What's Wrong with MQ?
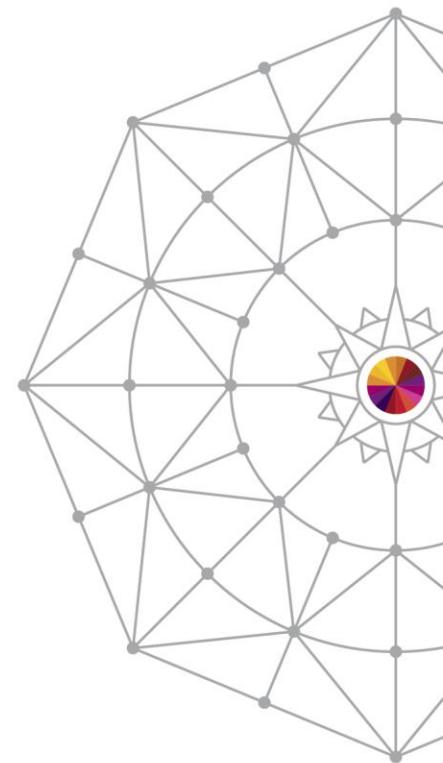
Lee Wheaton
M&T Bank

August 6, 2014
Session 15503

What's wrong with MQ?

- Disclaimer
  - The web sites listed in this presentation belong to their respective owners, and as such, all rights are reserved by these owners. These web sites are provided for informational purposes only and are provided "AS IS" without warranty of any kind. It should not be construed that the author of this presentation or SHARE is endorsing or recommending these websites or the companies and products listed therein. Any visits you make to these web sites are done strictly at your own risk.
  - As of July 18, 2014, these web links were still valid URL's.

What's wrong with MQ?

- At our shop, very little actually.
  - High uptime, integrity and performance with MQ
  - Very low incident rate
  - But even so, we still get asked "the question"

- Being in the middle(ware) when a problem occurs, MQ support tends to get pulled into upstream and downstream application issues even when MQ is purring like a kitten.

  https://t-rob.net/2012/11/02/sure-its-always-an-mq-problem/

- Most of our 'MQ' problems involve events outside of MQ and not MQ itself which is the basis for this session

- Resolution of some of these 'outside' events can be very challenging especially when it is having a production impact.

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval

Session Topics

- Session Scope
- Being Proactive vs. Reactive to problems
- Forensic Tools - MustGather
- Identifying Points of Failure in the MQ Infrastructure
- Case Studies for Production MQ Issues
  - Problems 'r Us
  - Application related
  - Network related
  - Server related
  - Audience Picks (Appendix F)
- Q&A
- Appendices

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval

Session Scope

- Main focus on problem events where MQ is the 'victim' and not the Root Cause
  - Anticipating problems by Being Proactive
  - Actual production case studies reviewed
  - Review troubleshooting used for some of these problems

- This session intended to be a supplement to the many excellent resources already available on identifying MQ specific errors, e.g.
  - See Anaheim/Boston SHARE session abstracts presented by Lyn Elkins, Moraq Hughson, Neil Johnston and others
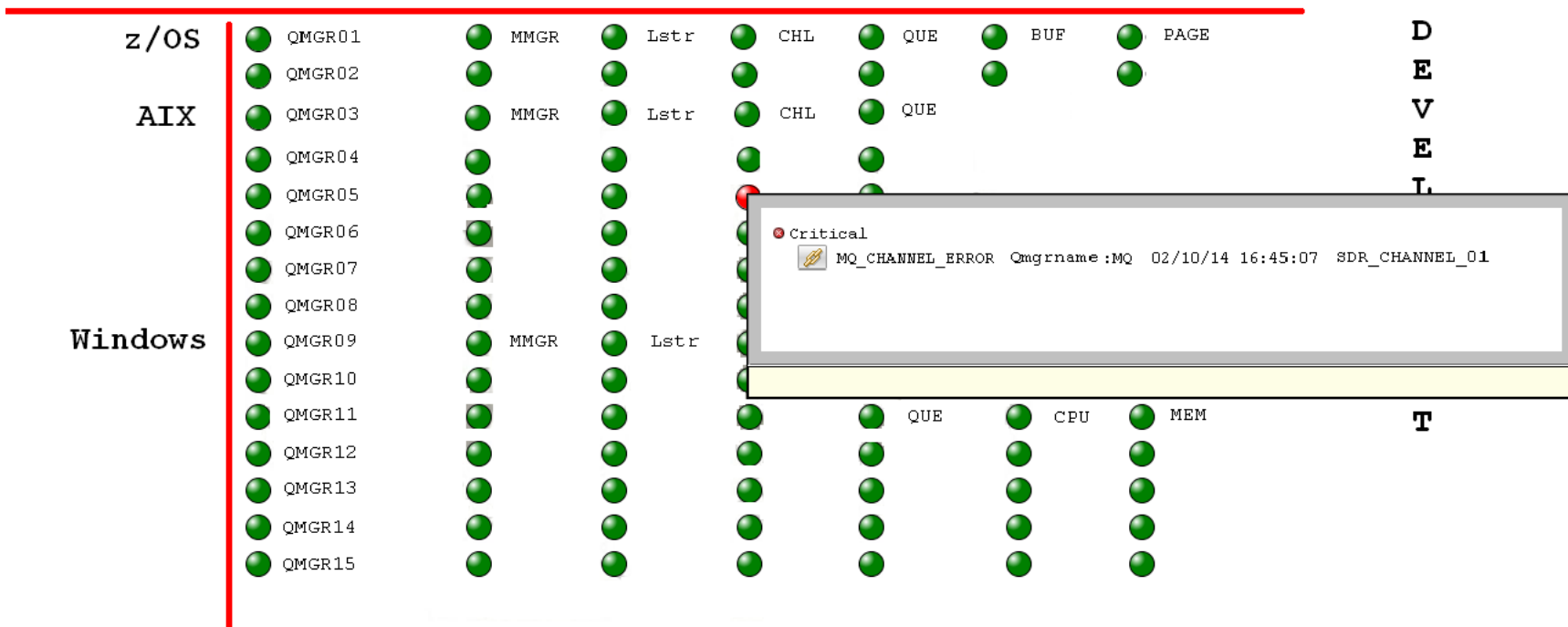
Proactive vs. Reactive

- Obviously, when a problem occurs, we're already in reactive mode. Question is do we continually fly by the seat of our pants or do we employ a standard and planned approach when problems occur?

