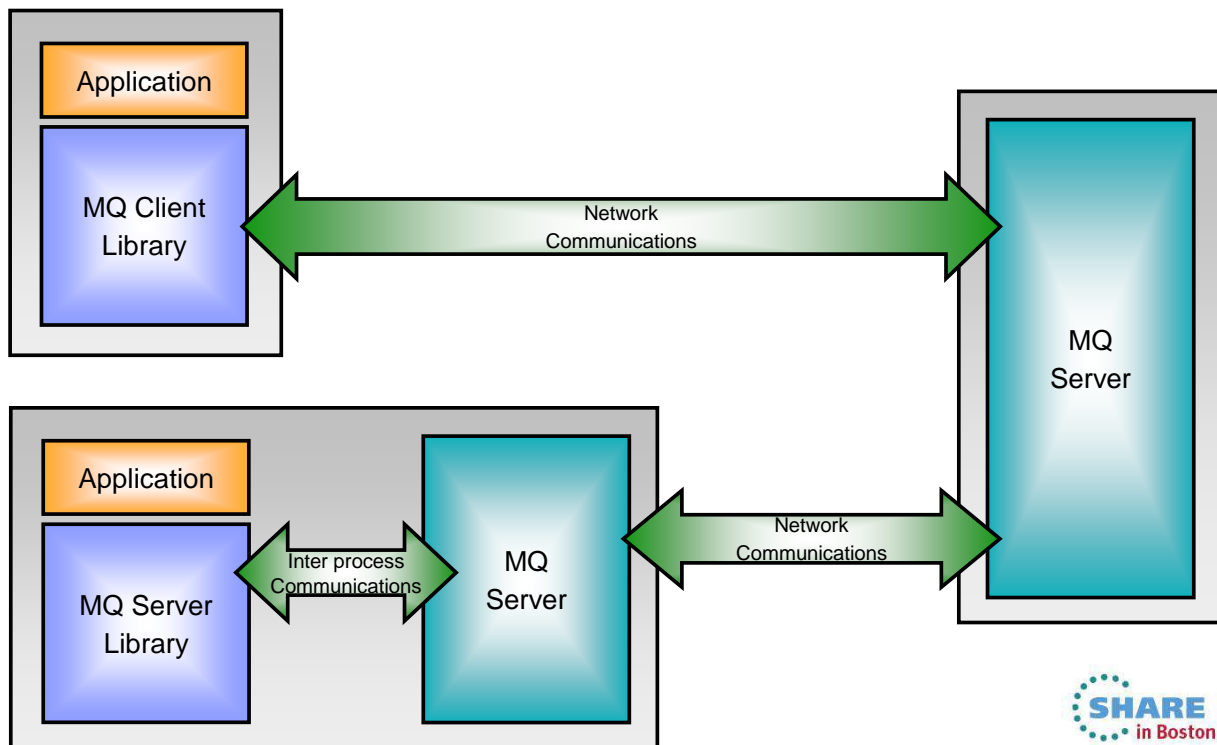


A shared resource - Notes

N
O
T
E
S

- The main message we'd like to get across here is that a queue manager is a shared resource. It is designed to be used by multiple applications. The queue, not the queue manager is designed as the isolation point for applications.
- When creating a new application, there are various resources that you need, a machine, with an operating system, with network connectivity and discs. I refer to these things as the landscape. Then there are application resources, data, maybe in files, a queue, some configuration.
- Then there is middleware, your application hosting environments such as CICS, IMS, or WAS; your database subsystems such as DB2 or IMS; and your queue managers with WebSphere MQ. These middleware resources are intended to be shared resources, much like the landscape, and not part of the application.
- This naturally means the ownership of the queue manager is different too, it is managed not by the application team but by MQ specialists – that's probably you if you're listening to this presentation!

Localisation



Localisation - Notes

N
O
T
E
S

- Localisation is also an often quoted reason for having more queue managers. That is, ensuring that there is a queue manager geographically close to where the applications run.
- There are pros and cons with running an application remotely connecting to a more centralised queue manager versus having a dedicated locally hosted queue manager which are shown on the next slide.

Client vs Local

Remote Client Application

- Pros
 - Lower cost to deploy, only pay for central Server
 - Central operation of Queue Manager
 - Zero local admin except for initial set-up
 - Applications can connect to any remote queue manager instance
 - Connections have failure resilience with automatic reconnect
- Cons
 - No network, no queuing
 - Authentication, e.g. SSL required

Complete your sessions evaluation online at SHARE.org/BostonEval

Local Queue Manager

- Cons
 - License Cost for each queue manager
 - Application team manage Queue Manager
 - Local control over administration and monitoring
 - Applications can only connect to local queue manager
 - No failover when local queue manager is unavailable
- Pros
 - No network, messages are queued locally
 - O/S authenticates user



