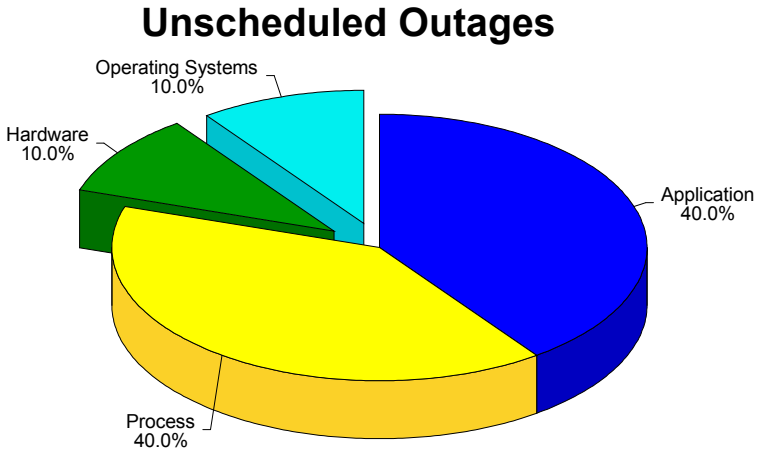
Software Group

WebSphere MQ
High availability solutions

Carl Farkas
SW IOT TechWorks zWebSphere Application Integration Consultant
IBM France D/2708
Paris, France
Internet : farkas @ fr.ibm.com
Notes : Carl Farkas/France/IBM @ IBMFR

Outages are evil and expensive



Source: Gartner Group

So what do we do?

- The objective is to achieve 24x7 processing of all messages
- Not always achievable, but we can get close - how many 9's
  - 99.9% availability = 8.5 hours downtime/year
  - 99.999% = 5 minutes
  - 99.9999% = 30 seconds
- Avoid application awareness of availability solutions

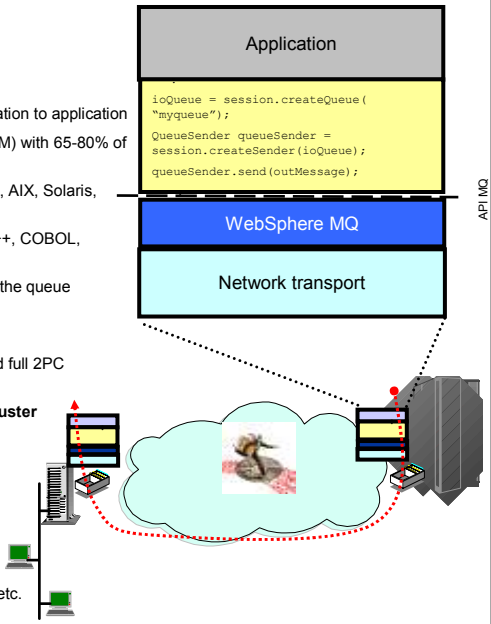
So what tools does MQ provide to avoid outages?

Agenda

- MQ Clusters
- Shared Queues
- Hardware clustering (eg. HACMP, MCS, MQ multi-instance, etc.)

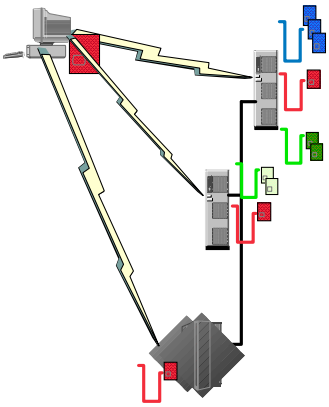
What's WebSphere MQ?

- A simple, efficient API for sending/receiving data messages
- Enables fast, reliable **asynchronous messaging** from application to application
- An industry standard for Message-Oriented-Middlewares (MOM) with 65-80% of the market.
- Can be used on over **45 different platforms** (Windows, Linux, AIX, Solaris, HP-UX, iSeries-AS/400, etc, etc., and of course…… System Z !)
- Can be used from **all major programming languages** (C, C++, COBOL, Fortran, BAL, PL/I, Java/JMS, VB, RPG, etc.)
- Includes support for the **MQ Client** – a remote API accessing the queue manager resources.
- Other unique features of MQ…
  - Provides a 2PC syncpoint manager for distributed MQ and full 2PC **transactional** participation on z (CICS, IMS, RRS)
  - Support for high-availability and load-balancing via **MQ Cluster** as well as **Shared queues** with Sysplex on Z
  - Support for **point to point** and **pub/sub** messaging
  - Support for message grouping and segmentation
  - Support for SSL authentication and encryption
  - Support for message compression
  - Exploits each platform, and in particular z/OS, taking full advantage of RACF, Sysplex, SMF, CICS & IMS Bridges, etc.

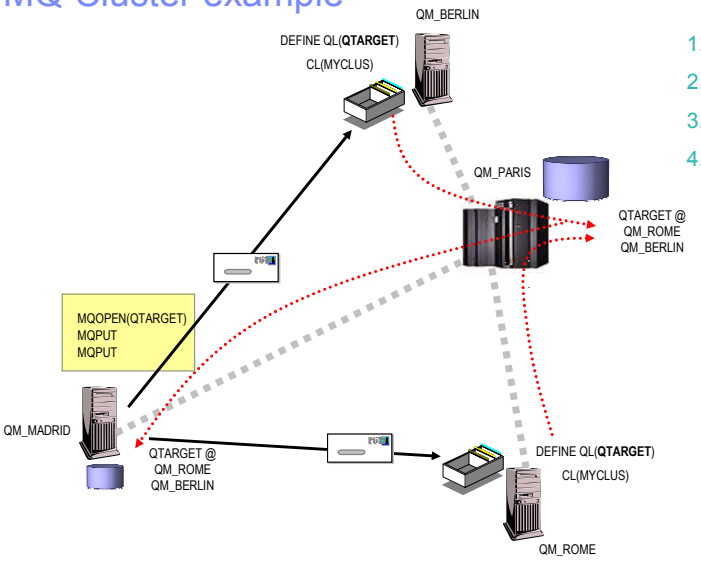


MQ Clusters

- **Simplify administration**
  - Queues are "auto-magically" available remotely to all MQ Cluster members without any administrative definition
- **A "logical" queue can have multiple "physical" copies residing throughout the MQ network**
  - High availability
  - Load-balancing
- **Message distribution by….**
  - "round-robin"
  - Weighting
  - Priority
  - User exit