- No plan = firefighting, extended outages, stress, mistakes.

- Multi-tasking is a Myth. According to a number of studies, the human brain is essentially a uniprocessor with preemptive multitasking. Having a game plan in advance of a crisis will help the support person stay focused.

- What can we do to be proactive by planning for problems before they occur?

- What can we do to minimize outages when they do occur?

Being Proactive

- Follow Websphere MQ 'Best Practices'
  http://www.ibm.com/developerworks/websphere/library/techarticles/0807_hsieh/0807_hsieh.html
  http://www.ibm.com/developerworks/websphere/library/techarticles/0712_dunn/0712_dunn.html
  http://documents.bmc.com/products/documents/62/63/26263/26263/index.htm (TCB info dated)
  http://www.mqtechconference.com/sessions_v2013/WMQ_Best_Practices.pdf
  http://www-01.ibm.com/support/docview.wss?uid=swg24006699

- Know and document your MQ Infrastructure and touch points

- Know IBM's support process, MustGather requirements and be specific, concise, accurate & yet detailed in the PMR title/detail
  http://www-304.ibm.com/webapp/set2/sas/f/handbook/getsupport.html
  http://www-01.ibm.com/support/docview.wss?uid=swg21229861
  http://www-01.ibm.com/software/support/lifecycle/index_w.html
  http://www-01.ibm.com/support/docview.wss?uid=swg21312967

- Of course, be on a IBM supported MQ release and as current as possible. Have application teams perform validation/sign-off.

- Alert IBM Account Rep and IBM duty manager before your major MQ upgrade (See Appendix A for sample Alert form)

Being Proactive (2)

M&T Bank MiddleWare Services

z/OS	QMGR01	MMGR	Lstr	CHL	QUE	BUF	PAGE
	QMGR02						
AIX	QMGR03	MMGR	Lstr	CHL	QUE		
	QMGR04						
	QMGR05						
	QMGR06						
	QMGR07						
	QMGR08						
Windows	QMGR09	MMGR	Lstr				
	QMGR10						
	QMGR11			QUE	CPU	MEM	
	QMGR12						
	QMGR13						
	QMGR14						
	QMGR15						

> Critical
> MQ_CHANNEL_ERROR Qmgrname:MQ 02/10/14 16:45:07 SDR_CHANNEL_01

One example of a MQ Monitoring Dashboard for Administrators

MQ monitoring software with histograms & a dashboard view are a must.
MQ by Line of Business Dashboards/Portals can be very powerful tools.

Being Proactive (3)

- Interpret FFST and TRC files & DLH/MQMD/GMO/PMO/Others
  http://hursleyonwmq.wordpress.com/2007/05/04/introduction-to-ffsts/
  http://www-01.ibm.com/support/docview.wss?uid=swg21174924
  See section on Data Type descriptions in MQ Application Programming Reference

- Obtain latest info by attending SHARE, IBM training, IMPACT or Capitalware MQTC as well as subscribe to IBM and MQ listservers and newsletters. And of course,
  http://www-947.ibm.com/support/entry/portal/overview/software/websphere/websphere_mq

- Maintain problem history file and a reference file for all APARs, flashes etc. for quick keyword searches when a problem occurs

- Use IBM's Support Assistant (see Boston/Anaheim abstracts)
  http://www-01.ibm.com/support/docview.wss?uid=swg27012682
  http://www-01.ibm.com/support/docview.wss?uid=swg21624944 (alternative solution)

- Monitor change management notifications as well as problem tickets in your company for applicability to the MQ infrastructure

Being Proactive (4)

- Employ/use change management standards for MQ migrations

- Periodically perform MQ health checks
  Run default tests on each queue manager via MQ Explorer and review test results. Also see MO71.
  http://hursleyonwmq.wordpress.com/2007/02/19/customizing-websphere-mq-explorer-to-warn-you-about-the-things-you-care-about/
  https://www-950.ibm.com/events/wwe/grp/grp004.nsf/vLookupPDFs/WMQ%20Services%20Healthcheck/$file/WMQ%20Services%20Healthcheck.pdf
  See Appendix D

- Deliberately create and document certain MQ problems and their resolution (in test of course)
  e.g. Remote server crashes and how do I redirect messages in XMITQ to another remote server
      Place a channel indoubt and see what needs to be done to resolve it without deleting or
      duplicating messages

- Do have a game plan in place to execute when problems do occur. Have relevant contact info readily at your disposal.