Client vs Local - Notes

N

O

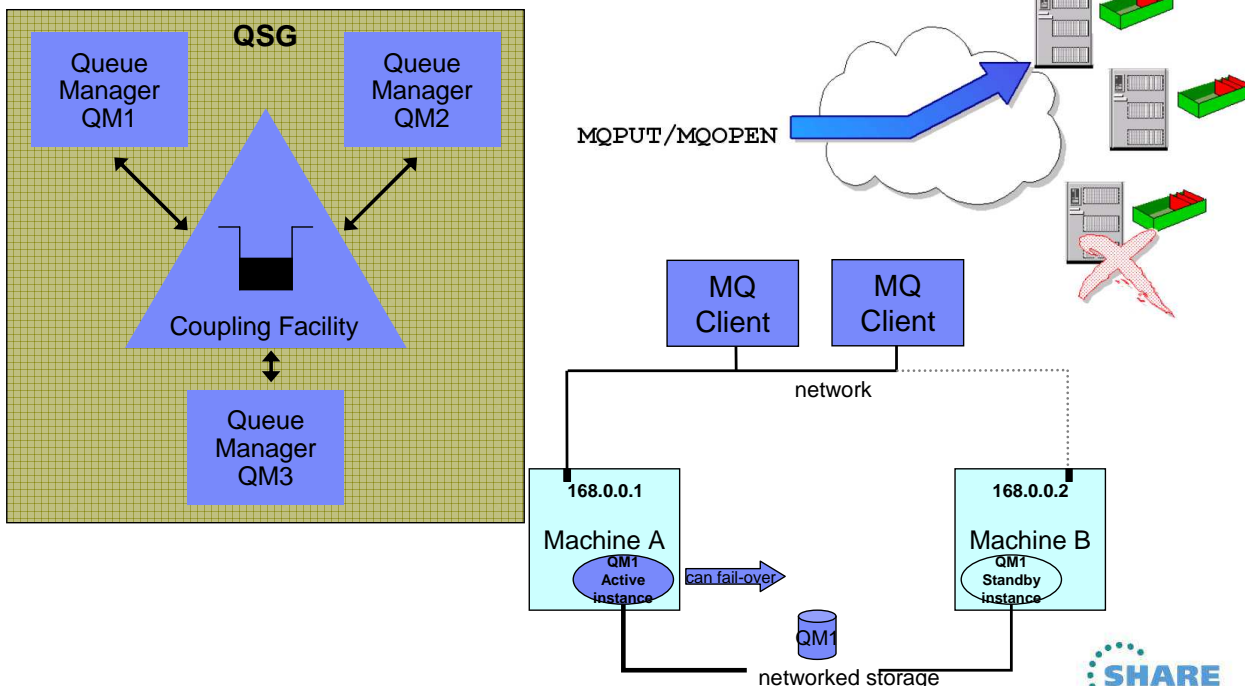
T

E

S

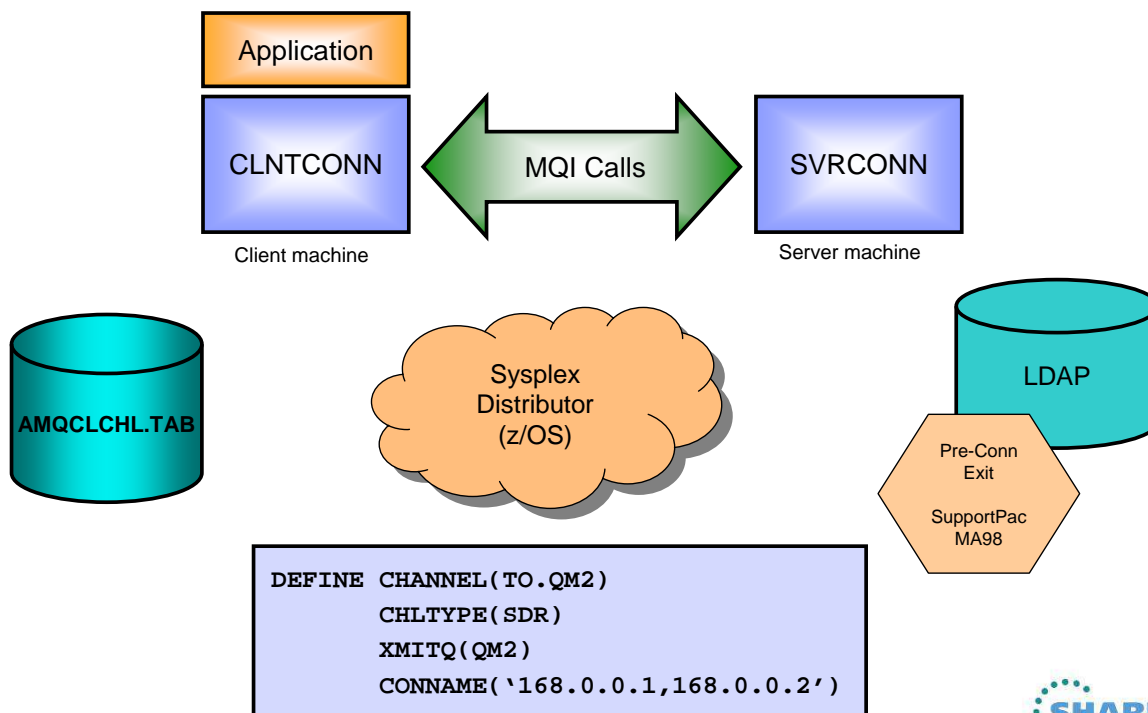
- Here are a selection of points to be aware of when selecting whether an application needs to have a local queue manager or whether using client connectivity to a queue manager elsewhere will suffice. There is no one correct answer of course.