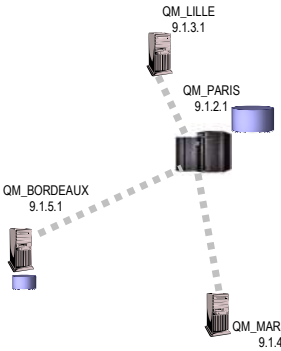
MQ Cluster example



1. Define the queues
2. MQOPEN
3. MQPUT
4. MQPUT

5

MQ Cluster definition example



QM_PARIS
ALTER QMGR REPOS(MONCLUS)

DEFINE CHANNEL(TO.QM_REPOS1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME(9.1.2.1) CLUSTER(MONCLUS)

QM_LILLE

DEFINE CHANNEL(TO.REPOS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME(9.1.2.1) CLUSTER(MONCLUS)

DEFINE CHANNEL(TO.QM_LILLE_APPSERVER1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME(9.1.3.1) CLUSTER(MONCLUS)

DEFINE QLOCAL(QCIBLE) CLUSTER(MONCLUS)

QM_MARSEILLE

DEFINE CHANNEL(TO.REPOS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME(9.1.2.1) CLUSTER(MONCLUS)

DEFINE CHANNEL(TO.QM_MARSEILLE_APPSERVER2) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME(9.1.4.1) CLUSTER(MONCLUS)

DEFINE QLOCAL(QCIBLE) CLUSTER(MONCLUS)

QM_BORDEAUX

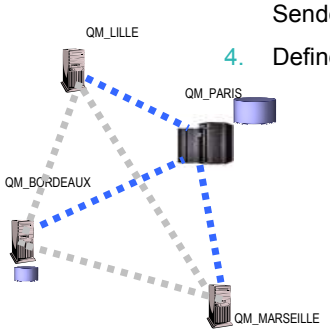
DEFINE CHANNEL(TO.REPOS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME(9.1.2.1) CLUSTER(MONCLUS)

DEFINE CHANNEL(TO.QM_BORDEAUX APPCLIENT) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME(9.1.5.1) CLUSTER(MONCLUS)

Administration economies

With MQ Clusters....

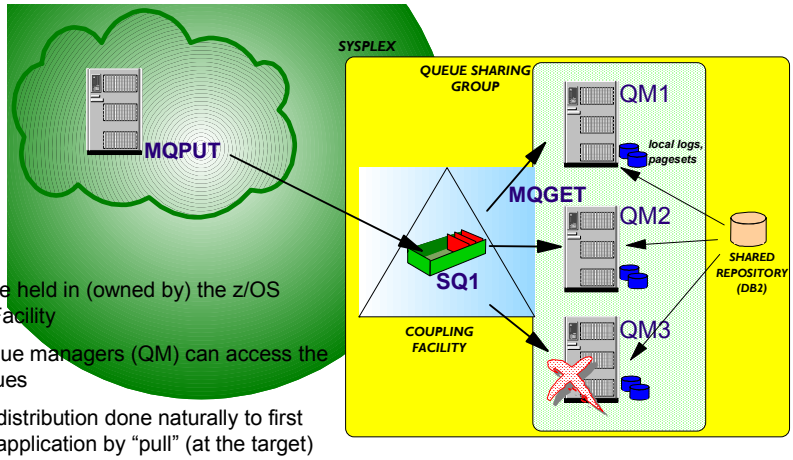
1. Define a full repository on (at least) one machine
2. Define the Receiver Cluster channel on the the QM with the full repository
3. For every other QM in the Cluster, define the Sender and Receiver Cluster channel
4. Define each Cluster object (eg. Queue)



	W/O cluster	With cluster
CHANNELS	12	7
QLOCAL	4	4
QREMOTE	12	0
XMIT QUEUE	12	0

- *Major time-saver for the MQ administrator*
- *Less definitions: fewer errors*

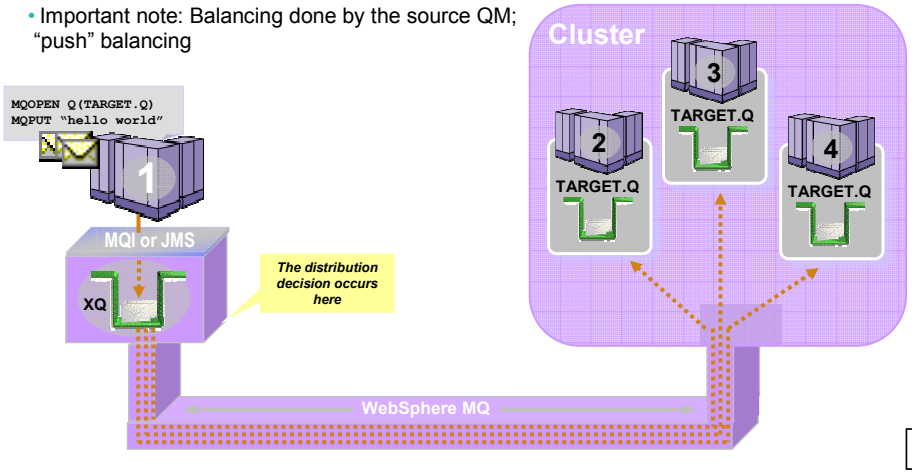
MQ Shared Queues



- Queues are held in (owned by) the z/OS Coupling Facility
- All the queue managers (QM) can access the same queues
- Workload distribution done naturally to first available application by “pull” (at the target)
- Large messages (>63KB) off-loaded to DB2 BLOBs