- See Appendix E for more information on Being Proactive

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval

Forensic Tools - MustGather

Research
- Search Google/Bing
- Search www.ibm.com
- IBM Support Assistant
- MQ blog/listserver/dvlpworks
- Other resources
  - AMQERR0?.LOG, FFST
  - O/S SYSLOG/Event log
  - Monitor/CPU/Memory/disk
  - /log/*qmgr*/active/S00?.LOG
  - Data mine histograms

MQ Tools (partial list)
- dspmq, dspmqver
- strmqtrc, qmgr actvtrc()
- dspmqrte, amqsreq(c), put, get, rfhutil?
- monq()-msgage, qtime monch()-xqtime, nettime
- channel ping, chstatus, qstatus, conn
- IBM MQ SupportPacs

Network
- telnet dns or ip port#
- netstat -a -b -n -o –s
- ping dns or ip -n -l –f
- tracert(e) dns or ip
- nslookup dns or ip
- pathping dns or ip
- Network 'sniffer' trace
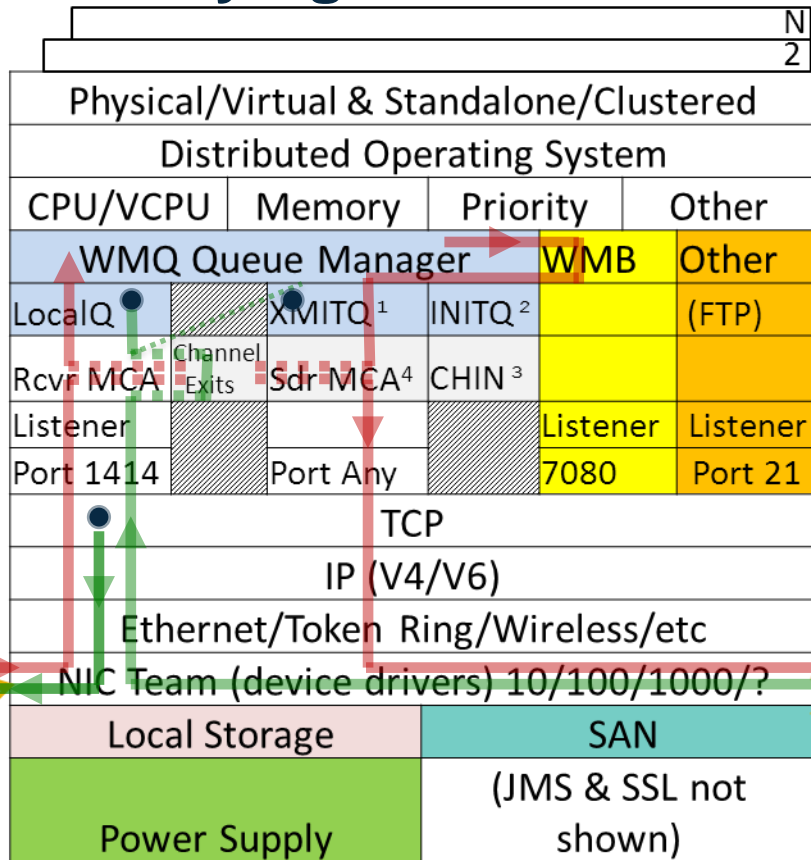- Wireshark/Netmon/tcpdump

Other (partial list)
- Build Diagnostic Scripts
  - bat or cmd
  - shell script
  - Rexx/clist
- Microsoft Regmon
- Microsoft Procmon
- O/S snap dumps
- z/OS slip traps via IBM
- SNMP event log traps ?
- Abend-Aid

- Windows WER Dump
- VMWare Vsphere
- RMF monitor (z/os)
- SMF reporting (z/os)
- Windows MustGather hyperlink
- Windows dump via LiveKD

Identifying Points of Failure in the MQ Infrastructure

Left diagram (Distributed):

N	
2	

Physical/Virtual & Standalone/Clustered			
Distributed Operating System			
CPU/VCPU	Memory	Priority	Other
WMQ Queue Manager		WMB	Other
LocalQ	XMITQ [1]	INITQ [2]	(FTP)
Rcvr MCA	Channel Exits	Sdr MCA [4]	CHIN [3]	
Listener			Listener	Listener
Port 1414	Port Any	7080	Port 21
TCP			
IP (V4/V6)			
Ethernet/Token Ring/Wireless/etc			
NIC Team (device drivers) 10/100/1000/?			
Local Storage	SAN		
Power Supply	(JMS & SSL not shown)		

Right diagram (z/OS):

N	
2	

Standalone/Clustered/Sysplex			
z/OS Operating System			
CPU	Memory	WLM	Other
WMQ Queue Manager 1-n		CICS 1-n	Other
LocalQ	XMITQ [1]	INITQ [2]	(Web)	(FTP)
Rcvr MCA	Channel Exits	Sdr MCA [4]	CHIN [3]	(Java)	(WMB)
Listener			Listener	(Unix)
Port 1414	Port Any	Port n	Port n
TCP**			
IP (V4/V6)**			
Ethernet/Token Ring			
OSA Team 1gb/10gb/?			
Local Storage	SAN		
Power Supply	(JMS & SSL not shown)		