High Availability



Complete your sessions evaluation online at SHARE.org/BostonEval

Client connectivity availability



Complete your sessions evaluation online at SHARE.org/BostonEval

Sessions this week

Session Number	Title
13926	MQ Parallel Sysplex Exploitation, Getting the Best Availability From MQ on z/OS by Using Shared Queues [z/OS]
13929	What's available in MQ and Broker for high availability and disaster recovery? [z/OS & Distributed]
13925	MQ Clustering - The basics, advances and what's new [z/OS & Distributed]

Complete your sessions evaluation online at SHARE.org/BostonEval

High Availability - Notes

N

- A very good reason for having multiple queue managers is for availability. Having more than one single point means that there is redundancy in the system, and if one queue manager or machine is unavailable, other queue managers in the system are able to carry on running your work.

O

- There are a number of availability solutions on WebSphere MQ, different topologies that you can use to arrange your queue manager for highest availability - on the chart here we are reminded of a few examples of these, Queue Sharing Groups (z/OS); Multi-Instance Queue Managers (Distributed); Queue Manager Clustering. Also there are a number of techniques to provide your clients with highest availability, Client channel definition table; Pre-Connect exit; Comma-separated list in CONNAME; Connection Sprayer e.g. Sysplex Distributor (z/OS).

T

E

- Various sessions this week have covered these topics in detail, so we will not be doing that in this session. We just leave you with the message that there are very good reasons to have extra queue managers.

S

How much redundancy?

- Allow for peaks and outages
- Plan to have more queue managers than you need
 - Assume one is always down
- If one queue manager is overloaded split it into 3

Complete your sessions evaluation online at [SHARE.org/BostonEval](https://www.share.org/BostonEval)

How much redundancy? - Notes

N

- So we are looking at two things in this session, when can I consolidate and when do I need another queue manager. Another factor to consider is redundancy. You need to have enough queue managers to cope with peaks of workload and to cope with outages of queue managers.

O

- The plan is to have enough queue managers so that even if one queue manager is currently unavailable you still have the required redundancy.

T

- If you get a queue manager that is overloaded and you need to split that queue manager, split it into three, to provide the same redundancy goal.

E

S

Capacity

- Response time of persistent messages
- Too much data being logged
 - Restart time too long after failure
- z/OS: Buffer pools filling up
- z/OS: Page set capacity
- Number of channel connections
- Amount of data from remote applications

Complete your sessions evaluation online at [SHARE.org/BostonEval](https://www.share.org/BostonEval)

Capacity - Notes

N

- When thinking about whether to consolidate queue managers or whether to need to put a new queue manager in place, capacity is going to be a factor in that decision, but as we've seen, not the only factor.

O

- This slide gives a few examples of the sorts of things that can be a factor in the capacity of your queue manager. Using Accounting and statistics can help you to understand whether you are affected by any of these.

T

E

S

Notes

N
O
T
E
S

- Response time of persistent messages
- Persistent message response time is mainly due to disc response times. This is true both for z/OS and Distributed. Also of course, if you become CPU bound then this will affect response times of both persistent and non-persistent messages. This is a problem to be particularly aware of on the Distributed platforms.
- Some z/OS specific notes:-
 - You can see this in statistics against the commit. You can improve this by improving your DASD. Traditional 3390s have several downsides, Only one z/OS image can write to a volume at a time; I/O request from jobs for same device is queued within z/OS; Data Set placement is critical so you need to ensure things like dual logs are on different devices, and separate log n and log n+1 onto different devices.
 - In comparison modern DASD such as RAID or SHARK has a different architecture with no volume to disk mapping. Therefore, the rewrite can go to different disk. It allows concurrent writing to disk from different images. This means that Data Set placement not so critical. Using Parallel Access Volume z/OS can (read and) write concurrently to different location on same volume which eliminates queuing on z/OS image.
 - Use of VSAM striping may also improve performance in this area. With VSAM striping writing 4 consecutive pages maps one page to 4 volumes, so time to transfer data is less. This can give significant performance benefits for large messages. Archive log cannot exploit this, so may still be a bottleneck. You should consider using VSAM striping if you have a peak rather than sustained high volume.

Notes

N
O
T
E
S

Too much data being logged

- In general terms, understanding the disc I/O rates that you can get for your logs, and knowing the amount of data you need to be able to log to meet your response times and SLAs is important. If you try to do more messages per second through a queue manager than your discs are capable of your response times are going to increase. You'll of course have different I/O rates between local discs versus network attached discs, and different kinds of discs will give you different rates.
- Some z/OS specific notes:-
 - A typical high logging rate is 75-100MB a second but may be lower with mirrored DASD. Slower DASD means less throughput. Understand how long you can survive if you are unable to archive (for example, tape problems).
 - On a queue manager restart, usually have to read up to 3 checkpoints in the log. The more data to be read, the longer it takes. Transactions that have long units of work can cause longer restart times. We suggest checkpoints every 10-15 minutes. You should measure the restart time and evaluate them against your business requirements.