MQ Workload Balancing with Clustering (distributed MQ)

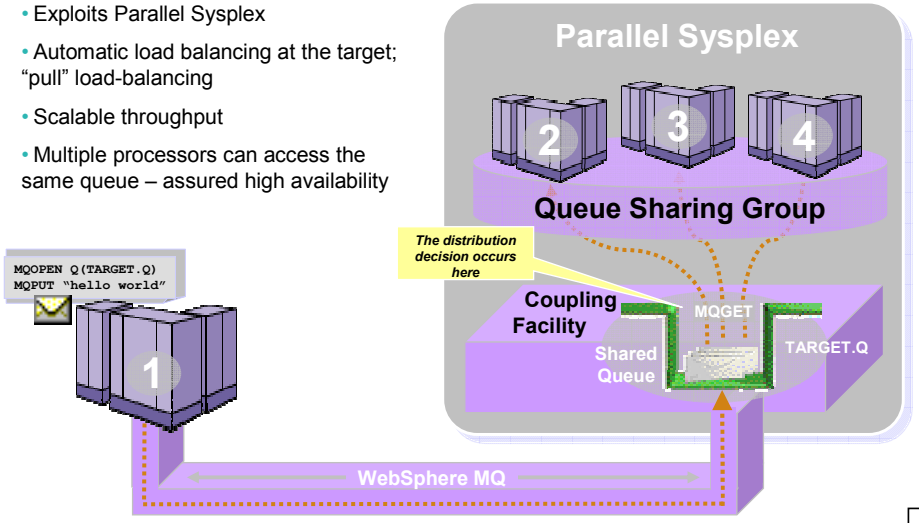
- Balancing either at every MQPUT or MQOPEN
- "Round Robin" balancing by default; with possibility to assign priorities & weightings
- Important note: Balancing done by the source QM; "push" balancing



1

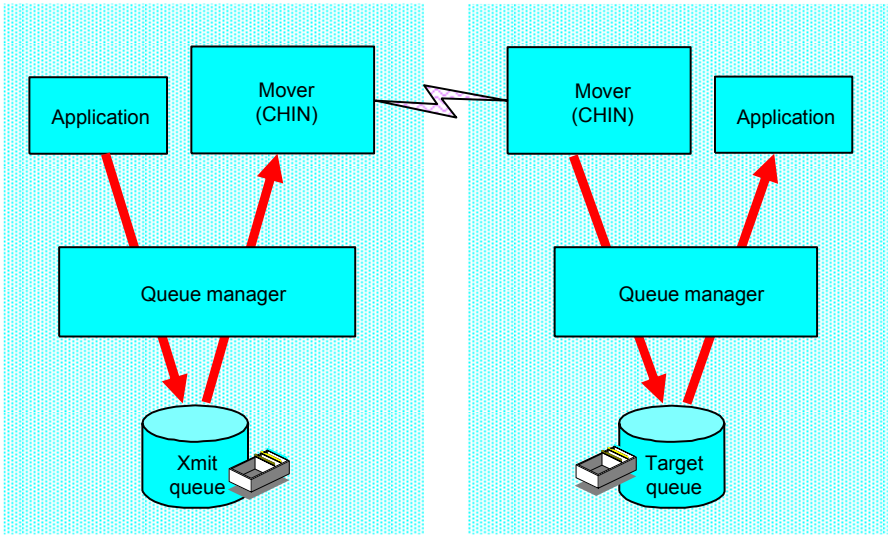
MQ Shared queues on z/OS – the z-exclusive!

- Exploits Parallel Sysplex
- Automatic load balancing at the target; "pull" load-balancing
- Scalable throughput
- Multiple processors can access the same queue – assured high availability