* - Sender channel startup sequence: 1) message is written to the transmit queue, 2) which may result in a trigger message being put on the system.channel.initq, 3) the channel initiator supervisor picks up the trigger message, and 4) starts the Sender channel that was specified in the transmit queue trigger definition. The sender MCA services the messages on the transmit queue.
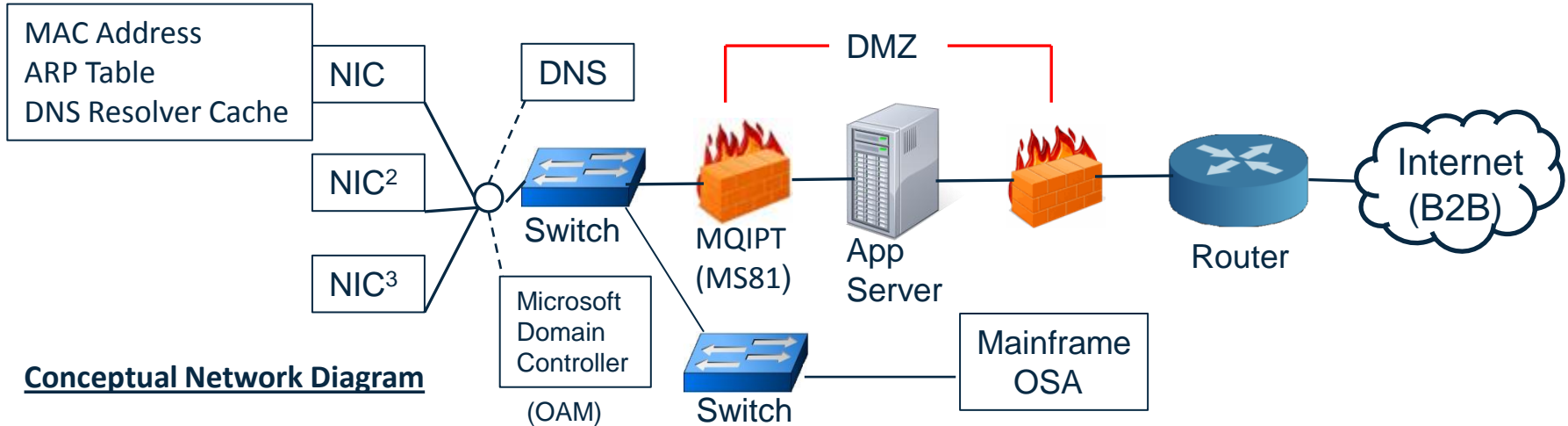
**- On z/OS, the TCP/IP protocol stack is a component of the z/OS Communications Server which is not shown in the diagram. Also of note is the Communications Storage Manager component which provides the I/O buffer area for TCP/IP traffic.

Identifying Points of Failure in the MQ Infrastructure (2a)

Left diagram (Distributed):

❸ N/2

❹ Physical/Virtual & Standalone/Clustered			
❺ Distributed Operating System			

❻ CPU/VCPU	Memory	Priority	Other

❼ WMQ Queue Manager			WMB ❼ Other

LocalQ		XMITQ [1]	INITQ [2]		(FTP)
Rcvr MCA	Channel Exits	Sdr MCA[4]	CHIN [3]		
Listener				Listener	Listener
❽ Port 1414		Port Any		7080 ❽	Port 21

| TCP |
| ❾ IP (V4/V6) |
| ❾ Ethernet/Token Ring/Wireless/etc |
| ❾ NIC Team (device drivers) 10/100/1000/? |

| ❷ Local Storage | SAN |
| ❶ Power Supply | (JMS & SSL not shown) |

Right diagram (z/OS):

N/2

Standalone/Clustered/Sysplex			
z/OS Operating System			

CPU	Memory	WLM	Other

WMQ Queue Manager 1-n			CICS 1-n	Other

LocalQ		XMITQ [1]	INITQ [2]	(Web)	(FTP)
Rcvr MCA	Channel Exits	Sdr MCA[4]	CHIN [3]	(Java)	(WMB)
Listener				Listener	(Unix)
Port 1414		Port Any		Port n	Port n

| TCP** |
| IP (V4/V6)** |
| Ethernet/Token Ring |
| OSA Team 1gb/10gb/? |

❿

| Local Storage | SAN |
| Power Supply | (JMS & SSL not shown) |

* - Sender channel startup sequence: 1) message is written to the transmit queue, 2) which may result in a trigger message being put on the system.channel.initq, 3) the channel initiator supervisor picks up the trigger message, and 4) starts the Sender channel that was specified in the transmit queue trigger definition. The sender MCA services the messages on the transmit queue.

**- On z/OS, the TCP/IP protocol stack is a component of the z/OS Communications Server which is not shown in the diagram. Also of note is the Communications Storage Manager component which provides the I/O buffer area for TCP/IP traffic.

SHARE in Pittsburgh 2014

Identifying Points of Failure in the MQ Infrastructure (2b)

Legend for Conceptual Diagram of MQ Infrastructure (slide 13):

❶ Power supply: good connections, regulated/fluctuations/surges, generator backup?

❷ Storage media: OS and/or MQ on Local storage and/or SAN? Latency? Contention? Media Failure/Recovery?

❸ Scalability/Failover: ~~Standalone~~, Clustered, Sysplex? How do servers interact and affect each other? What components are used or shared? How do you define failover and your recovery time objective?

❹ Physical/Virtual: Standalone=outage? Virtual adds another layer to your OS plus there are other OS's competing for the Virtual Hosts' resources. Both physical and virtual can have hardware issues.

❺ Operating System: What version and are subordinate components compatible? Vendor supported? Patches needed?

❻ CPU/Memory/WLM: How many CPUs and how much memory installed? Can they support the existing and future workload? Hardware failure? Is WLM/priorities affecting MQ performance?

❼ Applications/Services/Regions: Multiple applications/services/regions are competing for the same host resources and could impact MQ. Downstream issues?

❽ Port Conflicts: Beware of potential port conflicts. See www.iana.org .

❾ TCP/IP or NIC: Can have software/hardware issues. Garden hose effect. More later.

❿ Network: Subnet/network contention? Device Failure? More later.