Notes (z/OS specific)

N
O
T
E
S

Buffer pools filling up

- You should aim to keep short lived messages (for example seconds or maybe minutes) in buffer pool, and avoid buffer pool filling up (try to keep <85% full). However, you'd expect long lived messages to be moved out of the buffer pool to the page set.
- To improve your page set I/O, if you have short lived messages then move that queue to a different buffer pool. If you have multiple queues on one page set, separate out the heavy use queue to its own page set. If your expected I/O rate exceeds DASD rate, need to split the queue, or use different queue manager.

Page set capacity

- How many messages do you want to keep at once - are you storing a video of war and peace in a queue? The impact of a checkpoint on a big buffer pool can impact throughput and response time - depends how far you are up the response time curve.
- Some people already have the maximum number of maximum sized page sets, and have run out of space, perhaps due to a large number of large messages. The solution is to have new queue manager.

Notes

N
O
T
E
S

Number of channel connections

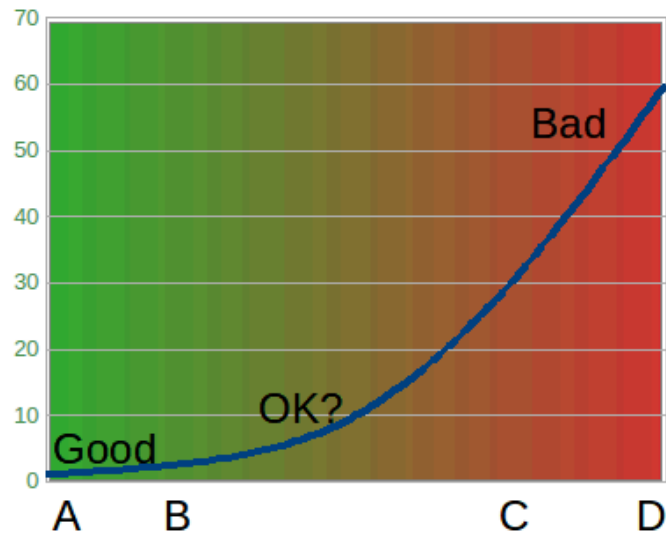
- On distributed, the number of channel connections is ultimately restricted by storage and machine resources such as sockets. Channels run in channel pool processes, and with the majority of distributed platforms providing a 64-bit architecture, this allows for a greater number of channels. In tests in the Lab, we have run 50,000 client channels into a queue manager on AIX. However, that said, it is hard to do anything useful with 50,000 connections that doesn't run out of other machine resources too, and of course, the reconnection times are high if a network outage is suffered, with that many connections. Our advise is not to have more than 10,000 connections into one queue manager.
- On z/OS, the capacity of the CHINIT address space depends on message sizes processed. You can have around 9000 channels when messages sizes are < 32K, less if using SSL. The A/S is restricted to 2GB, so for example, with 100MB messages you can only have around 10 channels.
- Consider a 3 tier architecture with an intermediate queue manager concentrating the connections. Messages will be moved more efficiently in batches by channels, although this will affect the response time back to applications when compared against a 2 tier architecture with applications connected directly.

Amount of data from remote applications

- As noted above, with big messages (100MB) CHINIT can only support a small number of channels. Also, with a lot of data to send, dispatcher TCBs are tied up and may affect other channels running on the same dispatcher. With lots of persistent data Adapter TCBs are tied up due to logging bottlenecks.

Accounting & Statistics

- Two systems in A-B can be merged.
- Two systems at C (or beyond) cannot be merged and keep the same throughput and response time.
- Before considering a merge understand where you are.



Complete your sessions evaluation online at SHARE.org/BostonEval

Accounting & Statistics – Notes

N
O
T
E
S

- When trying to determine whether a queue manager can be merged, or needs to be split, don't forget about accounting and statistics. This can help you to understand the current capacity of your queue manager.

Using Accounting

- Get and put of a message have similar CPU costs
 - Elapsed times may differ – eg logging of put
- Open and close are inexpensive
 - Except for dynamic queues
- Check for indexed access to message
 - Number of gets by msgid or correlid
 - Number of messages skipped
 - Gets taking lots of CPU
- Check time spent logging
 - Usually at commit or backout
 - If logging time and small messages then request was out of syncpoint
 - Large (>100KB) messages delayed during put
- Monitor number of gets involving page set I/O
 - Messages were out on disk
 - Internal read ahead task could not keep up

Complete your sessions evaluation online at SHARE.org/BostonEval

Using Accounting - Notes

N

- There is a whole session on using SMF data this week, so we will not repeat that, but here are a few things that you might discover from using accounting information.