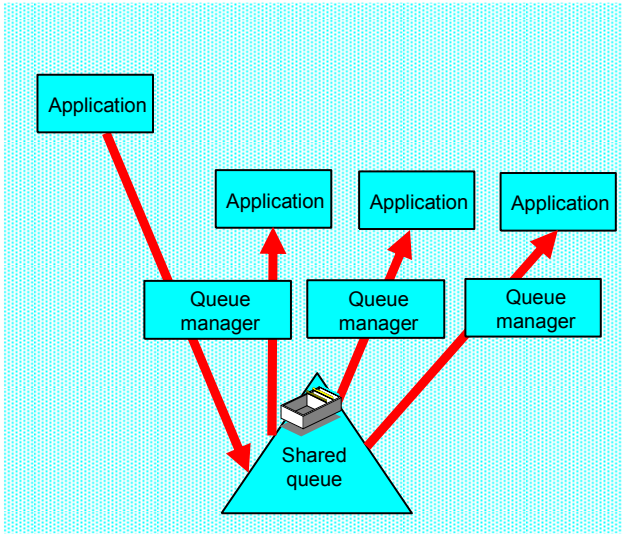


1

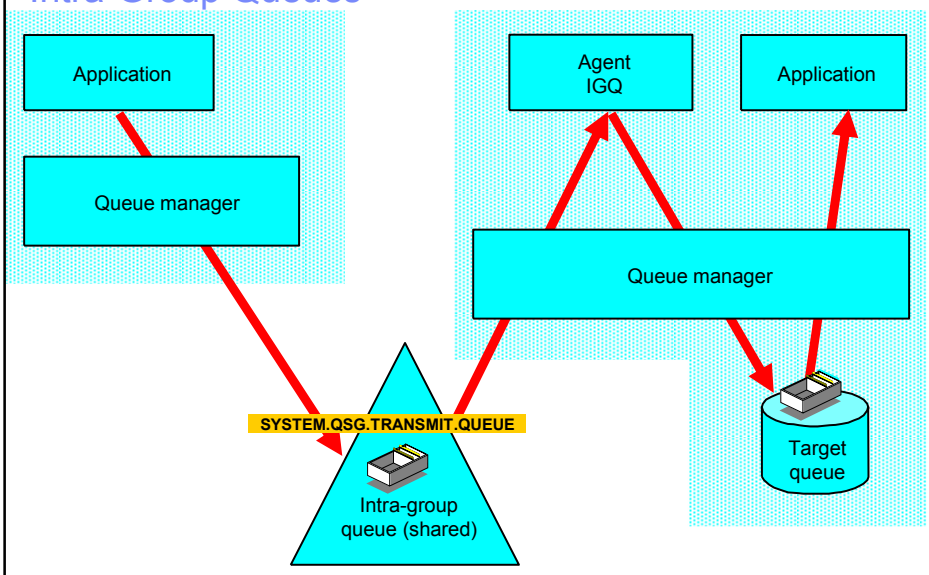
Classic MQ - MQ transmissions



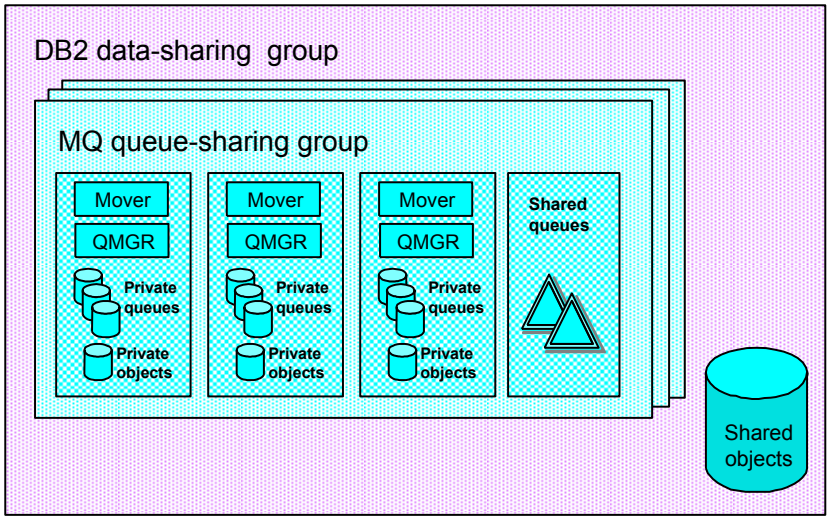
MQ z/OS - MQ z/OS transmissions via SQ



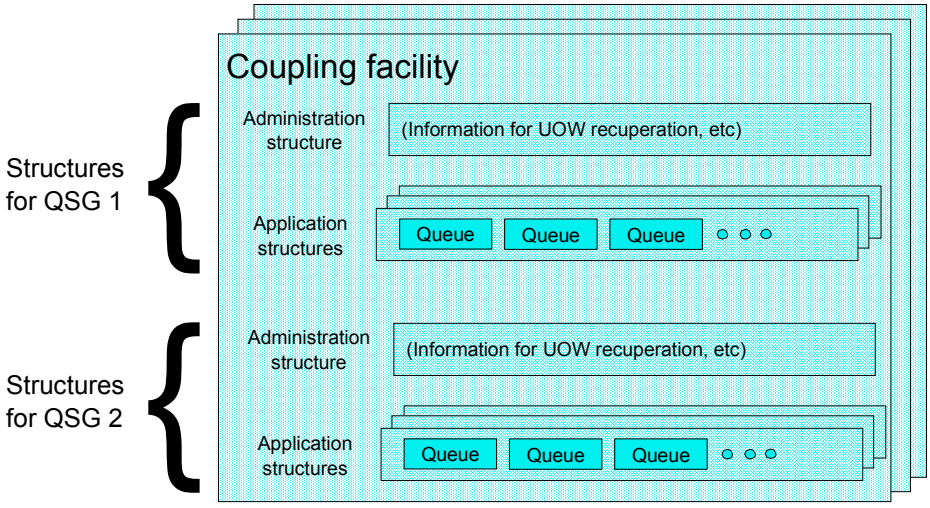
Intra-Group Queues



Queue Sharing Group – QSG



QM Structures



Planning the Coupling Facility Structures

- Naming convention for your QSGs (max 4 chars)
  - Consider adopting the name of an existing QMgr
- Number of QMgrs in each QSG (max 32)
  - Application throughput requirements
  - Redundancy
- Number of application structures
  - Max 512 per sysplex, 63 max per QSG
  - Max 512 queues per structure
- Size of structures
  - Administration – at least 10MB
  - Application – msg size, max no. of msgs, no. of queues
- See MQ for z/OS Concepts and Planning Guide, chapter 17, for recommendations, structure sizes, DB2 allocations, etc. (or InfoCenter http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/topic/com.ibm.mq.csqsat.doc/zc12690_.htm)

Configuration for shared queues

1. Customize and execute the sample MQ jobs in SCSQPROC to perform **DB2 initializations**
   a) CSQ45CSG (create Storage Group)
   b) CSQ45CDB (create DB)
   c) CSQ45CTS (create TableSpace)
   d) CSQ45CTB (create 12 tables/indices)
   e) CSQ45BPL (bind plans)
   f) CSQ45GRT (grants)
2. Customize supplied CSQ4CFRM job to **initialize the CF** with IXCMIAPU utility

```
STRUCTURE NAME(MASGCSQ_ADMIN)
       INITSIZE(10240)
       SIZE(20480)
       PREFLIST(CF06,CF05)
       REBUILDPERCENT(5)
       FULLTHRESHOLD(85)

STRUCTURE NAME(MASGAPPLICATION1)
       INITSIZE(20480)
       SIZE(81920)
       PREFLIST(CF06,CF05)
       REBUILDPERCENT(5)
       FULLTHRESHOLD(85)
```

3. **Activate the structures** with SETXCF START

Configuration for shared queues, cont'd

4. Customize and run sample CSQ45AQS to invoke CSQ5PQSG tool for **creating the QSG**

```
//stepname EXEC PGM=CSQ5PQSG,
//   PARM='ADD QSG,qsg-name,dsg-name,DB2-ssid'
```

5. Customize and run sample CSQ45AQM to invoke CSQ5PQSG tool for **adding the queue managers** into the QSG:

```
//stepname EXEC PGM=CSQ5PQSG,
//   PARM='ADD QMGR,qmgr-name,qsg-name,dsg-name,DB2-ssid'
```