Identifying Points of Failure in the MQ Infrastructure (3)

MAC Address
ARP Table
DNS Resolver Cache

NIC

NIC^2

NIC^3

DNS

Switch

Microsoft
Domain
Controller

(OAM)

Switch

Conceptual Network Diagram

DMZ

MQIPT
(MS81)

App
Server

Mainframe
OSA

Router

Internet
(B2B)

- When a Network component(s) becomes congested, it can result in queuing delays, dropped packets or blocking of new connections as well as an increase in retransmissions from the originating source (i.e. server). Loose cable connections can also cause an issue.
- MQ messages greater in length than the Maximum Transmission Unit (MTU) will be broken up into two or more Network packets. Use of MQ compression could reduce the number of Network packets created but with some processing overhead on the sending/receiving side to compress/decompress. Mismatches in MTU settings between Network devices can result in fragmentation and additional retransmissions, or a black hole router condition.

http://www.windowsnetworking.com/articles-tutorials/trouble/TCPIP-Troubleshooting-Structured-Approach-Part1.html

Identifying Points of Failure in the MQ Infrastructure (4)
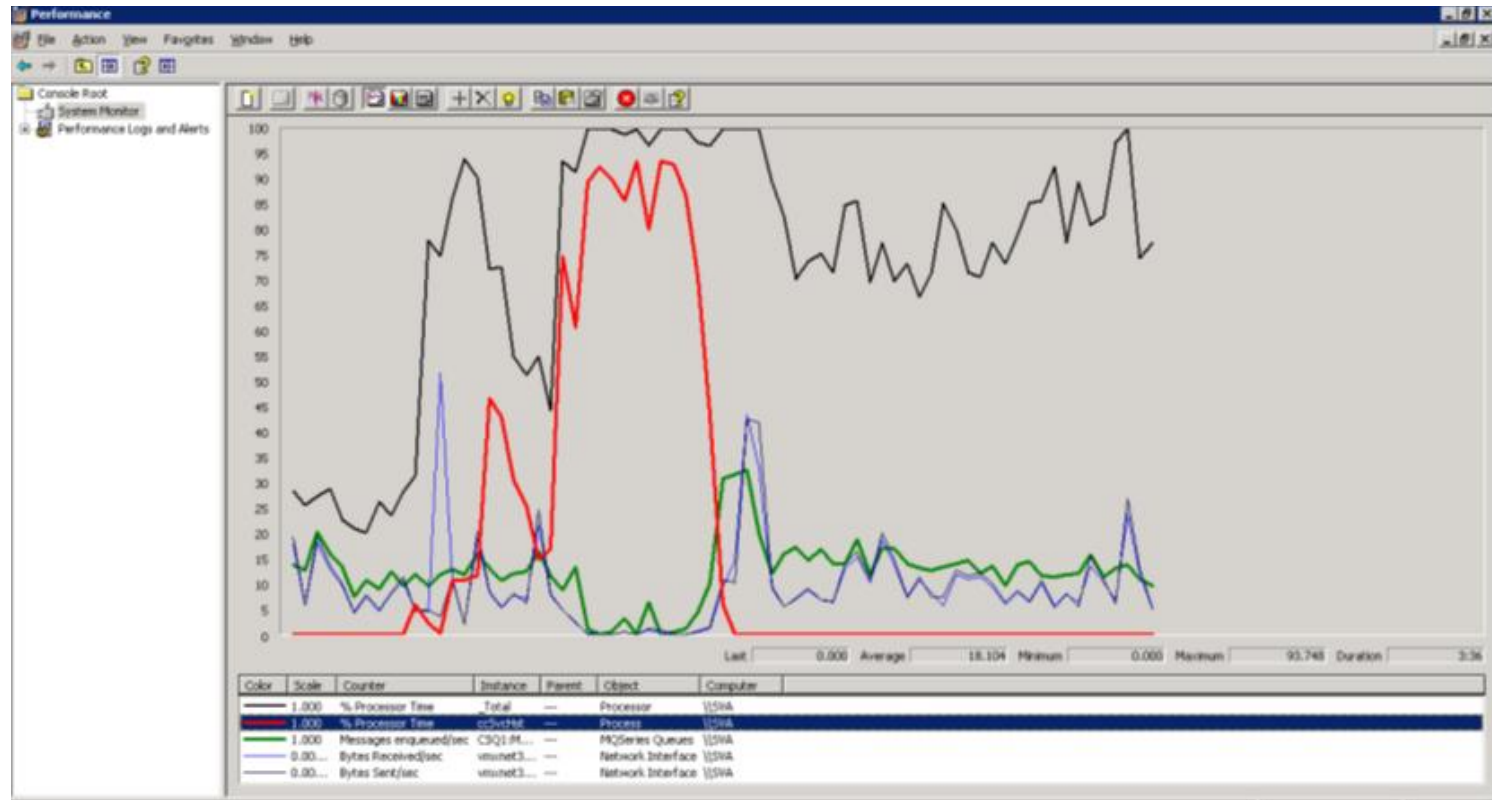
- CPU spike (red) in Virus Scan service & drop-off in MQ/Network traffic
  - Client app response time increases
  - Windows server MQ nettime 18 sec.
  - Mainframe MQ nettime 9 sec.
  - XMITQ buildup on both sides
  - Telecom sees no network latency
  - Mainframe MQ traffic drops off

CASE STUDIES

Case Studies for Production MQ Issues

- A brief summary of actual production MQ issues
  - symptoms/error messages
  - resolution
  - broken up into four sections:
    - Problems 'r Us
    - Application related
    - Network related
    - Server related

- For most issues, MQ was the 'victim' and not the Root Cause

- See Appendix F for additional case studies

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval

Case Studies: Problems 'r Us

This section centers on problems that fall in our MQA jurisdiction.