O

T

E

S

Capacity Planning Resources

- SHARE sessions this week
 - 13884: The Dark Side of Monitoring MQ - SMF 115 and 116 Record Reading and Interpretation [z/OS]
 - 13887: WebSphere MQ CHINIT Internals [z/OS]
- Performance Reports
 - MP16: WebSphere MQ for z/OS - Capacity planning & tuning
 - MP1B: WebSphere MQ for z/OS V7.1 - Interpreting accounting and statistics data
 - MP1x: Release specific reports

Complete your sessions evaluation online at SHARE.org/BostonEval

Capacity Planning Resources – Notes

N

- There are a number of resources that you should be familiar with in order to help you with capacity planning. These are listed on this slide.

O

T

E

S

Queue Managers Summary

- A queue manager is a shared resource
- Ensure your system has enough redundancy
- Using Accounting & Statistics to understand your capacity

Complete your sessions evaluation online at [SHARE.org/BostonEval](https://www.share.org/BostonEval)

Queue Managers Summary – Notes

N

▪ This page reminds us of the main messages we want you to take away from this session.

O

▪ Remember that a queue manager is designed to be a shared resource. You do not need a separate queue manager for every application. Of course there will be times when such practice is needed, but it shouldn't be the default position.

T

▪ Redundancy is a big factor in deciding whether you have enough queue managers. Don't consolidate down to the point where you don't have the required redundancy anymore!

E

▪ Capacity will be a big factor in helping you understand whether you have enough queue managers. Make use of Accounting & Statistics to understand your current capacity and whether you need more queue managers or whether you can consolidate some of the ones you already have. Don't forget about point number two though, and always ensure you are not affecting your redundancy.

S

Message Brokers

Agenda



- **Planning your IB deployment topology**
 - Operation modes
 - Execution Groups (servers) vs Additional instances
 - Scaling
 - Deployment with Applications, Libraries and Services
 - Operational environment
 - HA multi instance
 - Global cache
- Understanding and controlling your broker's behaviour
 - Performance statistics
 - Resource Statistics
 - Activity Log
 - Workload Management
 - Other tools
- Making changes

2 Complete your sessions evaluation online at SHARE.org/BostonEval



Why should you care about the topology



- Optimising your broker's performance can lead to real \$\$ savings... Time is Money!
 - Lower hardware requirements, lower MIPS
 - Maximised throughput... improved response time
 - Ensure your SLAs are met
- IBM Integration Bus is a sophisticated product
 - There are often several ways to do things... which is the best?
 - Understanding how the broker works allows you to write more efficient message flows and debug problems more quickly
- It's better to design your message flows to be as efficient as possible from day one
 - Solving bad design later on costs much more
 - Working around bad design can make the problem worse!

Complete your sessions evaluation online at SHARE.org/BostonEval



Operation modes



The operation mode that you use for your broker is determined by the license that you purchase.



The following modes are supported:

- **Developer Edition mode**
 - All features are enabled for evaluation purposes. Limited to 1 message a second per flow.
- **Express Edition mode**
 - Limited number of features are enabled for use within a single execution group. Message flows are unlimited.
- **Scale mode**
 - Limited number of features are enabled. Execution groups and message flows are unlimited.
- **Standard Edition Edition**
 - All features are enabled for use with a single execution group. The number of message flows that you can deploy are unlimited.
- **Advanced mode**
 - All features are enabled and no restrictions or limits are imposed. This mode is the default mode, unless you have the Developer Edition.
- **Remote Adapter Deployment**
 - Only adapter-related features are enabled, and the types of node that you can use, and the number of execution groups that you can create, are limited.

Complete your sessions evaluation online at SHARE.org/BostonEval



Message Flow Deployment Algorithm



**How many Execution Groups (servers) should I have?
How many Additional Instances should I add?**

Execution Groups	Additional Instances
<ul style="list-style-type: none"> • Results in a new process/address-space • Increased memory requirement • Multiple threads including management • Operational simplicity • Gives process level separation • Scales across multiple EGs 	<ul style="list-style-type: none"> • Results in more processing threads • Low(er) memory requirement • Thread level separation • Can share data between threads • Scales across multiple EGs

Recommended Usage
<ul style="list-style-type: none"> • Check resource constraints on system <ul style="list-style-type: none"> • How much memory available? • How many CPUs? • Start low (1 EG, No additional instances) • Group applications in a single EG • Assign heavy resource users to their own EG • Increment EGs and additional instances one at a time <ul style="list-style-type: none"> • Keep checking memory and CPU on machine • Don't assume configuration will work the same on different machines <ul style="list-style-type: none"> • Different memory and number of CPUs

Ultimately have to balance Resource Manageability Availability

Complete your sessions evaluation online at SHARE.org/BostonEval



Consider how your message flows will scale



- How many **execution groups (servers)**?
 - As processes, execution groups provide isolation
 - Higher memory requirement than threads
 - Rules of thumb:
 - One or two execution groups per application
 - Allocate all instances needed for a message flow over those execution groups
 - Assign heavy resource users (memory) to specialist execution groups
- How many **message flow instances**?
 - Message flow instances provide thread level separation
 - Lower memory requirement than execution groups
 - Potential to use shared variables
 - Rules of thumb:
 - There is no pre-set magic number
 - What message rate do you require?
 - Try scaling up the instances to see the effect; to scale effectively, message flows should be efficient, CPU bound, have minimal I/O and have no affinities or serialisation.