6. **Modify the CSQZPARM** to tell the QM that it's part of a QSG

```
CSQ6SYSP ....
QSGDATA=(qsg-name,dsg-name,db2name,db2serv ),
```

7. Start up your QM!

Creating MQ objects for Shared Queues

- MQ recognizes the structure by its short name (12 characters); z/OS recognizes the structure by its full name (16 characters)
- qsg-name || str-name (Application structures)
- qsg-name || CSQ_ADMIN (Administration structure)

- Use CSQUTIL to define the application: DEFINE CFSTRUCT

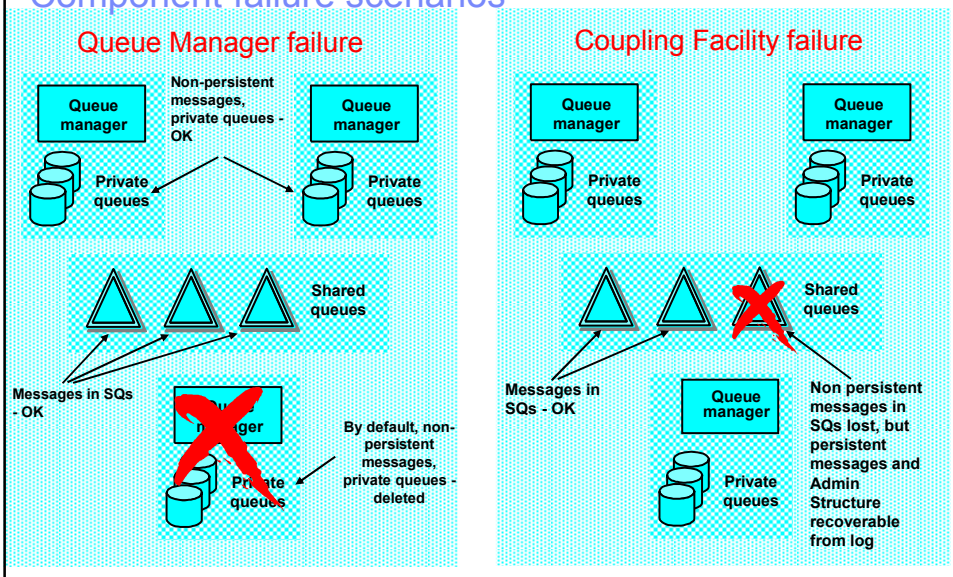
```
DEFINE CFSTRUCT(str-name)CFLEVEL(level) RECOVER(YES | NO)
```

- Define your queues as usual, but specify the CFSTRUCT name

```
DEFINE QLOCAL(queue-name) QSGDISP(SHARED) CFSTRUCT(str-name)
```

- z/OS will actually create the structure the first time MQ references it

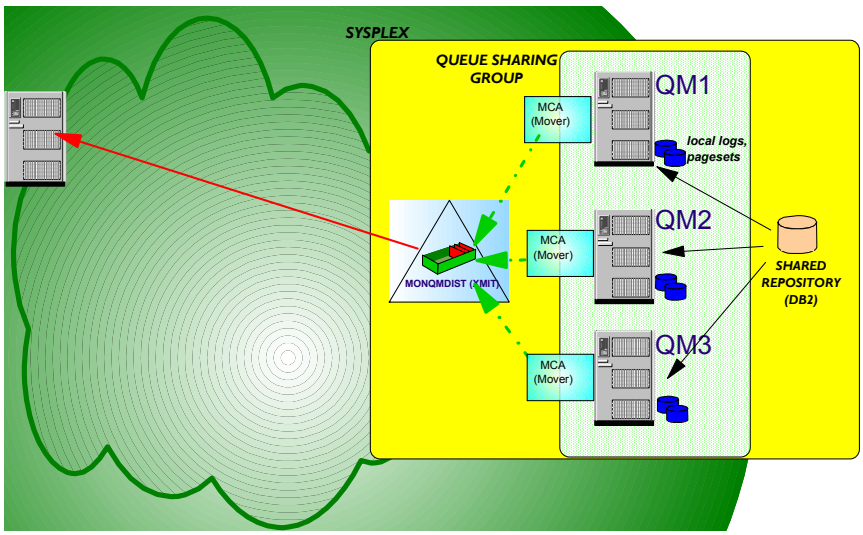
Component failure scenarios



Component failure – the moral

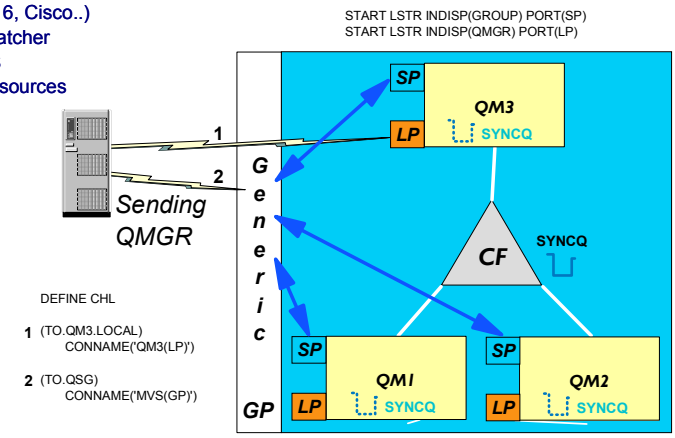
- Coupling facilities are very, very robust today
  - Might be an external box with a a UPS & non-volatile memory, but….
  - Typically just an LPAR with a very special, simplified, streamlined "Operating System"
- Coupling facilities can be "duplexed" whereby a 2nd copy is maintained, and this can be many kilometers away for DR purposes
  - This is a z/OS and hardware feature – totally transparent to MQ
  - Recovery to the primary CF is performed automagically, again, transparent to MQ
- You also should be thinking about using the RECOVER(YES) option on structures, and using BACKUP CFSTRUCT to perform regular shared queue backups

Shared Channels – outbound

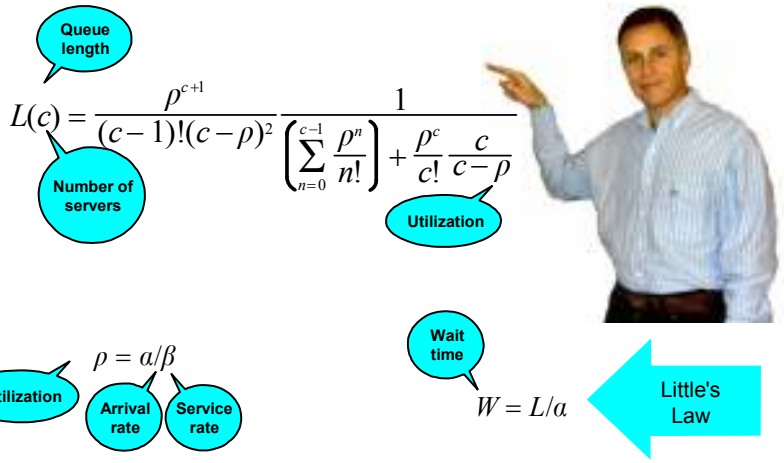


Shared channels – inbound

- Support pour dispatchers
  - Sysplex Distributor
  - DNS routeurs (2216, Cisco..)
  - IBM Network Dispatcher
  - TCP/IP WLM DNS
  - VTAM Generic Resources
  - etc.

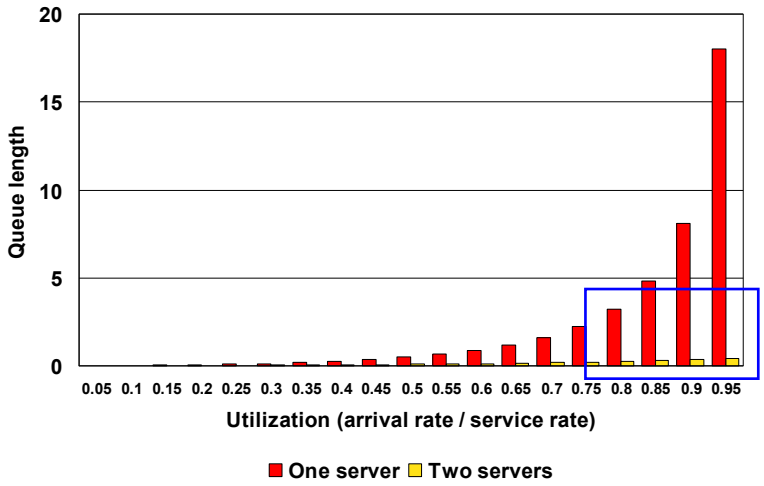


Queuing theory 101

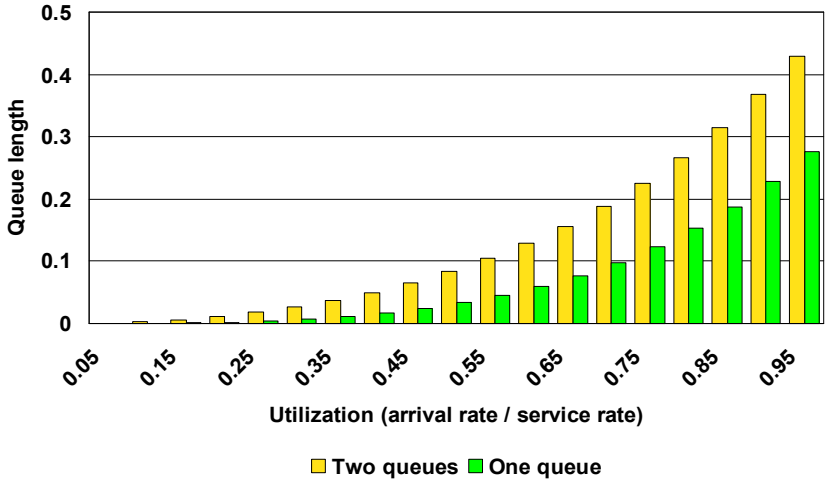


Scalability - Cluster technology

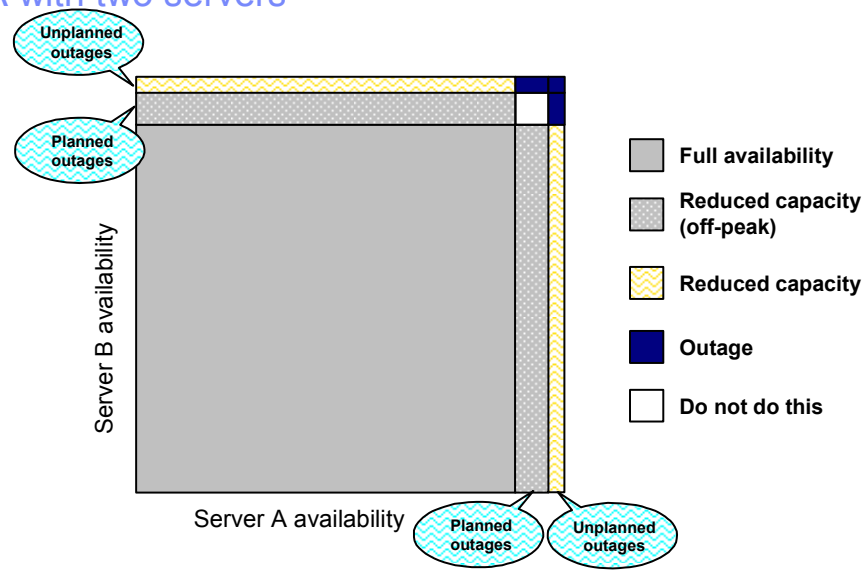
From queue theory, we know….