- CICS-MQ program getting MQ reason 2111 (CCSID error) due to SCSQANLE library not being defined to the CICS region

- New features/opportunities encountered in MQ 7.x
  - sharecnv setting on SVRCONN causing connection drops & other
    Note-also see connection factory sharecnv (y/n) setting
    http://www.mqtechconference.com/sessions_v2013/MQTCMQClient.pdf

  - connection factory default extended polling interval causing 5+ second delay in MQ JMS responses going back to the client

  - Suspect destinations consumers Allow Read Ahead causing connection drops under load & has potential to lose non-persistent msgs

  - MQGET 2119 error due to CCSID conversion change from MQ JMS client to MQ server in MQ 7.x (some behavior revised in 7.0.1.6)
    http://webspheremq.wordpress.com/2012/02/08/amq6047-from-websphere-mq-v7-server-and-jmswmq2002-reason-2119-from-websphere-mq-v7-client/
    http://www-01.ibm.com/support/docview.wss?uid=swg21222078

Case Studies: Application

At our shop, a portion of our MQA time is spent supporting developer related issues. Examples:

- 'Lost' MQ message due to app MQGET nosync & subsequent abend
  - id w/ application logging, MA0W, actvtrc, MQ trace, 3rd party audit package, MQ and/or OS event logs

- Developer using insufficient MQ error handling resulting in loop
  - Distributed app got unexpected condition and went into transaction retry loop resulting in significantly high cpu on the mainframe & logfull

- Queue depth buildups/backups due to downstream servicing application resource issues (e.g. abends, deadlocks/deadly embrace, enqueues on common shared resources resulting in processing serialization, wait events, CICS region consistently at max tasks, insufficient or ineffective servicing bees/threads, mixing short & long running messages, un-tuned/elongated message/txn servicing times)

Case Studies: Application (2)

- 'Lost' MQ messages because app team prematurely activated a pre-production server resulting in messages being pulled from production

- Developer did not commit their MQPUT of the request and receives 2033 (timeout) on their MQGET w/ wait on the response

- Application initiating more MQ connections/workload than it can handle (in conjunction with other processes/connections e.g. database) resulting in high cpu/memory on app server and broken MQ connections

- Client application leaving orphaned processes with MQ connections still intact, and/or 'keepalive' not being used
  - http://www-01.ibm.com/support/docview.wss?uid=swg21232484
  - http://www-01.ibm.com/support/docview.wss?uid=swg21177012
  - http://www.ibm.com/developerworks/websphere/library/techarticles/0710_titheridge/0710_titheridge.html (fyi)

Case Studies: Network

- Network articles:
  http://www-01.ibm.com/support/docview.wss?uid=swg21237211
  https://www.ibm.com/developerworks/community/blogs/aimsupport/entry/websphere_mq_channels_are_we_really_just_the_messenger?lang=en_us
  http://cpacket.com/wp-content/files_mf/introductiontonetworklatencyengineering.pdf

- Reason codes 2059 or 2009 due to firewall failing over, firewall rules push, firewall cpu/load at or near 100%, or incorrect firewall rules

- Mainframe OSA adapter at/or near 100% dropping network packets or OSA adapter failover causing channel ECONNRESET

- Bulk data transfer (FTP/backups) maxing out same network segments shared by your MQ traffic resulting in dropped packets and broken MQ connections. Use Network sniffers & monitors. Consider QoS for MQ.

- Mainframe TCP/IP priority not set high enough causing channel issues

- Sporadic 10054 connection drops between MQ z/OS and distributed MQ due to OSPF routing task on the mainframe not getting serviced. Dispatching priority of OMPROUTE was increased.
  http://www-01.ibm.com/support/docview.wss?uid=isg1II12026

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval

Case Studies: Network (2)

- DNS resolution failure or DNS unreachable resulting in MQCONN or Sender channel error
  - Used when DNS name entered on channel conname

  - ALSO used when IP address entered on sender channel conname
    https://www.ibm.com/developerworks/community/blogs/aimsupport/entry/websphere_mq_channels_waiting_on_domain_name_server?lang=en

  - Display CHSTATUS shows state(binding) & substate(nameserver)

  - Use nslookup to see if DNS resolves IP/DNS properly & triangulate
    http://www-01.ibm.com/support/docview.wss?uid=swg27024309
    http://technet.microsoft.com/en-us/library/cc731991.aspx

Case Studies: Server

- 2003 errors (MQRC_BACKED_OUT) and poor response time when MQ z/OS MSTR and/or CHIN regions are at low processing priority
  http://www-01.ibm.com/support/docview.wss?uid=swg21171730

- Bulk data transfers (FTP/backups) on the same server as the MQ client impacting MQ connections and the business application

- AMQ8561 or 2063 due to domain controller being unavailable for (Windows) OAM access check

- AMQ7227 on OAM access check when MQ Client in domain x and MQ server in domain y and domain x does not trust domain y. (You may also see RPC call failed error message because MQ server does not have MQ clients domain x defined to its LAN advanced TCPIP setting).
  https://www.ibm.com/developerworks/community/blogs/aimsupport/entry/websphere_mq_domain_user_security?lang=en

- MQ queue manager comes down when server backup job enqueues on MQ checkpoint file. AMQ6119 CreateFile. Search on pmr22036. Rescheduled server backup job to off-peak time.

- Mainframe cpu maxed out resulting in qdepth buildups due to servicing apps slow down

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval

Case Studies: Server (2)

- SAN storage issue (if server o/s or MQ deployed on the SAN)
  - See if you can access your server or MQ related storage
  - Check for gaps in server/MQ log times for SAN latency
  - Lock on a particular SAN volume on the storage array that prevents other hosts from connecting. MQ may seem to hang.
  - Review your platform's monitor for disk i/o latency
    http://blogs.technet.com/b/askcore/archive/2012/02/07/measuring-disk-latency-with-windows-performance-monitor-perfmon.aspx

- Client App server applied Anti-Virus/Security definitions resulting in broken MQ connections. Changes backed out which cleared it up.

