IBM Websphere MQ

IBM MQ series --> Now it is called Webphere MQ

Websphere products (java based products) infrastructure -->	RAD (Rational application development) instead of eclipse, Application server
Websphere integration products	Websphere MQ, WMB, Data power, BPM (processor, lambordi) Helps in data transferring between the two applications
Websphere portals (Application deployed on websphere application server)	Customize the web page dynamically change the content by dragging drop features
Versions	5.3 started deploying web applications 6.0 7.0 contains publish subscribe mechanism 7.1 contains clustering using multi instance queue manager 7.5 8.0 (Latest)
what is MQ	Messaging and queueing Messaging -- program to program communications over the network between the diversified (large number of products) applications in the form of messages Application (source) ---------->(Queue Request)---------->Application (target) <----------(Queue Response) <---------- Application (source) write a message in a queue and application (target) read from queue
Uses of MQ	1. Routing 2. point to point communication (both application are not connected directly) 3. platform independent
Types of messaging	1. Synchronous -- send some data and waiting for a response message 2. asynchronous -- send some data and forget (doen't wait for response message from destination server) Queue is the safe place to store messages Transaction is initiated but takes tomorrow only, so the message is stored in queue
Queueing	safe place to store messages, it follows FIFO (First in First out) Eg: application store some data if there any network lag: message is on queue then it will read message one by one Application (source) ---------->(Queue Request)----------x---->Application (target) <----------(Queue Response) <----------

Message

Two parts:
1. Header (Reply to queue manager, To , From field)
2. Body (contains MQ details, Business data, Messages)

Persistancy messaging: if a message is having persistancy, those message will be recovered. Queue manager is restarted. WMQ is responsible to deliver those messages.
Non persistancy messaging: if there is any failures, non persistant takes place. application is running but server restarted then application data is lost. so we cannot recover it.

Benefits of WMQ:

1. common application program interface (MQI) -- object called queue manager, storing data.

App(Java) ----> QueueManager <----MQI-- New_App(Java)
-------> Queue request ---------->
<------- Queue response <----------

Application writes messages in a queue, the request and response takes between any kind of application.

2. Assured message delivery: -- (Once and only delivery) 99.99 % message gets delivered. Once App writes a message on queue, any issues in between even tough the message gets delivered.. Only persistent message will get delivered.

3. Time independent processing: -- Source APP will send message even though the target APP is not working or shutdown.

4. Parallel processing -- you can send multiple request in single queue parallely,

5. Faster application development -- Java app with connection with DB.

Java App ---------> Queue manager ---- developes in .net ----> .NET App

User enter name and mobile number ---> returns the message in queue -- > option called insert or update --> execute in .NET App

Saved in .xml format --> it converts in web MQ readable format

Planning

1. **MQ Server** --> License fee + create queue manager (providing messaging and queueing functionality

2. **MQ client --**> FREE + no queue manager + App can connect to multiple destination + provide failover functionality

App connect to Chennai QM (it fails) so it diverts to Bangalore QM

Flexible services to customer

According to the business requirement we takes any one

1. Client server architecture
2. server to server architecture
3. Hub and scope communication

Client server architecture

APP -----MQ Client connect ----------> MQ Server installed in DB_APP
(QM-1, QM-2……)

APP ----MQ Client interface to connect ---------> QM <---- DB_APP

Two types to MQ server to connect QM
1. Binding mode - no network required, fast, same system
2. Client mode - network required, slow, connect to multiple destinations

Server Server architecture

App1 MQ Server ---------- MQ server (App2)
QM <----------channel----------> QM

send messages to channel, asynchronous communcation takes place.

Eg: Few APPs will generate 1000s of messages, then server-server architecture is best to use

Hub n Spoke

App1 ---------- Hub (MQ server) ---------- App2
----------> QM <----------
----------> QM1 <---> QM2 <---------- Create two QM
if one fails, then other one will work, monitor from centralized servers. Number of installations and licenses reduced.

Installation

1. MQM - Mount point creation (Like C drive in computer, space is less so we cannot transfer a large file). If I install MQ server, IBM suggests to go for MQM mount point. Rise a request for MQ installation to middleware administrator.

2. MQM group creation: root user in linux environment, administrator in windows. To alter or create MQ objects, so we use MQM group.

3. create MQM user

4. add MQM user to the group

5. installation should be root(administrator) user

6. configuration using MQM user

we have to coordinate with systems operations team to install MQ

Configuration

Webphere MQ commands

1. Control commands: mostly used, 40-50 commands, create QM objects, we use MQ explorer through that utility we can execute commands.