Complete your sessions evaluation online at SHARE.org/BostonEval



Deployment with Applications, Libraries and Services



- Develop, deploy and manage your integration solutions using applications, libraries:



Application

- A means of encapsulating resources to solve a specific connectivity problem
- application can reference one or more libraries



Library

- a logical grouping of related routines and/or data
- libraries help with reuse and ease of resource management
- library can reference one or more libraries



Services

- an integration service is a specialized application with a defined interface that acts as a container for a Web Services solution

- These concepts span all aspects of the Toolkit and broker runtime, and are designed to make the development and management of integration solutions easier.

Complete your sessions evaluation online at SHARE.org/BostonEval



Consider your operational environment



- Network topology
 - Distance between broker and data
- What hardware do I need?
 - Including high availability and disaster recovery requirements
 - IBM can help you!
- Transactional behaviour
 - Early availability of messages vs. backout
 - Transactions necessitate log I/O – can change focus from CPU to I/O
 - Message batching (commit count)
- What are your performance requirements?
 - Now... and in five years...

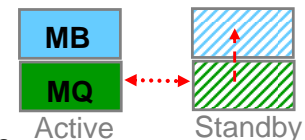
Complete your sessions evaluation online at SHARE.org/BostonEval



HA with Multi-Instance Brokers



- **IIB Exploits MQ Multi-instance queue manager capability**
 - MQ provides basic failover without HA coordinator
 - HACMP, VCS, HA Linux no longer required in many scenarios to restart MQ and MB!
 - MB SAP Input node exploits for state management to give multi-broker and HA SAP support



- **Active and Stand-by Queue Manager and Brokers**
 - Start multiple instances of a queue manager on different machines
 - One is “active” instance; other is “standby” instance
 - Shared data is held in networked storage (NAS, NFS, GPFS) but owned by active instance
 - Automatic MQ Client reconnect will attempt to make failures transparent as possible

V53	Unknown	Unknown	Unknown	Stopped	Unknown
V7	2414	V7_2009-04-21_12.3...	Yes	Running	rockall(Active)
V7B	2415	V7B_2009-04-21_12....	Yes	Running	rockall(Active)
V7C	Unknown	Unknown	Unknown	Running as standby	llareggub(Active), rockall(Standby)

- **IIB Exploitation**

1. Standby broker not running; an MQ service can restart broker once MQ recovery complete
2. Standby broker is running, but not fully initialized until MQ recovery complete

Complete your sessions evaluation online at SHARE.org/BostonEval

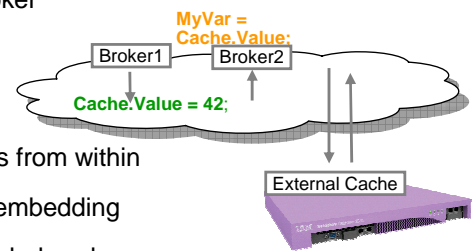


Global Cache



IB contains a built-in facility to share data between multiple brokers

- Improve mediation response times and dramatically reduce application load
- Typical scenarios include multi-broker request-reply and multi-broker aggregation
- Uses WebSphere Extreme Scale coherent cache technology



Support for external software and hardware caches

- Access separate eXtreme Scale and DataPower XC10 appliances from within the broker
- Allows broker to interact with enterprise caching solution without embedding additional libraries
- Cache access, activity log, resource statistics etc. just like embedded cache
- Operationally configured using dynamic configurable service
- New EG options to specify SSL connections to external WXS grids
 - Uses existing MB SSL infrastructure to configure certificates

Cache Expiry options

- New `getGlobalMap()` variant to set the time to live for data in the embedded global cache.
 - `MbGlobalMap evictMap = MbGlobalMap.getGlobalMap("...", new MbGlobalMapSessionPolicy(30));`
 - `evictMap.put("key", "val");`
- Specify a value in seconds. The default value is 0, which means data never gets automatically removed.

Programming and operational enhancements

- Insert and lookup map data using a wider range of Java object types for simplified programming logic
- Support for highly available multi-instance configurations

Clients default to SSL:	<input checked="" type="checkbox"/>
SSL protocol:	SSLv3
SSL key alias:	myKey

10 [Complete your sessions evaluation online at SHARE.org/BostonEval](http://SHARE.org/BostonEval)



Agenda



- Planning your IB deployment topology
- **Understanding and controlling your broker's behaviour**
 - Performance statistics
 - Resource Statistics
 - Activity Log
 - Workload Management
 - Other tools
- Making changes

11 [Complete your sessions evaluation online at SHARE.org/BostonEval](http://SHARE.org/BostonEval)



WebUI Statistics



- Using the WebUI in Integration Bus v9:
 - Control statistics at all levels
 - Easily view and compare flows, helping to understand which are processing the most messages or have the highest elapsed time
 - Easily view and compare nodes, helping to understand which have the highest CPU or elapsed times.
 - View all statistics metrics available for each flow
 - View historical flow data



Coordinated Request Reply MQ Application

Start

Stop

Statistics on

Statistics off

▼ Throughput per message flow for last 15 seconds. Last updated at 09:50:29 GMT Daylight Time.

Flow name	Message Rate (messages/s)	Average Elapsed Time/Invocation (ms)	Average CPU Time/Invocation (ms)
Request	1.00	4.4	0.8
Reply	1.00	3.9	0.5
BackendReplyApp	1.00	1,001.7	1.1

Complete your sessions evaluation online at SHARE.org/BostonEval



Resource Statistics



- View resource statistics for resource managers in IIB such as JVM, ODBC, JDBC etc

name	InitialMemoryInMB	UsedMemoryInMB	CommittedMemoryInMB	MaxMemoryInMB	CumulativeGCTimeInSeconds	CumulativeNumberOfGCCollections
summary	32	41	66	-1	3	277
Heap Memory	32	16	36	256		
Non-Heap Me...	0	25	30	-1		
Garbage Colle...					2	262
Garbage Colle...					1	15

Statistics

User Trace All Flows

Trace Nodes All Flows

Service Trace

Flow Debug Port

Properties...

Start Resource Statistics

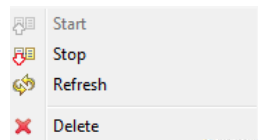
Stop Resource Statistics

View Resource Statistics

Complete your sessions evaluation online at SHARE.org/BostonEval



Activity Log



- View activity as it happens using explorer
- Filter by resource managers

Message ...	Timestamp	RM	MSGFLOW	Message Summary
i BIP11506I	15-Jul-2013 12:50:07.000...		JavaComputeTransform...	Committed a local transaction.
i BIP11501I	15-Jul-2013 12:50:07.000...		JavaComputeTransform...	Received data from input node 'JCTransformNoXPathInput'.
i BIP11506I	15-Jul-2013 12:50:08.000...		JavaComputeTransform...	Committed a local transaction.
i BIP11501I	15-Jul-2013 12:50:08.000...		JavaComputeTransform...	Received data from input node 'JCTransformNoXPathInput'.
i BIP11506I	15-Jul-2013 12:50:09.000...		JavaComputeTransform...	Committed a local transaction.
i BIP11501I	15-Jul-2013 12:50:09.000...		JavaComputeTransform...	Received data from input node 'JCTransformNoXPathInput'.
i BIP11506I	15-Jul-2013 12:50:10.000...		JavaComputeTransform...	Committed a local transaction.
i BIP11501I	15-Jul-2013 12:50:10.000...		JavaComputeTransform...	Received data from input node 'JCTransformNoXPathInput'.
i BIP11506I	15-Jul-2013 12:50:11.000...		JavaComputeTransform...	Committed a local transaction.
i BIP11501I	15-Jul-2013 12:50:11.000...		JavaComputeTransform...	Received data from input node 'JCTransformNoXPathInput'.
i BIP11506I	15-Jul-2013 12:50:13.000...		JavaComputeTransform...	Committed a local transaction.
i BIP11501I	15-Jul-2013 12:50:13.000...		JavaComputeTransform...	Received data from input node 'JCTransformNoXPathInput'.
i BIP11506I	15-Jul-2013 12:50:14.000...		JavaComputeTransform...	Committed a local transaction.
i BIP11501I	15-Jul-2013 12:50:14.000...		JavaComputeTransform...	Received data from input node 'JCTransformNoXPathInput'.
i BIP11506I	15-Jul-2013 12:50:15.000...		JavaComputeTransform...	Committed a local transaction.
i BIP11501I	15-Jul-2013 12:50:15.000...		JavaComputeTransform...	Received data from input node 'JCTransformNoXPathInput'.
i BIP11506I	15-Jul-2013 12:50:16.000...		JavaComputeTransform...	Committed a local transaction.
i BIP11501I	15-Jul-2013 12:50:16.000...		JavaComputeTransform...	Received data from input node 'JCTransformNoXPathInput'.
i BIP11506I	15-Jul-2013 12:50:17.000...		JavaComputeTransform...	Committed a local transaction.
i BIP11501I	15-Jul-2013 12:50:17.000...		JavaComputeTransform...	Received data from input node 'JCTransformNoXPathInput'.
i BIP11506I	15-Jul-2013 12:50:19.000...		JavaComputeTransform...	Committed a local transaction.
i BIP11501I	15-Jul-2013 12:50:19.000...		JavaComputeTransform...	Received data from input node 'JCTransformNoXPathInput'.

Complete your sessions evaluation online at SHARE.org/BostonEval



Activity Log

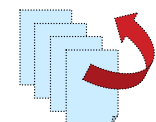


- **New Activity Logging Allows users to understand what a message flow is doing**
 - Complements current extensive product trace by providing end-user oriented trace
 - Can be used by developers, but target is operators and administrators
 - Doesn't require detailed product knowledge to understand behaviour
 - Provides qualitative measure of behaviour
- **End-user oriented with external resource lifecycle**
 - Focus on easily understood actions & resources
 - "GET message queue X", "Update DB table Z"...
 - Complements quantitative resource statistics
- **Flow & resource logging**
 - User can observe all events for a given flow
 - e.g. "GET MQ message", "Send IDOC to SAP", "Commit transaction"...
 - Users can focus on individual resource manager if required
 - e.g. SAP connectivity lost, SAP IDOC processed
 - Use event filters to create custom activity log
 - e.g. capture all activity on JMS queue REQ1 and C:D node CDN1
 - Progressive implementation as with resource statistics, starting with JMS, C:D and SAP resources
- **Comprehensive Reporting Options**
 - Reporting via MB Explorer, log files and programmable management (CMP API)
 - Extensive filtering & search options, also includes save data to CSV file for later analysis

Message...	Timestamp	Message Summary
i BIP12001I	17-Jun-2011 10:10:50.85...	Connected to JMS provider 'WebSphere_MQ'
i BIP12002I	17-Jun-2011 10:10:50.85...	Created a 'Transaction_None' session for JMS provider 'WebSphere_MQ'
i BIP12004I	17-Jun-2011 10:10:50.93...	Created JMS producer for destination 'ASYNCREQUESTQ'
i BIP12007I	17-Jun-2011 10:10:50.93...	Sent a JMS message to queue 'ASYNCREQUESTQ'
i BIP12004I	17-Jun-2011 10:10:50.92...	Created JMS producer for destination 'ASYNCRECEIVEQ'
o BIP12014E	17-Jun-2011 13:47:51.65...	Failed to send message to 'ASYNCRECEIVEQ'
i BIP12001I	17-Jun-2011 13:47:54.99...	Connected to JMS provider 'WebSphere_MQ'
i BIP12004I	17-Jun-2011 13:47:55.00...	Created JMS producer for destination 'ASYNCRECEIVEQ'

- **Log Rotation facilities**
 - Rotate resource log file when reaches using size or time interval

Complete your sessions evaluation online at SHARE.org/BostonEval



Controlling Integrations with Policy



Integration Workload Management

- Provide intelligent mechanisms to control processing speed
- Most common scenario is to reduce back-end server load
- Design allows more policy-based processing over time
- Can be applied to new or existing integration data flows

Policy defines threshold limits and relevant actions

- Set thresholds for integration data flow throughput
- Specify actions at threshold, for example:
 - NOTIFY: Higher (or lower) than threshold generates publication
 - DELAY: Excessive workload will have latency added to shape throughput

Web Console used to manage WLM policy

- Sophisticated behaviour controllable by broker WLM policy
- Workload can be managed across classes of message flows (e.g. batch vs. online)
- Policies stored in local registry, and dynamically configurable
- Developer can also specify limits as integration data flow properties

Values that you do not define on this page are inherited from the message flow, if they are.

Policy Name: BatchWorkloads

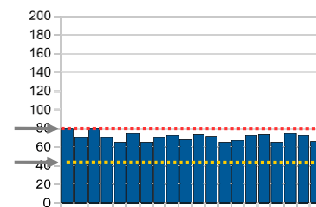
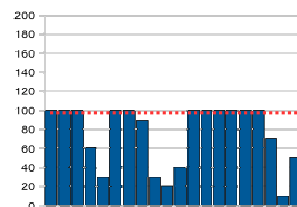
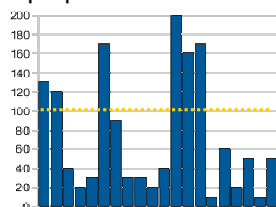
Targets and Limits

Notification Threshold: 100 message

Maximum Rate: 300 message

Additional Instances: 1

Start additional instances when flow starts: Vse



16 Complete your sessions evaluation online at SHARE.org/BostonEval

Some tools to understand your broker's behaviour



- *PerfHarness* – Drive realistic loads through the broker
- OS Tools
 - Run your message flow under load and determine the limiting factor.
 - Is your message flow CPU, memory or I/O bound?
 - e.g. “perfmon” (Windows), “vmstat” or “top” (Linux/UNIX) or “SDSF” (z/OS).
 - This information will help you understand the likely impact of scaling (e.g. additional instances), faster storage and faster networks
- Third Party Tools
 - *RFHUtil* – useful for sending/receiving MQ messages and customising all headers
 - *NetTool* – useful for testing HTTP/SOAP
 - *Filemon* – Windows tool to show which files are in use by which processes
 - *Java Health Center* – to diagnose issues in Java nodes
- MQ Explorer
 - Queue Manager administration
 - Useful to monitor queue depths during tests
- Integration Bus
 - WebUI statistics
 - IIB Explorer for activity log and resource statistics

Complete your sessions evaluation online at SHARE.org/BostonEval



Agenda

- Planning your IB deployment topology
- Understanding and controlling your broker's behaviour
- **Making changes**

Testing and Optimising Message Flows

- Run driver application (e.g. *PerfHarness*) on a machine separate to broker machine
- Test flows in isolation
- Use real data where possible
- Avoid pre-loading queues
- Use a controlled environment – access, code, software, hardware
- Take a baseline before any changes are made
- Make ONE change at a time.
- Where possible retest after each change to measure impact
- Start with a single flow instance/EG and then increase to measure scaling and processing characteristics of your flows