- Application server not right sized and snapping MQ connections
  - Upon going to a bigger application server with more CPU and Memory, the problem went away

- Random 2397 (MQRC_JSSE_ERROR)/AMQ9699 due to client application passing bad MQ SSL packet and Windows dropping the LSASRV service. Server has to be rebooted to restore SSL services.
  http://support.microsoft.com/kb/828873

Summary

- In this session, we reviewed being proactive before problems occur, IBM's MustGather process and related forensic tools, identifying potential points of failure and case studies where MQ was the 'victim' and not the root cause of a problem.

  Hopefully the materials in this session will be useful to you the next time someone asks you "What's wrong with MQ"? You can say "absolutely nothing". "Now what can I do to help triage this issue".

What's wrong with MQ?

Thank you for attending!

APPENDICES

Appendix A

Sample IBM Management Alert Form

(Contact your IBM Account Rep for what you should use)

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval

Appendix B

From: IBM Websphere MQ 7.1 Configuration Manual

TCP/IP connection limits

- The number of outstanding connection requests that can be queued at a single TCP/IP port depends on the platform. An error occurs if the limit is reached.

- This connection limit is not the same as the maximum number of clients you can attach to a WebSphere® MQ server. You can connect more clients to a server, up to the level determined by the server system resources. The backlog values for connection requests are shown in the following table:

  http://pic.dhe.ibm.com/infocenter/wmqv7/v7r1/index.jsp?topic=%2Fcom.ibm.mq.doc%2Fcs11490_.htm

- If the connection limit is reached, the client receives a return code of MQRC_HOST_NOT_AVAILABLE from the MQCONN call, and an AMQ9202 error in the client error log (/var/mqm/errors/AMQERROn.LOG on UNIX and Linux systems or amqerr0n.log in the errors subdirectory of the WebSphere MQ client installation on Windows). If the client retries the MQCONN request, it might be successful.

- To increase the number of connection requests you can make, and avoid error messages being generated by this limitation, you can have multiple listeners each listening on a different port, or have more than one queue manager.

Appendix C

Websphere MQ Request for Enhancement (RFE)

Please help make MQ a better place by submitting an RFE to IBM. See:

http://www.ibm.com/developerworks/rfe/

Appendix D

The following excerpts came from an archived article on IBM's developerWorks at http://www.ibm.com/developerWorks . The Engagement Manager has granted permission for a reprint of these excerpts for this presentation. The article contains additional content on MQ health checks that can be performed by the MQ administrator.

Performing health checks for WebSphere Message Broker and WebSphere MQ

Author: Jun Tao (Tony) Cui (cuijunt@cn.ibm.com)
        Software Engineer - EMC
First Published January 23, 2008 in developerWorks

Regularly scheduled health checks are important in an SOA in which an advanced ESB provides connectivity and universal data transformation for both standards-based and non-standards-based applications and services. This article shows you how to how to perform a health check on WebSphere Message Broker and WebSphere MQ, including log check, message queue check, flow check, and database check. >>> *Next excerpt* >>>

Overview of health checks

A health check includes an operation system and file system check, log check, queue check, and flow check. Here is a brief description of these types of health checks:

Operation system and file system check

Examines the performance and capacity of the operation system and file system. For example, CPU and memory performance are checked to ensure that the system has enough resources to run the applications. The file system is checked to ensure that there is enough free space to store the temporary data and persistent files. >>> *Next Excerpt* >>>

Appendix D (2)

Performing health checks on WebSphere MQ

This section shows you how to do a file system check, log check, and queue check on WebSphere MQ.

WebSphere MQ file system check

The file system check ensures that there is enough space available for the WebSphere MQ log. If the disk where it resides fills up, you will get WebSphere MQ transaction errors. Assuming that the log file is at the default location of /var/mqm, use the following command to check file system space:

df -k /var/mqm

If the percentage of used space exceeds 90%, delete unneeded content or add another disk to this logical disk.

WebSphere MQ log check

After the file system check, examine the log files to look for exceptions. The default locations for the WebSphere MQ log files are /var/mqm/errors and /var/mqm/qmgrs/<queue manager name>/errors.

1. Check /var/mqm/errors for any global exceptions related to WebSphere MQ.
2. Check /var/mqm/qmgrs/<queue manager name>/errors. If WebSphere Message Broker uses a queue manager, check the file /var/mqm/qmgrs/WBRK6_DEFAULT_QUEUE_MANAGER/errors for queue manager exceptions.

WebSphere MQ queue check

The queue check examines queue manager status and queue manager components such as listener and channel.

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval

Appendix D (3)

WebSphere MQ queue manager status check

The queue manager contains queue components such as the queue, listener, and channel. In the WebSphere Message Broker runtime, the queue manager that message flows depend on must be running. To check queue manager status check,use the following command:
su - mqm
dspmq

A list of queue managers with status will be shown. Check whether the queue managers used by WebSphere Message Broker are all running. You can also use WebSphere MQ Explorer to view queue manager status. If a queue manager is not running, use the command strmqm <queue_manager_name> to start it.

WebSphere MQ listener status check

A listener in the queue manager is the bridge between the queue manager and the application. The message flow on WebSphere Message Broker connects to the queue manager through the listener, so the listener must be running. To

view the status of the listener, use the command ps -ef|grep runmqlsr | grep <listener_port>, where <listener_port> is the listening port of this listener. For example, ps -ef|grep runmqlsr | grep 141 gets the status of listeners whose listening ports are 141*, including 1410, 14101, and so on. To view the status of single listener, use the following command:
su - mqm
runmqsc <queue_manager_name>
display lsstatus(listener_name)

You can also use WebSphere MQ Explorer to view the listener status. If the listener is not running, use the following command to start it:
su - mqm
runmqsc <queue_manager_name>
start listener(listener_name)

Appendix D (4)

The runmqsc command enters the WebSphere MQ command window, and the configuration command to WebSphere MQ can be run through this command window.