1. dspmqver -- shows version of MQ
2. crtmqm -- create QM
3. strmqm -- start QM, for default QM no need to specify Queue manager name
4. endmqm -- stop QM
5. dltmqm -- delete QM, while deletion QM should be in stopped state
6. dspmq -- all QM status, we use -m <QM name> if any specific QM name

2. MQSC (runmqsc) scripting commands: if i want to create or alter QM attributes and objects.

1. ALTER
2. CLEAR
3. DEFINE
4. DISPLAY
5. END
6. START
7. STOP
8. REFRESH
9. RESET

10. RESOLVE
11. SUSPEND
12. PING
13. RESUME

3. PCF commands: Programmable command format
   PCF = control commands + MQSC
   Eg: MQ explorer (windows/linux)

we have to follow systematic approach in organization

Queue manager creation

Responsible for messaging and queueing functionality

Naming convention:
1. QM and its objects we can give up 48 chars names but for channel 20 chars
2. Always use UPPERCASE, shortname and application specific (if 300 Apps, if any one app fails, we can go to that app QM and fix it)

crtmqm - display all attributes of create QM

crtmqm -a Allow this group access to queue manager files. (MQM group leaders access this)

-c descriptive text: description of the QM

-d default transmission queue name (MQM group leaders

-ll use linear logging, continuous writing log file, we ave to clean up, restart,media recovery, damaged objects we cannot recreate

-lc use circular logging (default) 1 -x--> 2--x--> 3 again it will come to 1 restart recovery

-lp log primary, default these are visible, if QM existed then secondary logs are generated

-ls log secondary

-u dead letter queue, we can assign only one dead letter queue to QM, undelivered messagees will be stored here.

-q default queue manager for only one default queue manager we are able to create for installation, there might be multiple QM created with -q, but latest is default one

-ld	log file path
-md	directory structure used for queue manager data
-sa	After 7.0, start automatic option only for windows
-sax	same but for multiple instances
-lf	log file size, specified in units of 4kb pages

endmqm -I TEST — Force ending of QM

crtmqm -q -sa TEST — multiple command attributes

MQ Explorer - Content

Queue Managers

Filter: Standard for Queue Managers

Queue manager name	Command level	Queue manager status	Platform	Queue-sharing group name	Dead-letter queue	Description	Default bind ty
QM_1	701	Stopped	Windows				Shared
QM_2	701	Stopped	Windows				Shared
TEST	701	Stopped	Windows				Shared
TEST_1	701	Running	Windows				Shared

amqmdain auto TEST — alternative to -sa command…. It will be started automatically
system gets slow if we have multiple automatic QM

amqmdain manual TEST

dspmq --o installation — all QM's installation path

Create MQ --> Start MQ

- IBM WebSphere MQ
  - Queue Managers
    - TEST
      - Queues
      - Topics
      - Subscriptions
      - Advanced
        - Channels
        - Client Connections
        - Listeners
        - Services
        - Process Definitions
        - Namelists
        - Authentication Information
    - TEST_1

Queue creation — 4 Types:

1. Local
2. Alias
3. Model
4. Remote

Local Queue — Queue is a safe place to store messages
only local queue can store messages
Application can put messages but cannot get messages

Alias Queue — cannot store messages,
always pointing into queue or topic(from version 7 alias queue pointing into topic also

Model Queue — Template, QM will use for dynamic queue creation (dynamic Q's also local Qs)

Remote Queue	cannot store any messages just we are defining structure only. (distributing queueing one QM communicate with other QM App --------> QM1 ,--------> QM2 < -- App2 ----------->Q.RMT <------------->Local Q (msgs will be here) Local Q is residing with the Remote Q manager.
Other Queues are	1. Dead Letter Queue: 2. Initiation Queue (Triggering mechanism) 3. XMITQ (Transmission Queue)
Create QM objects	we need to use MQSC commands
Steps:	
runmqsc QM1	Command tostart using MQSC commands
DEFINE QLOCAL(Q1.LCL)	once created we cannot change queue name
DEF QL(Q2.LCL) DEFPSIST(Yes)	if we need persistency messages then use Yes
DISPLAY QLOCAL(Q1.LCL)	create date, usage is normal, CURDEPTH (10) how many messages are there? MAX DEPTH(5000) QM can store 5000 messages. MAXMSGL(4kb)
DISPLAY QSTATUS(Q1.LCL)	display Queue status details
amqsput Q1.LCL QM1 **MSG1** **MSG2** **MSG3**	Testing purposes, inserted 3 messages and display it has it holds 3 messages EXECUTE Command IN NEW WINDOW
amqsbcg Q1.LCL QM1	we browse the messages
DELETE QLOCAL(Q1.LCL)	deletes local queue, queue must be empty
CLEAR QLOCAL(Q1.LCL)	clears the queue, 0 messages
ALTER QLOCAL(Q1.LCL) **MAXDEPTH(100)**	altering the queue
DISPLAY QMGR	see QM details