Scalability - Shared Queue technology



HA with two servers

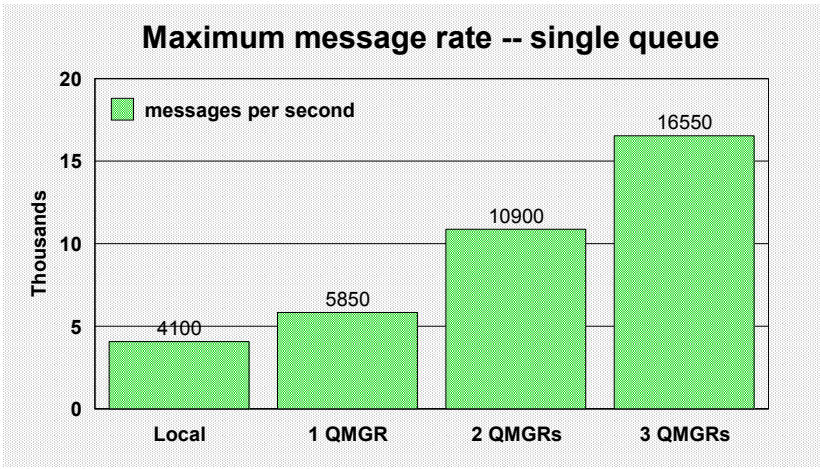


Availability - example

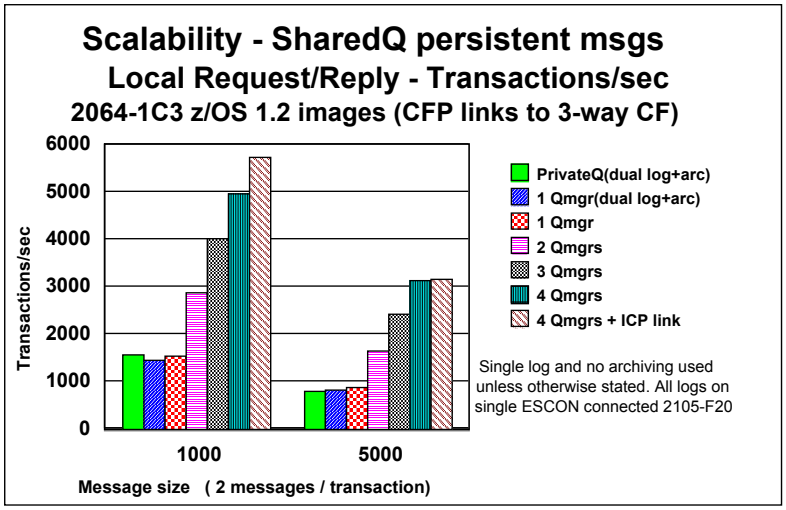
One server		Two servers	
Planned outages	1	Planned outages	0
Unplanned outages	0.5	Unplanned outages	0.002
Total	1.5	Total	0.002

Outages in hours per month

Performance – non persistent



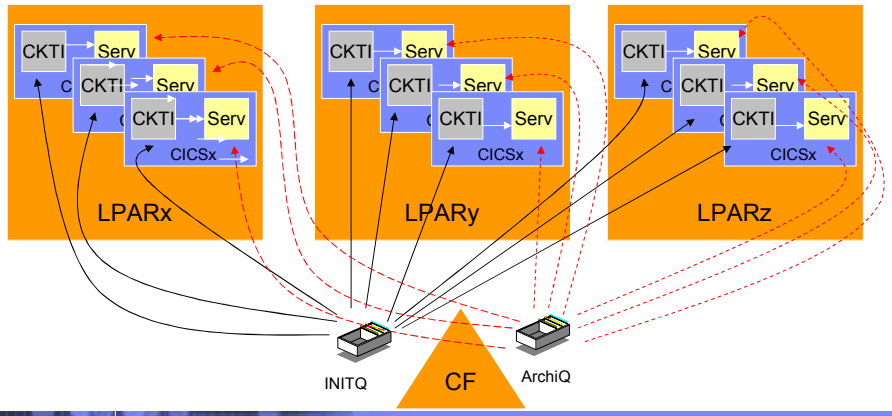
Performance - persistent



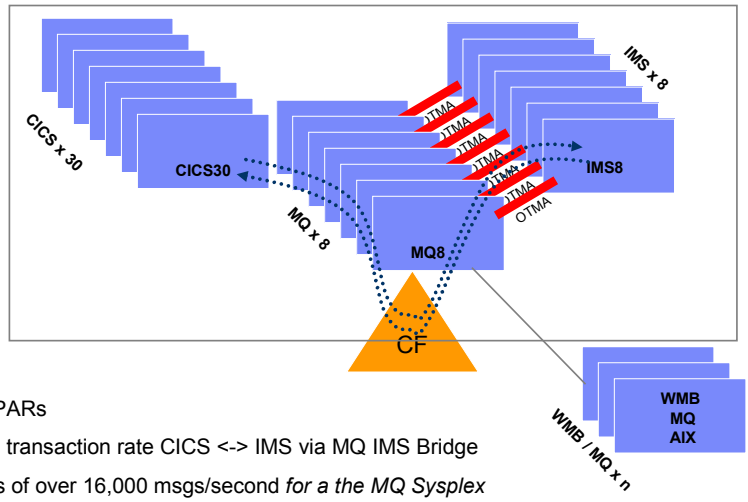
Over 11.000 msgs/second – near linear scalability

Shared Queues

- 3 x LPARs
- > 3 million MQPUT/day and > 5 million MQGET/day (2004)
- 400 MB application structures



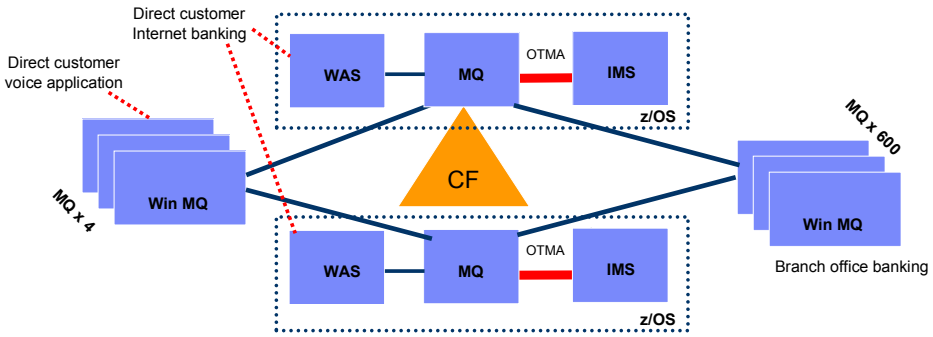
Shared Queues - Large banking institution



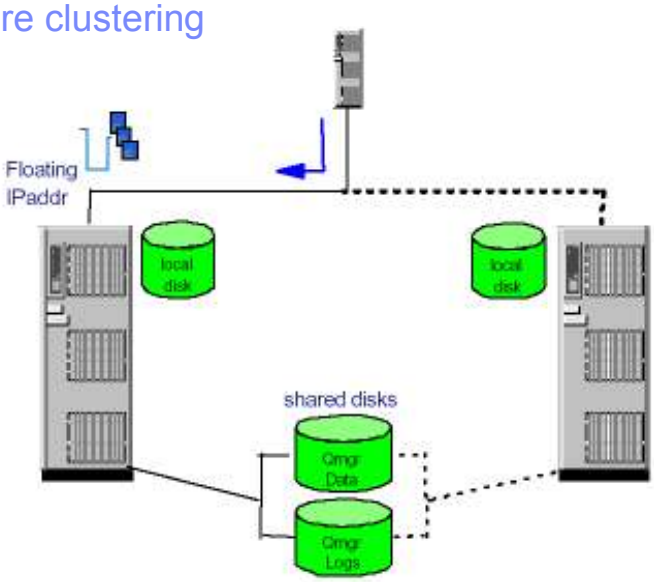
- 8 x LPARs
- Large transaction rate CICS <-> IMS via MQ IMS Bridge
- Peaks of over 16,000 msgs/second *for a the MQ Sysplex*
- Peaks of over 60 CICS-OTMA trans/second *for a single CICS (of 30)*

MQ IMS Bridge - Svenska Handelsbanken (SHB)

- Large Swedish bank with over 600 branch offices, using MQ IMS Bridge for many years
- Supporting several core Intranet, Extranet and Intranet messaging applications
  - WAS z/OS used for customer Internet banking, sending MQ-OTMA requests
  - Direct MQ-OTMA query/reply by 600 branch offices
  - Direct MQ-OTMA query/reply for customer voice application
- Attracted by ease-of-use, speed and reliability of MQ IMS interface
- Running with peaks of 8M msgs/day over Bridge



Hardware clustering



Comparison of Technologies

	Access to existing messages	Access for new messages
Shared Queues	continuous	continuous
WMQ Clusters	none	continuous
	automatic	continuous
HA Clustering	automatic	automatic
No special support	none	none

In summary

- MQ Clusters
  - Available on all MQ v6 and v7 platforms
  - Provides a simple load-balancing, scalability solution
  - Provides a minimal HA solution
  - Uses a “Push” type of logic – sending QM distributes
  - Requires non-affinity of messages with respect to QM
  - A given message is only available to a single QM in the MQ Cluster
- MQ Shared queues
  - Available exclusively on MQ z/OS
  - Provides a robust, optimized load-balancing, scalability and HA solution
  - Uses a “Pull” type of logic – most available receiving QM takes message
  - Requires non-affinity of messages with respect to QM
  - Messages are available to all the QMs in QSG
- Hardware clustering (eg. HACMP, MCS, ARM, etc.)
  - Neither load-balancing
  - Restart is relatively fast (even faster with MQ 7.0.1 and Multi-Instance QM)
  - Relies upon externalized disks and typically an O/S HA feature

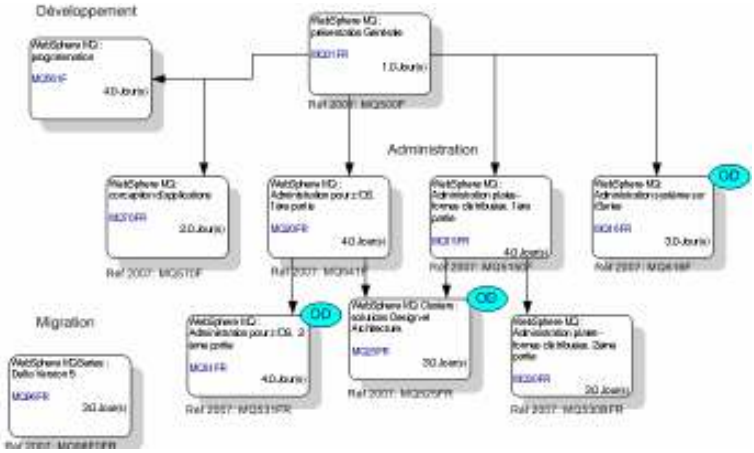
Backup

MQ Bibliography

- GC34-6926 WebSphere MQ v7 for z/OS Concepts and Planning Guide
- SC34-6929 WebSphere MQ v7 for z/OS System Administration Guide
- SC34-6927 WebSphere MQ v7 for z/OS System Setup Guide
- SG24-7839 High Availability in WebSphere Messaging Solutions
- SG24-6523 Parallel Sysplex Application Considerations
- REDP3636 MQ Queue Sharing Group in Parallel Sysplex Environment (Redpaper, draft)
- SG24-6864 WebSphere MQ in z/OS Parallel Sysplex (Redbook)
- SupportPac MP1E – MQ z/OS v6 Performance Report
- SupportPac MP16 – Capacity Planning & Tuning for WebSphere MQ

The complete MQ library is available in PDF at
http://www.ibm.com/software/integration/wmq/library/
Or online at
http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp

MQ Education (example IBM France)



For education details, see http://www.ibm.com/software/websphere/education/curriculum/appint/wmq/