WebSphere MQ channel status check

The channel defines a the message path. If user-defined channels are used in the WebSphere Message Broker environment, these channels must be running. To get the detailed information, use the following command:
su - mqm
runmqsc <queue_manager_name>
display channel(channel_name)

To get the channel status, use the command display chstatus<channel_name> from

the WebSphere MQ command window. You can also use WebSphere MQ Explorer to

view the status of the channels. If the channel is not running, start it by entering the

following command in the WebSphere MQ command window:

start channel<channel_name>

WebSphere MQ queue depth check

Queue depth remains low if the environment is in normal status. If exceptions occur or message flow is blocked, queue depth will increase in some queues. So you can check overall system health by checking the queue depth. If many messages remain in one queue, exceptions may be blocking the message flow. To determine queue depth, use the following command:
su - mqm
runmqsc <queue_manager_name>
display ql(*) CURDEPTH

Appendix D (5)

* means that this command will show the depth of all queues. WebSphere MQ Explorer can give you an overall picture of queue depth. If there are many messages in one queue, check the message flow and find the pain point. After that, increase the maximum depth of the queue to avoid queue overflows using the command

alter ql(<queue_manager_name>) MAXDEPTH(depth) from the WebSphere MQ command window. Investigate whether the messages are useful in the production environment. If the messages are not helpful, use the command clear ql(<queue_manager_name>) from the WebSphere MQ command window to clear the messages from the queue.

WebSphere MQ queue input/output count check

Queue input/output count is the number of applications that write messages to or read messages from the queue. The input/output count should remain low. A high input/output count may be caused by an exception in the message flow, such as a dead lock loop. To view the queue input/output count, enter the following command in the WebSphere MQ command window:

display ql(<queue_manager_name>) IPPROCS OPPROCS

>>> *End of Excerpts* >>>

Appendix E

Supplementary Information on Being Proactive

- Do get involved with your application teams when they design, develop and implement their MQ solutions
  e.g. Why are you hard coding the queue manager name?
      Why are you using persistence on Inquiry requests/responses?
      What is your traffic volume?

- Have and regularly exercise your MQ DR plan

- If possible, participate in business application stress testing to monitor and gauge any potential impact to your production MQ infrastructure

- Monitor your MQ and OS stats/event logs in development for potential production issues before the project goes 'live'

- Do have MQ naming standards and a MQ standard build doc

Appendix F

Other Case Studies:

1. Application spawning runaway MQ connections
   - Control on SVRCONN using maxinst and maxinstc parameters/2059
2. Receiver channel in a paused state and sender gets AMQ9558 due to max qdepth reached on receiving side. Servicing application not picking up messages quick enough. Bumped up queue max depth to resolve issue until application solution in place.
   http://www-01.ibm.com/support/docview.wss?uid=swg21052117
3. Invalid MQ handle addresses or MQ command issued after a MQDISC resulting in unusual MQ errors. Use MQ trace, application activity trace (actvtrc), MA0W or OS trace
   http://www.ibm.com/developerworks/websphere/library/techarticles/1306_bushby/1306_bushby.html
4. Application not using FAIL_IF_QUIESCING on z/OS Qmgr shutdown resulting in Orphaned CSQxCHIN region which has to be cancelled through the console
5. Bug in TCP/IP software on RCVR side causing remote SDR channel AMQ9208 econnreset error. OS and Sniffer trace found problem. Obtained and applied patch from Vendor to TCP/IP stack to resolve issue.
6. Sender-Receiver Channel connection broken and sender can't reconnect to the Receiver channel because its still running.
   http://www-01.ibm.com/support/docview.wss?uid=swg21080140
7. Choke router at or near capacity breaking connections between MQ client in the DMZ and the MQ servers on the Internal Network which resulted in 2009 reason code.
8. Client getting MQ errors when their app server CPU and/or memory exceeds a certain threshold (e.g. Windows CPU>=60%)

Appendix F (2)

9. MQ server receiving 10054 connection reset by peer when client machine is recycled. Note-check for orphaned connections.

10. Single point of failure can affect multiple MQ servers on same subnet (spread them out)

11. May need to actually check, right at your server, for hardware fault lights being on

12. Some local queue objects corrupted after server reboot due to queues being in use and MQ recovery process nuked when server was rebooted.
Media recovery aside, the M&T MQ team created a batch script to gracefully end the queue manager and service, validate the amq* and run* processes were no longer running and then reboots the server. Script can be done manually or placed in exit.

   http://pic.dhe.ibm.com/infocenter/wmqv7/v7r5/index.jsp?topic=%2Fcom.ibm.mq.con.doc%2Fq018550_.htm
   http://www.ibm.com/developerworks/websphere/library/techarticles/1307_bhat/1307_bhat.html
   http://www-01.ibm.com/support/docview.wss?uid=swg1IV37034 (Note-did not have this problem but including fyi).

13. Qdepth rollercoaster effect due to insufficient CICS (MQ) TCB Threads allocated
   - 1st occurred when TCB threads managed by MQ moved to CICS 3.2

14. Failing NIC card on client or server resulting in MQ connection errors
   - Check OS-MQ-NIC event logs, display i conn, run TCP and MQ trace on target side, ping NIC IP using script and/or triangulation

15. 'Black Hole' router dropping (MQ) network packet
   http://support.microsoft.com/kb/314825 & http://support.microsoft.com/kb/159211

16. 2059 error-Multiple MQ servers assigned to same blade server/chassis and blade has a problem. Break up/assign members of a MQ server pool to different devices/segments

15503-What's wrong with MQ?

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval