



IBM Software Group

Cómo mantener los canales en funcionamiento

Morag Hughson and Paul Clarke

hughson@uk.ibm.com , paulg_clarke@uk.ibm.com

WebSphere. software

@business on demand software

© 2003 IBM Corporation

V1.0.1

IBM Software Group | WebSphere software



Introducción

N
O
T
E
S

IBM ofrece este documento como una serie de recomendaciones que se basan en un amplio conocimiento de IBM acerca de WebSphere MQ y MQSeries y en la experiencia con los clientes basada en el uso de estos productos y sus entornos y objetivos. La aceptación y aplicación de estas sugerencias dependerá del criterio del lector. Puede que escuche opiniones diferentes de otros expertos. Hay diversas áreas de WebSphere MQ que son susceptibles de opiniones diferentes, las cuales pueden variar en función del entorno específico donde se utilice WebSphere MQ.

Una vez que el lector se haya familiarizado con WebSphere MQ, podrá decidir qué recomendaciones desea seguir o tener sus propias opiniones para su entorno específico.

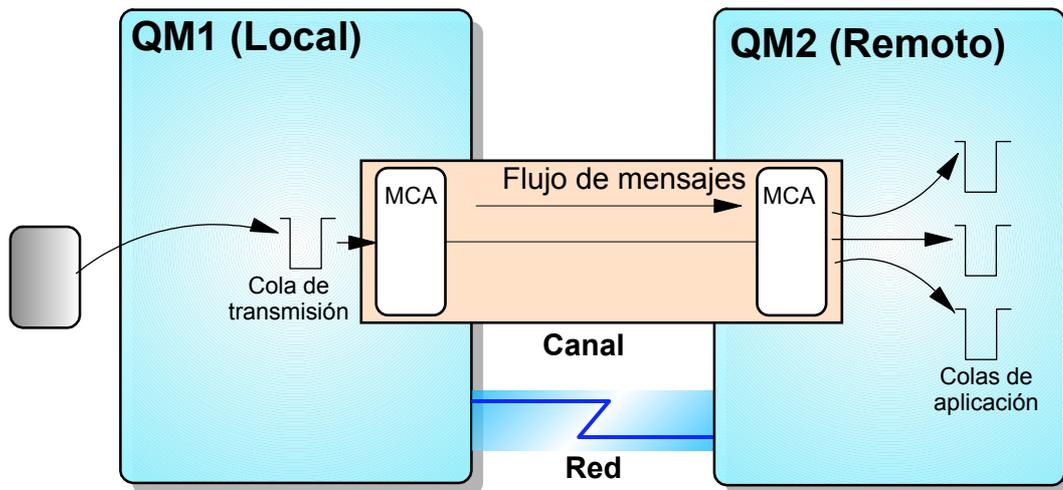
Este documento está destinado a los Administradores de Sistemas y a cualquier otra persona que ofrezca soporte a las redes de WebSphere MQ. Es de esperar que ofrezca las siguientes ventajas:

- Coherencia en los procesos de administración
- Máxima disponibilidad de las aplicaciones
- Ayuda para evitar errores comunes que suelen realizar los principiantes
- Asistencia al personal en la fase inicial para convertirse en expertos en WebSphere MQ
- En general, garantizar un inicio suave y alcanzar el éxito en los proyectos WebSphere MQ

La mayor parte de la información que se presenta aquí hace referencia a versiones de MQSeries y WebSphere MQ. Compruebe la documentación específica de su versión para asegurarse que se aplica esta información.

© 2003 IBM Corporation

Arquitectura de canal



Arquitectura de canal - Notas

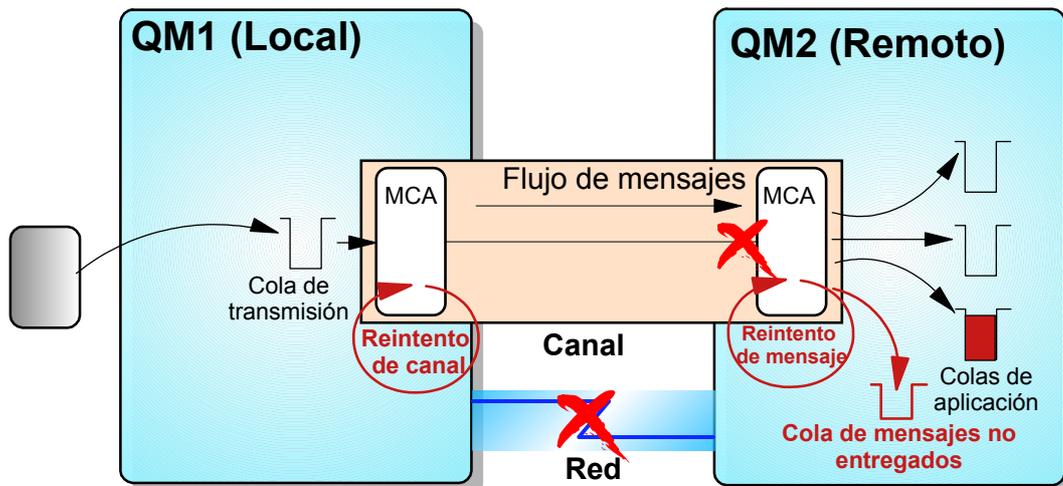
En este gráfico se representa la arquitectura de canal de WebSphere MQ. Vamos a utilizar este gráfico a lo largo de la presentación para ilustrar los diferentes temas del canal.

N En la parte principal del gráfico se muestra que las aplicaciones pueden transferir mensajes (ya sea a través de definiciones de cola remota o por direccionamiento explícito) que llegarán a una cola de transmisión para que se envíen al sistema remoto. QM1 será nuestro sistema local en este gráfico y QM2 será nuestro sistema remoto. El MCA emisor que da servicio a nuestra cola de transmisión obtiene los mensajes de la cola y los envía a través del sistema remoto, el protocolo de comunicación que se especifica en la definición de canal, donde los recibe el MCA receptor y los envía a la cola de la aplicación de destino correspondiente. Este par de MCAs se conoce como el canal.

Perfecto, pero ¿qué puede fallar?

O
T
E
S

¿Qué puede fallar?



- El canal puede finalizar: finalización del QMgr remoto
- Imposible entregar los mensajes: cola no válida o llena
- Anomalía en la red
- Rendimiento

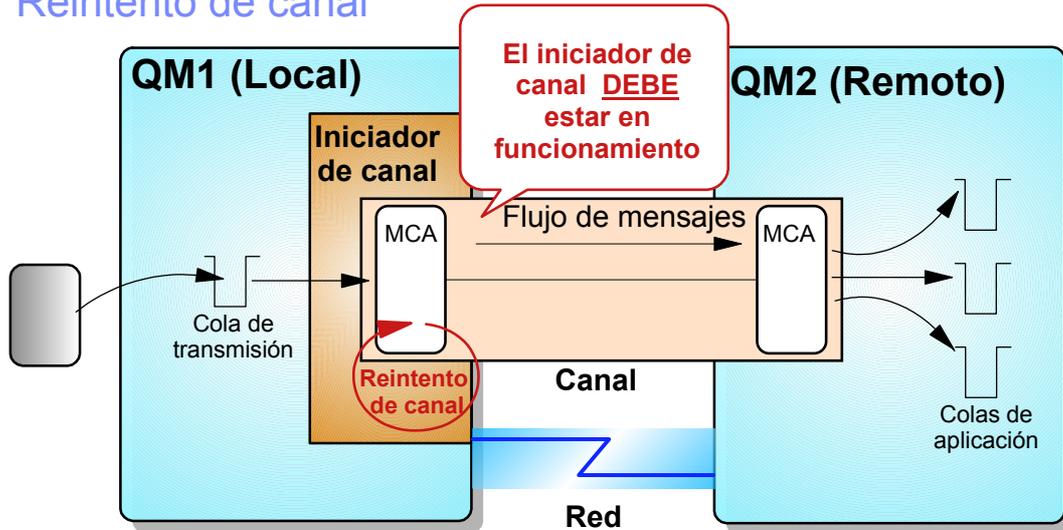
¿Qué puede fallar? - Notas

Esta página actúa más bien como si fuera una agenda sobre lo que se va a tratar hoy. Examinaremos rápidamente todo aquello que puede fallar y, a continuación, estudiaremos detenidamente cada uno de estos errores en páginas posteriores.

- N** El canal puede finalizar. Por ejemplo, debido a que el gestor de cola remoto está terminando. Para solucionar este tipo de problema, puede utilizar el iniciador de canal para volver a poner en marcha los canales.
- O** No se puede realizar la entrega de mensajes. Esto se puede deber a un error de aplicación, por ejemplo, que la cola especificada no es válida, o bien a la existencia de un problema en el sistema, por ejemplo, que la cola esté llena. Para continuar después de este tipo de problema, puede utilizar la cola de mensajes no entregados o puede utilizar la función de reintento de mensaje.
- T** Puede producirse una anomalía en la red, y, en función del tipo de anomalía, existen diferentes formas de que su canal lo detecte y se recupere.
- E** El rendimiento puede ser otro tema importante, por lo que analizaremos algunos de sus aspectos.

S

Reintento de canal



```
DEF CHL ....
SHORTRTY(10) SHORTTMR(60)
LONGRTY(999 999 999) LONGTMR(1200)
```

Reintento de canal - Notas

El mecanismo "catchall" para garantizar que los canales funcionan consiste en que WebSphere MQ reinicia el canal después de una anomalía. Esto se conoce como reintento de canal. El reintento de canal se puede utilizar en el envío de tipos de canales, es decir, emisores, servidores o emisores de clústeres.

N

Todos los canales de emisor deberán estar configurados con valores de reintento adecuados, y el Iniciador de Canal deberá estar en funcionamiento para utilizar el mecanismo de reintento.

O

Los valores de reintento deberán asegurarse de que haya bastante tiempo para solucionar los problemas en la red, por ejemplo, es probable que 2 minutos de reintentos no sean suficientes. Podemos encontrar dos tipos diferentes de valores de reintento: un reintento corto y un reintento largo. Los valores de reintento corto se suelen utilizar para un número pequeño de reintentos frecuentes, es decir, para intentar volver a conectarse al asociado tras cualquier tipo de interrupción temporal. Los valores de reintento largo se suelen utilizar para un gran número de reintentos menos frecuentes, es decir, si el motivo de que el canal haya finalizado es un problema, no se producirá una simple pequeña interrupción.

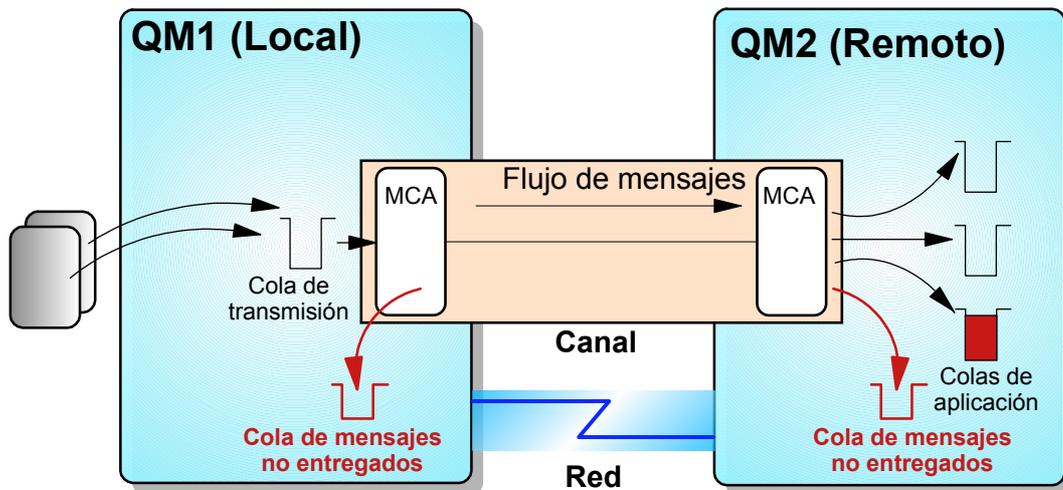
T

E

Observe que los números especificados en los valores de reintentos largos deben bastar para cubrir cualquier pausa, en el caso de que la secuencia de reintentos finalice sin haber realizado una transferencia de mensajes satisfactoria, el canal se tendrá que reiniciar manualmente.

S

Cola de mensajes no entregados



■ Área de almacenamiento para mensajes que, por ejemplo:

- ▶ Están mal direccionados
- ▶ No están autorizados
- ▶ Son demasiado largos

Cola de mensajes no entregados - Notas

Se recomienda que los gestores de colas tengan una cola de mensajes no enviados. Sin una cola de mensajes no enviados, un simple error de programación en la aplicación al especificar un nombre de cola no válido puede hacer que el canal quede inactivo. Obviamente sería inaceptable que diferentes aplicaciones utilizaran el mismo canal.

N Tenga en cuenta que la cola de mensajes no entregados (DLQ) podría utilizarse en ambos extremos del canal. En el extremo emisor, la cola de mensajes no entregados se utilizará si un mensaje en concreto es demasiado grande para que se pueda enviar por el canal. Esto sucede cuando el tamaño del mensaje excede del tamaño del mensaje definido para el canal, aún cuando la cola de transmisión puede aceptarlo. En el extremo receptor, la cola de mensajes no entregados se utilizará cuando el mensaje no se pueda colocar (PUT) en la cola de destino. Esto puede suceder por cualquier motivo, alguno de estos motivos puede ser temporal. Los parámetros de reintento de mensajes (definidos para un canal) y la rutina de salida de reintentos de mensajes están diseñados para tratar estos casos. En un momento, examinaremos los reintentos de mensajes.

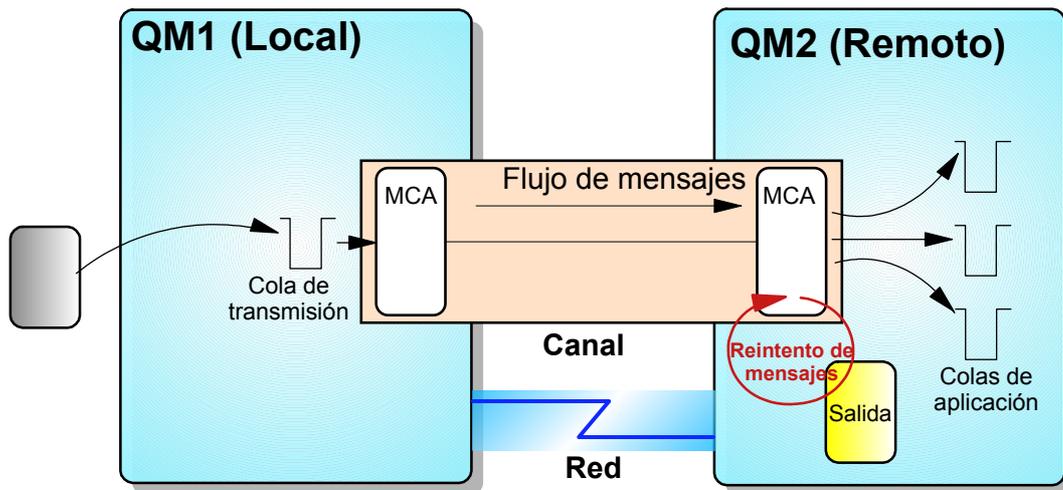
O Debe tener en cuenta que el uso de la cola de mensajes no entregados (DLQ), en ambos extremos de un canal, puede afectar al orden de entrega de los mensajes. Si hay algún problema con un mensaje que forme parte de una secuencia, este mensaje se enviará a la cola de mensajes no entregados y se interrumpirá la secuencia de mensajes.

T Algunas aplicaciones 'lentas' requieren que los mensajes se entreguen siempre en la misma secuencia que se enviaron. Estas aplicaciones necesitan que no haya cola de mensajes no entregados (DLQ). La consecuencia de no tener una cola de mensajes no entregados (DLQ) es que un mensaje corrupto en un canal lo interrumpirá y, por lo tanto, impedirá que cualquier aplicación reciba sus mensajes. Por este motivo, se recomienda que las instalaciones tengan una cola de mensajes no entregados (DLQ) y que las aplicaciones se modifiquen para que puedan manejar los mensajes que ocasionalmente puedan enviarse fuera de la secuencia. Esta modificación no es complicada y existen varias técnicas que se pueden emplear, como por ejemplo el uso del ID de correlación como si fuera un contador, teniendo una cola secundaria o bien utilizando una agrupación de mensajes.

E

S

Reintento de mensajes



```
DEF CHL ....
MRRTY(10) MRTMR(1000)
MREXIT(nombre_salida_reintento_mensaje) MRDATA()
```

Reintento de mensajes - Notas

Considere la opción de utilizar el reintento de mensajes, especialmente si el canal está funcionando sin una cola de mensajes no entregados.

N El mecanismo de reintento de mensajes se puede utilizar con o sin una rutina de salida. Sin una rutina de salida, el MCA receptor simplemente reintenta la colocación que ha fallado en cada intervalo. Los valores del tiempo que transcurre entre reintentos y el número de reintentos se extraen de la definición de canal. Si se utiliza con una rutina de salida, la salida sobrepasa estos números, pero puede cambiarlos por cualquier valor. El uso de una rutina de salida con el mecanismo de reintentos de mensaje permite procesar las anomalías de forma más inteligente. Por ejemplo, podrá redireccionar los mensajes si no se pueden transferir a la cola de destino.

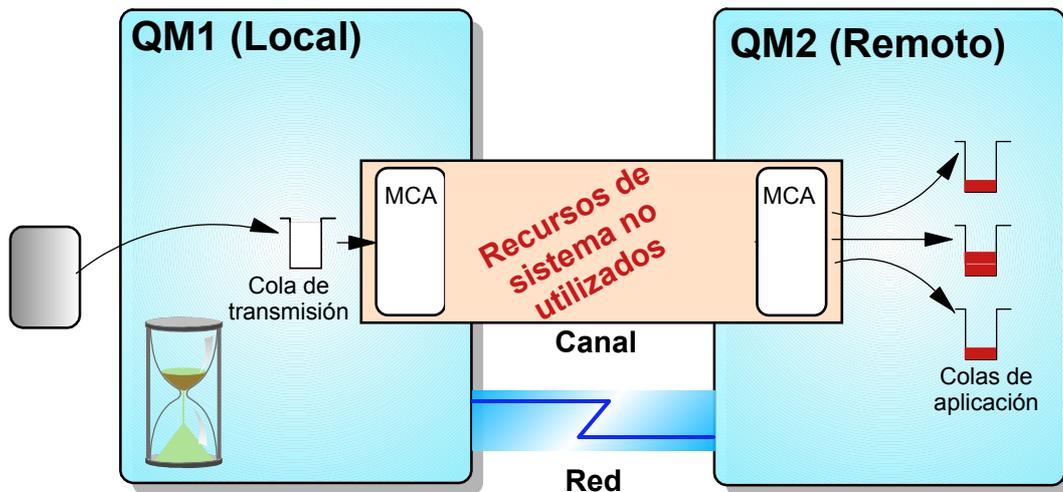
O El uso de una rutina de salida permite la opción de reintento por mensaje.

T Actualmente, z/OS no ofrece soporte para el reintento de mensajes.

E

S

Intervalo de desconexión



DEF CHL DISCONT(6000)

- **Combinar con la activación**

Intervalo de desconexión - Notas

El intervalo de desconexión o DISCONT es el atributo de canal en un canal emisor que se utiliza para controlar el tiempo que un canal debería seguir en funcionamiento aunque no hubiera mensajes para enviar.

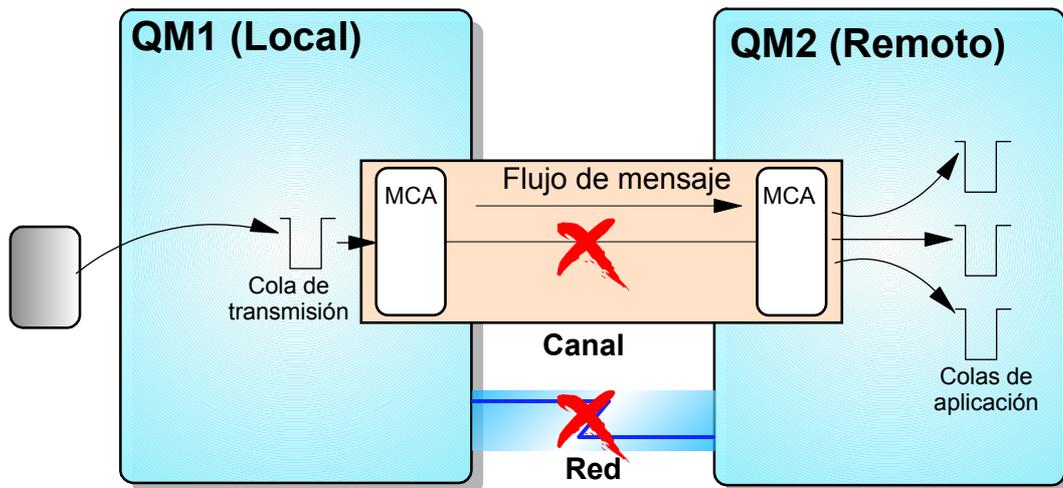
- N** El valor que se establece depende de varios criterios.
- La cantidad de canales y la importancia de los recursos de sistema
 - La fiabilidad de la red
 - El coste de las sesiones de comunicación

O El valor idóneo debería durar más que el tiempo medio empleado en el envío de mensajes a ese asociado. Esto significa que el canal se desconectará cuando no haya trabajo para entregar al asociado.

- T** Debería combinar el uso de un intervalo de desconexión de su canal con la activación del canal para que se reinicie. De este modo, el canal se iniciará automáticamente cuando haya trabajo y finalizará automáticamente cuando no haya más trabajo.

E
S

Anomalía en la red



- Pulsaciones
- Función Keepalive de TCP/IP
- Adoptar MCA

Anomalía en la red - Notas

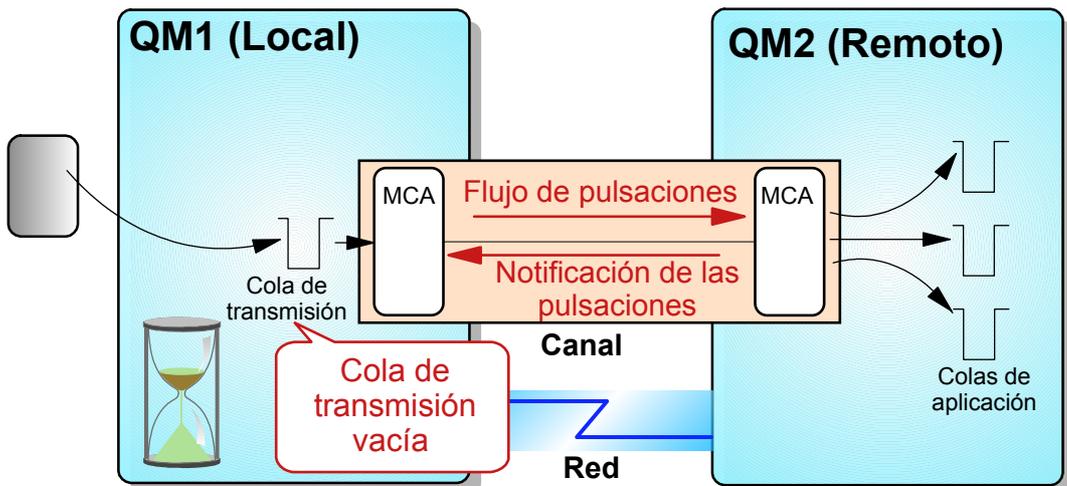
Cualquier anomalía que se produzca en la red y haya sido reportada al canal provocará que el canal finalice. Si la anomalía en la red no se reporta al canal, éste no finalizará y puede que no advierta la anomalía en la red.

- N** El canal depende de las funciones de comunicación del nivel inferior para informar sobre cualquier pausa en la red. Por ejemplo, el estado latente de un canal RECEPTOR está en una llamada al receptor. Si no se devuelve la llamada al receptor, el canal no sabrá que la conexión con el asociado se ha perdido. Una de las dificultades con MCAs consiste en identificar en el momento adecuado cuándo ha fallado la red, ya que los protocolos de comunicación no siempre devuelven los errores en el momento preciso. Normalmente LU6.2 se considera mucho mejor para esto que TCP/IP (es un protocolo más complejo).
- O**
- T** Los diferentes avances en los MCAs han mejorado considerablemente la situación. Examinaremos unos cuantos en las próximas páginas.

E

S

Pulsaciones



DEF CHL HBINT(300)

**Almacenamientos
intermedios ✓
libres
Cerrar colas ✓**

Pulsaciones - Notas

**N
O
T
E
S**

El intervalo de pulsaciones o HBINT es el atributo de canal que controla la frecuencia en que el extremo emisor del canal deberá comprobar que el receptor está bien cuando no haya mensajes para enviar. El valor se especifica en segundos en ambos extremos del canal, y ambos negocian un valor apropiado que se utilizará durante el inicio del canal. Este valor será el intervalo menos frecuente de los dos que se hayan especificado. Si uno o los dos extremos especifican 0, lo cual significa que no hay pulsaciones, este valor, siendo el valor de frecuencia mínimo, será el que se utilizará.

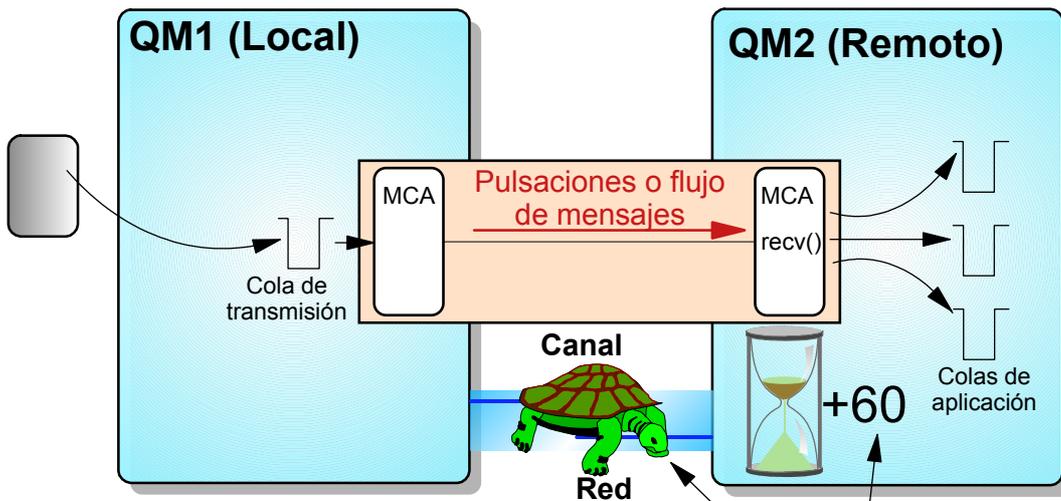
Los canales por omisión se definen con un intervalo de pulsación de 300 segundos (5 minutos). Se optó por un valor relativamente alto como valor por omisión con el fin de que hubiese muy poco impacto en la red. En la práctica, un valor inferior de unos 60 segundos es mejor en muchos casos.

Esta pulsación es, en realidad, un mensaje corto que se envía al MCA receptor (o CLNTCONN durante MQGET) para comprobar que todavía está disponible. Sólo se envía cuando no hay ninguna otra actividad en el canal durante los segundos de pulsación como mínimo. Dichos mensajes resuelven dos problemas: los MCAs emisores puede que ahora reciban un error inmediato de la red, en lugar de estar en su MQGET; asimismo, permiten que los MCAs receptores realicen algunos procesos como por ejemplo, comprobar la finalización del gestor de cola antes de enviar su respuesta.

Además, cuando el canal no transfiere mensajes regularmente, el canal decide no mantener su antememoria de almacenamiento y las colas. Por lo tanto, durante las pulsaciones, un canal liberará los almacenamientos intermedios innecesarios, áreas de mensajes y cerrará las colas de la antememoria.

En cuanto al rendimiento, el valor de las pulsaciones no debe ser muy pequeño. Pero tenga en cuenta que los valores de pulsaciones bajos no conllevan un tráfico de red adicional, siempre que el canal se mantenga ocupado. Suponga que una aplicación envía un mensaje cada dos segundos por un canal, incluso un valor bajo de pulsaciones de 3 segundos nunca dejará que se envíe una pulsación, ya que siempre habrá un mensaje para enviar.

Tiempo de espera de recepción



```
DEF CHL .... HBINT(300)
MQRCVBLKTO: +60
RCVTIME=+60
```

Reflejo del tiempo de latencia de la red ✓

Tiempo de espera de recepción - Notas

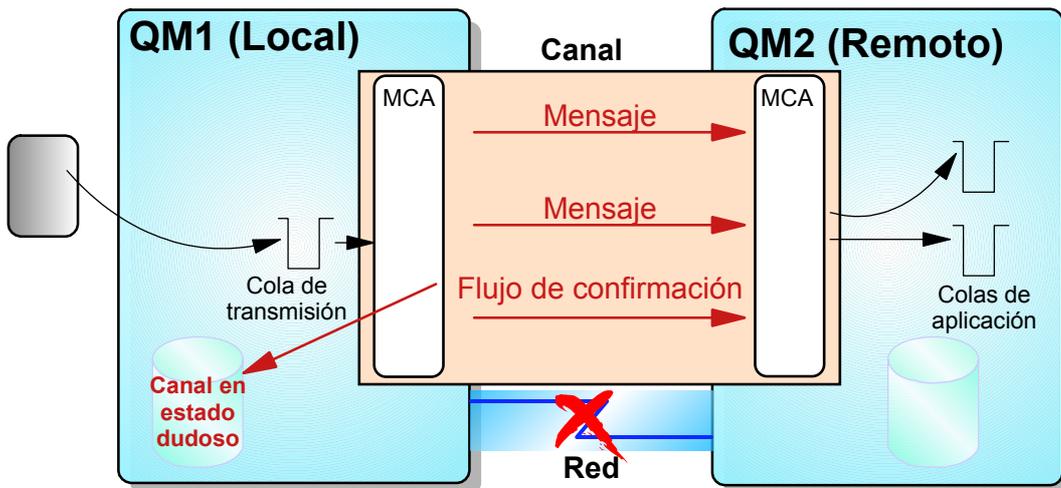
N
O
T
E
S

Como ya hemos mencionado, el intervalo de pulsaciones que utiliza el extremo emisor del canal para enviar periódicamente este mensaje de pulsación es un valor negociado, de tal modo que ambos extremos conocen el valor. Sin una pulsación, el extremo receptor del canal no puede predecir cuándo debería llegar el próximo fragmento de datos al canal, ya que la llegada de los mensajes no está determinada. No obstante, con una pulsación el receptor 'sabe' que dentro del intervalo de pulsaciones debería llegar un mensaje o una pulsación. Si esto no ocurre, significa que la sesión de comunicación se ha interrumpido. Esto se utiliza como base para el reconocimiento de anomalías en la red TCP/IP. Para los canales CLNTCONN, este tiempo de espera se utiliza también mientras se espera una respuesta de un mandato MQI (excepto cuando SVRCONN se encuentra en z/OS, el cual no da soporte a los diferentes clientes de pulsaciones).

De hecho, el receptor excederá el tiempo de espera si no se reciben datos en el doble de veces del intervalo de pulsaciones si el intervalo de pulsaciones negociado es inferior a 60 segundos, o 60 segundos después del intervalo de pulsaciones negociado si es mayor o igual a 60 segundos, por omisión, antes de dar por hecho que se producido una anomalía en la comunicación. De hecho, este calificador puede ajustarse para que refleje el tiempo de latencia de la red, en las plataformas distribuidas, cambiando la variable de entorno MQRCVBLKTO. Hay otra variable de entorno MQRCVBLKMIN que permite establecer un valor mínimo para calcular el tiempo de espera en función de las pulsaciones. El comportamiento se puede establecer en el sistema z/OS en V5.3.1 con el uso de RCVTIME atributos XPARM para especificar que el tiempo de espera y RCVTMIN se establezcan en el valor más bajo.

Por lo tanto, con un valor pequeño se detectarán con más rapidez las anomalías pero también puede resultar más caro en cuanto al tráfico en la red y el uso de recursos de la CPU si su canal se carga ligeramente.

Entrega asegurada



- **Datos de sincronización grabados en el disco en los dos extremos**
 - ▶ Permite volver a sincronizar después de una anomalía en la red
 - ▶ Los mensajes en un lote en estado dudoso no se pueden reasignar mediante un algoritmo de clústeres

Entrega asegurada - Notas

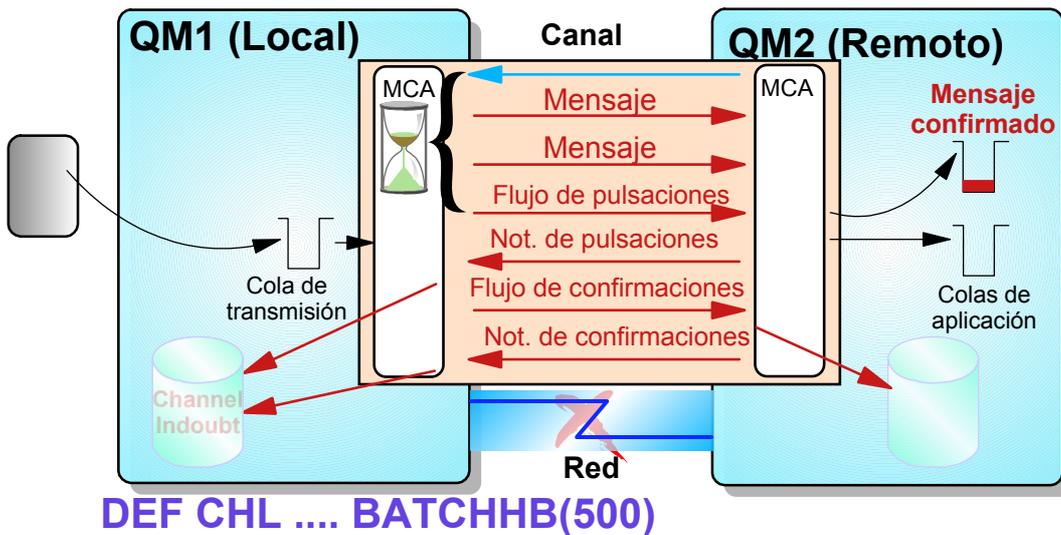
N
O
T
E
S

Aquí podemos observar los dos extremos del canal transfiriendo un lote de mensajes. Tenga en cuenta cómo el MCA graba datos en el disco al final del lote. Esto sólo se realiza para los lotes recuperables. Los datos que se graban contienen el ID de la transacción que se ha utilizado en el extremo emisor para recuperar todos los mensajes. Una vez que el extremo emisor haya emitido un flujo de 'confirmación' a su asociado, estará 'dudoso' hasta que reciba una respuesta. En otras palabras, el canal emisor no puede asegurar si los mensajes se han entregado correctamente. Si se produce una anomalía en la comunicación durante esta fase, el canal finalizará en estado dudoso. Cuando su asociado se vuelva observará que hay datos almacenados que han quedado pendientes con su asociado y preguntará al otro canal si el último lote de mensajes fue enviado o no. Con la respuesta que envía el asociado, el canal podrá confirmar o restituir los mensajes en la cola de transmisión.

Estos datos de sincronización se pueden ver al emitir el mandato DIS CHSTATUS(*) SAVED. Los valores que se muestran deberían ser iguales en ambos extremos del canal.

Tenga en cuenta que si el canal se reinicia en estado "dudoso", éste resolverá la situación automáticamente. No obstante, sólo podrá hacerlo si está comunicado con el mismo asociado. Si se cambian los atributos de canal o bien otro gestor de colas se apodera de la dirección IP o si se inicia un canal diferente que sirva a la misma cola de transmisión, el canal finalizará inmediatamente con un mensaje que indique que todavía está en estado "dudoso" en otro gestor de colas. El usuario deberá iniciar el canal direccionándolo al gestor de colas correcto o bien solucionar dicho estado "dudoso" manualmente, emitiendo el mandato RESOLVE CHANNEL. Tenga en cuenta que en este caso el usuario debe utilizar la salida de DIS CHS(*) SAVED para asegurarse de que se ha elegido correctamente la acción COMMIT o BACKOUT.

Pulsaciones por lotes



- Si el tiempo transcurrido desde la última comunicación es mayor al intervalo, se enviará el flujo de pulsaciones
Refleja la estabilidad de la red ✓

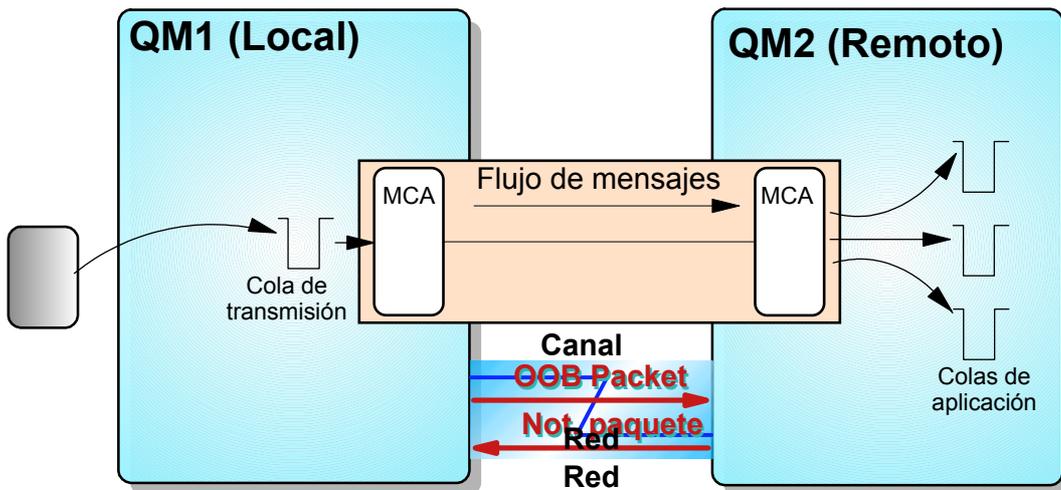
Pulsaciones por lotes - Notas

El problema con los canales que finalizan en estado "dudoso" se distingue con el algoritmo de clústeres. Si un canal de clústeres termina anormalmente, los mensajes que se tenían que enviar a través de este canal se reasignarán a otro canal y a otra instancia de la cola de clúster de destino. No obstante, no se podrán reasignar si son parte de un lote de mensajes que el canal emisor duda sobre si el canal receptor los ha recibido o no.

Para reducir la posibilidad de que un canal finalice en estado "dudoso", se ha añadido un flujo adicional al protocolo del canal. Dicho flujo se conoce como una pulsación por lotes. Este flujo se enviará una vez que los mensajes para el proceso por lotes se haya enviado, pero antes de que el canal se ponga en estado "dudoso" y envíe el flujo de confirmación. Cuando se haya recibido una respuesta del extremo receptor del canal, podremos estar prácticamente seguros de que se ha completado el proceso por lotes. Ahora aparece una ventana realmente pequeña donde el canal receptor puede finalizar y dejarnos en estado "dudoso". Sin el flujo de pulsaciones por lotes esta ventana es mucho más grande (desde la última vez que se recibió el flujo del asociado).

Para determinar si este flujo de pulsaciones por lotes se envía, anotaremos cuándo recibimos por última vez un flujo de nuestro asociado. Si el tiempo transcurrido no es superior al intervalo especificado en el BatchHeartbeat o en el parámetro BATCHHB en el canal, enviaremos el flujo, de lo contrario no se enviará. El intervalo especificado es por lo tanto un reflejo de la estabilidad de la red. Este intervalo se especifica en milisegundos.

Función Keepalive de TCP/IP



▪ QM.INI **KEEPALIVE=YES**

▪ XPARM **TCPKEEP=YES**

DEF CHL ... KAIN(360) (sólo z/OS)

Función Keepalive de TCP/IP - Notas

TCP/IP también tiene su propio protocolo de pulsaciones, denominado KeepAlive, que permite reconocer las anomalías en la red. Se recomienda el uso de KeepAlive.

N Únicamente en z/OS, se puede configurar como un parámetro de canal, permitiendo que los diferentes canales tengan valores de tiempo de espera específicos o que simplemente utilicen los valores del sistema. Para utilizar dicho parámetro de canal, KeepAlive tendrá que estar conectado a TCP/IP.

O En todas las plataformas, se puede configurar según el gestor de colas en WebSphere MQ el archivo qm.ini o equivalente. Los valores actuales que se utilizan, en este caso, para los temporizadores y contadores se establecen para toda la pila TCP/IP de la máquina. Los valores típicos para detectar las anomalías en la red es de 2 horas por omisión. Probablemente es demasiado alto para un entorno de WebSphere MQ, pero otras aplicaciones de la misma máquina puede que también quieran utilizar KeepAlive, por lo tanto los valores tendrán que establecerse en un término medio para satisfacer los requisitos de dichas aplicaciones.

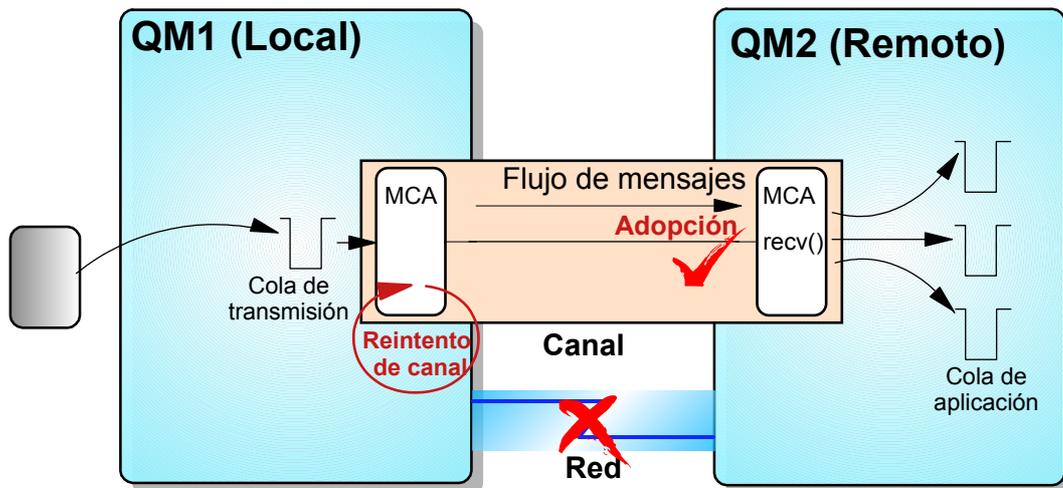
T Tendrá que consultar los manuales de TCP/IP para averiguar cómo se configura KeepAlive. Para sistemas AIX, utilice el mandato "no". Para sistemas HP-UX, utilice el mandato "netttune". Para sistemas Windows NT, tiene que editar el registro. En sistemas z/OS, actualice su conjunto de datos TCP/IP PROFILE y añada o cambie el parámetro INTERVAL en el apartado TCPCONFIG.

E Nota: en sistemas z/OS, se recomienda TCP/IP APAR PQ75195

S El uso de pulsaciones y del tiempo de espera de recepción eliminan la necesidad de utilizar KeepAlive cuando ambos extremos soportan ambas opciones; no obstante, se puede seguir utilizando KeepAlive. Cuando se comunique con las aplicaciones de WebSphere MQ que no soportan las pulsaciones ni las lecturas sin bloqueos, deberá seguir utilizando KeepAlive.

Se recomienda el uso de Keepalive para canales SVRCONN aunque las pulsaciones de cliente se encuentren disponibles, sólo se utilizan durante los MQGETs.

Adoptar MCA



- **QM.INI** **AdoptNewMCA, AdoptNewMCATimeout, AdoptNewMCACheck**
- **XPARM** **ADOPTMCA, ADOPTCHK**

AdoptMCA - Notas

Si un canal falla (por ejemplo, se produce un error de comunicación), no hay garantía alguna de que ambos extremos del canal detecten la anomalía al mismo tiempo. A menudo el extremo emisor detecta primero la anomalía ya que es el extremo que intenta enviar mensajes por el socket. Por lo tanto, el emisor intentará volver a conectarse al gestor de cola de destino. Cuando se establece la conexión con la máquina de destino, el gestor de cola descubrirá que ya se está ejecutando un canal con ese mismo nombre y desde esa ubicación. Por consiguiente, la nueva conexión será rechazada.

Al configurar `AdoptNewMCA` puede indicar al gestor de cola que si se realiza una nueva conexión y ya hay un canal con ese nombre ejecutándose desde la misma dirección de red y desde el mismo gestor de cola, se deberá finalizar el canal existente y sustituirse por esta nueva instancia. Es decir, la instancia de canal deberá ser 'adoptada' por la nueva conexión.

Los diferentes parámetros de configuración en la sección de los canales de QM.INI o equivalente (o en z/OS utilice XPARMS) permite cambiar ligeramente el comportamiento de adopción ('Adopt'). Establezca la lista de los tipos de canal que pueden adoptarse de esta manera para activar la adopción.

`AdoptNewMCA=SVR,SDR,RCVR,CLUSRCVR,ALL,FASTPATH`

`ADOPTMCA=SÍ/NO` (en z/OS)

Establezca el intervalo de espera entre la finalización del canal 'adoptado' de forma correcta y la finalización inmediata.

`AdoptNewMCATimeout=60`

Evidentemente, el nombre del canal debe coincidir pero además puede especificar que la dirección de red deba ser la misma y también el nombre del gestor de cola remoto. Esto impedirá que una conexión no autorizada desde una máquina diferente en la red adopte canales que ya están ejecutándose. Establezca qué atributos deben coincidir entre las conexiones adoptadas y las nuevas.

`AdoptNewMCACheck=QM,ADDRESS,NAME,ALL`

`ADOPTCHK=NONE/QMNAME/NETADDR/ALL` (en z/OS)

Rendimiento

- **Algunas preguntas para reflexionar**
 - ▶ ¿Cuál es la velocidad de llegada de la cola de transmisión?
 - ▶ ¿Cuáles son los tiempos de respuesta requeridos?
 - ▶ ¿Cuánto tarda un lote en transmitir?
 - ▶ ¿Cuál es el tiempo de inicio de un canal?
- **Las respuestas dependen del entorno**
 - ▶ Uso de la aplicación
 - ▶ Sistema operativo/Hardware
 - ▶ Ancho de banda de la red
 - ▶ Tamaño de los mensajes

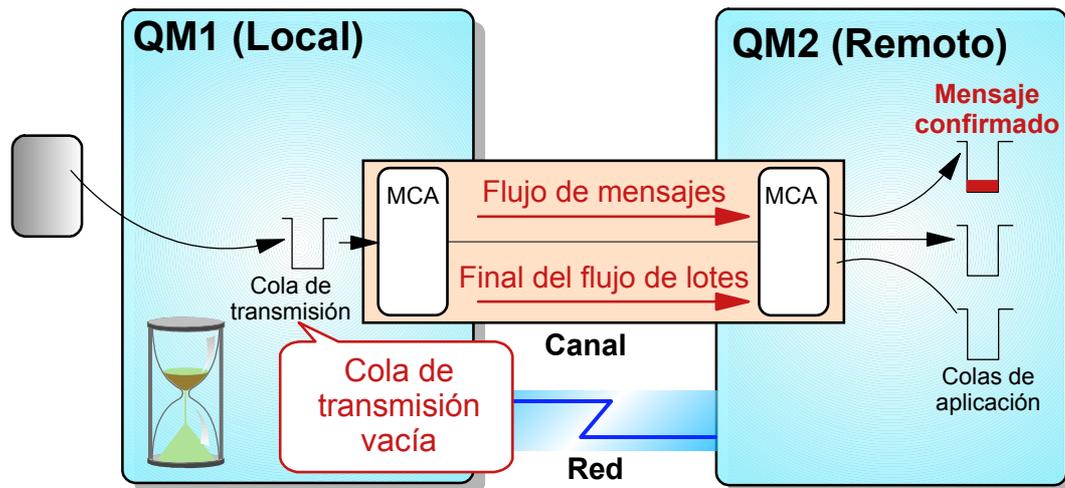


Configuración del rendimiento

- **Utilizar el intervalo de lotes (BatchInterval) para mantener los lotes abiertos**
 - ▶ si la velocidad de llegada es ligeramente más lenta que la del envío del lote completo
 - ▶ este uso puede incrementar el tiempo de latencia, pero mejorar el rendimiento total
- **Establecer el intervalo de desconexión a un valor bastante alto para que los canales se mantengan activos**
 - ▶ Reduce el número de veces que un canal se inicia/para/reinicia
 - ▶ Salvo que las cargas de red sean por "sesiones activas" o que esté alcanzando el límite máximo de canales
- **Tres casos**
- **Utilizar enlaces fiables**
- **Consultar las guías de planificación de la capacidad**
 - ▶ SupportPac MP00
 - ▶ SupportPac MP16



Intervalo de lote



- Duración mínima del lote de mensajes

DEF CHL BATCHINT(100)

► Los mensajes recuperables no llegarán antes que BATCHINT

Intervalo de lote - Notas

**N
O
T
E
S**

Un lote de mensajes recuperables finaliza tan pronto como la cola de transmisión esté vacía. Esto puede conllevar más actividad en la línea y en el disco de la necesaria. Piense en una aplicación que envía 3 mensajes permanentes a la cola de transmisión: existen muchas posibilidades de que el canal recupere el primer mensaje antes de que el segundo haya sido transferido. Por consiguiente, el canal completará el lote, y el segundo y tercer mensaje irán en un segundo lote.

Si se añade un pequeño retraso al lote, supongamos, BATCHINT(100), los 3 mensajes se enviarán en el primer lote con sólo la mitad de actividad de disco. Tenga en cuenta que puede parecer que el mensaje tarda más en enviarse, ya que hay un retraso de 100 ms. como mínimo. BATCHINT no tiene ningún efecto en mensajes no permanentes NPMSPEED(FAST), ya que estos mensajes se transfieren fuera del punto de sincronismo. Generalmente las aplicaciones envían aquellas consultas de bases de datos que son urgentes como no permanentes y las actualizaciones de bases de datos que no son urgentes como mensajes permanentes.

El efecto en la red y el valor de BATCHINT depende en gran medida de la red de mensajería de MQ y del perfil de mensajería. Cuanto mayor sea el número de canales que tenga la máquina, mayor será el ahorro potencial.

Una manera de determinar si BATCHINT será beneficioso consiste en observar el tamaño de lote efectivo. Emita un mandato DISPLAY CHSTATUS y divida el número de mensajes por el número de lotes. Si dicho número es muy bajo, (pongamos entre 1 y 2), el canal tendrá que confirmar cada 1 ó 2 mensajes, lo cual es ineficaz y puede aprovecharse de BATCHINT. Tenga en cuenta que esto es sólo una indicación, ya que si todos los mensajes fueran no permanentes, BATCHINT no serviría de nada.

Otro aspecto a tener en cuenta es el uso del disco para los procesos de canal. Si observa que se utiliza mucho disco, existen muchas posibilidades de que BATCHINT reduzca este coste y mejore el rendimiento global.

Caso 1

■ Transferencia masiva de la cola de transmisión precargada

- ▶ **Utilice un valor de BatchSize grande (se da por bueno un valor por omisión de 50),**
 - salvo que se utilicen mensajes muy grandes (coste de retransmisión)
- ▶ **El valor de BatchInt debería ser 0**



© 2003 IBM Corporation

Caso 1 - Notas

Nuestro primer caso de ajuste de rendimiento se trata de una transferencia masiva de los mensajes que se encuentran en una cola de transmisión precargada.

N El objetivo aquí consiste en enviar los mensajes en grandes lotes, reduciendo en consecuencia el coste final del proceso por lotes por mensaje. Por lo tanto, sería conveniente utilizar un tamaño de lote grande (se considerará bueno un valor por omisión de 50) y el intervalo de lote se debería establecer en 0. No es necesario, en este caso, mantener el lote abierto, ya que todos los mensajes están precargados en la cola de transmisión.

O El equilibrio en el rendimiento reside en la cantidad de mensajes que debería enviar en un lote para obtener un bajo coste de final de proceso de lote por mensaje, considerando el coste de retransmisión que supondría reenviar un lote entero, en el caso de que se perdiera la conexión con la red. Si transfiere mensajes muy grandes, puede que se alcance dicho equilibrio con un tamaño de lote inferior a 50.

T

E

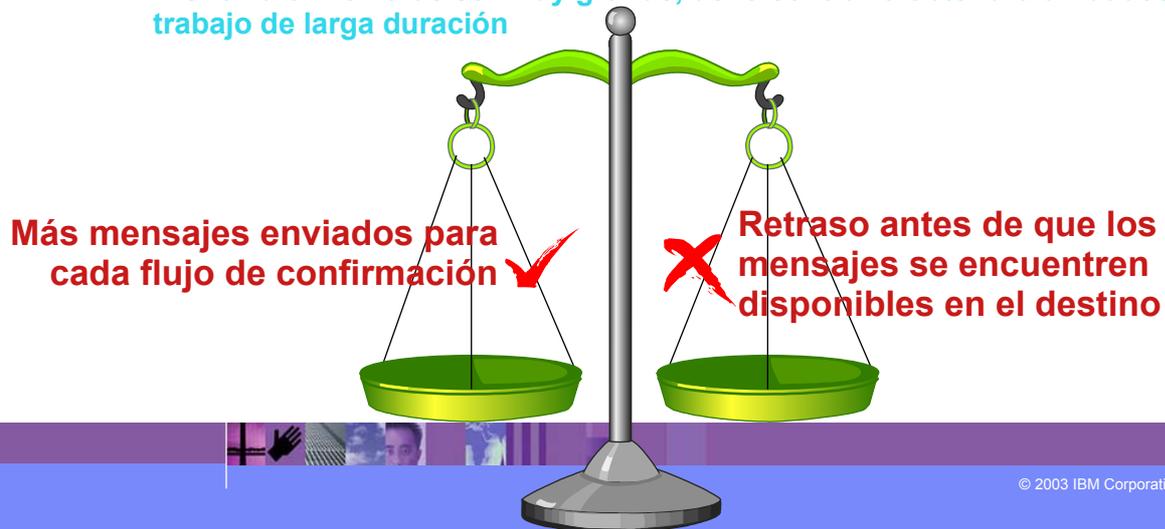
S

© 2003 IBM Corporation

Caso 2

■ Transferencia lenta en los procesos diferidos

- ▶ Los números exactos dependen de la velocidad de llegada
- ▶ Intente obtener lotes altos
- ▶ Establezca el valor BatchInt a un retraso aceptable antes de realizar la transmisión, por ejemplo 1 minuto
- ▶ Dicho valor no ha de ser muy grande, de lo contrario obtendrá unidades de trabajo de larga duración



Caso 2 - Notas

Nuestro segundo caso de ajuste en el rendimiento se trata de una transferencia lenta de mensajes para procesos diferidos.

N El objetivo en este caso consiste en lograr un tamaño de lote razonable del mensaje, reduciendo por lo tanto el coste del final del proceso por lotes por mensaje. Este es un uso clásico del intervalo de lote. Los mensajes llegan a una velocidad ligeramente más lenta que la velocidad del canal para procesarlos. Al mantener el lote abierto un poco más, se consigue que se procesen en un lote más mensajes juntos.

O El equilibrio en el rendimiento reside en la cantidad de mensajes que se pueden enviar en un lote, teniendo en cuenta que los mensajes recuperables parecerá que tarden más en entregarse, ya que no se confirman en las colas de destino hasta que el lote finaliza, es decir después de un intervalo de lote como mínimo.

T Si el tiempo de entrega no es un problema, asegúrese de no establecer un intervalo de lote demasiado grande, ya que corre el riesgo de generar unidades de trabajo de larga duración.

E

S

Caso 3

■ Petición/Respuesta síncrona

- ▶ Es posible que no quiera utilizar BatchInt si el tiempo de respuesta es clave
- ▶ Si el volumen permanente es bajo, considere un BatchSize más pequeño
- ▶ Para los flujos de mensajes no permanentes, utilice NPMSPEED(FAST) y un BatchInt distinto de cero



Caso 3 - Notas

Nuestro tercer caso de ajuste en el rendimiento se trata de un modelo de petición/respuesta síncrona donde el tiempo de respuesta no importa.

N Esta situación es similar a la del caso 2, salvo que esta vez la balanza se inclina más hacia un tiempo de respuesta más rápido en lugar de más mensajes en un lote.

O Si el volumen de mensajes permanentes es bajo, es posible que quiera utilizar un tamaño de lote más pequeño y no utilizar en modo alguno el intervalo de lote, o bien utilizar un valor bastante bajo (pocos segundos) para garantizar que los mensajes se encuentran disponibles sin retrasos.

T Si tiene mensajes no permanentes, deberá utilizar la velocidad de mensajes no permanentes o el valor FAST para NPMSPEED, y establecer un intervalo de lote distinto de cero, ya que los mensajes no permanentes que se entregan por un canal rápido estarán disponibles inmediatamente. Se sigue utilizando el valor del tamaño de lote, aunque dicho valor no afecta al tiempo de respuesta.

Algunas cifras de ejemplo:

E A una velocidad prevista de 30 milisegundos de mensajes no permanentes, establezca BatchInt=2000 (2 segundos) y BatchSize=50 para un tamaño de lote real de 50.

S

Enlaces fiables

- **Único proceso**
 - ▶ Para MCA y el agente de gestor de cola local
 - ▶ Sin IPC por encima para mandatos MCA MQI
 - importante para mensajes no permanentes
- **Configurado en el archivo ini del gestor de cola**
 - ▶ Sección de canales

MQIBINDTYPE=FASTPATH

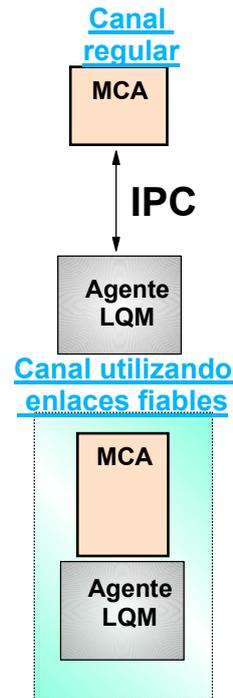
- **No utilizar si deja de probar el canal**
- **Tenga cuidado al emitir**

STOP CHL() MODE(FORCE)

(versiones anteriores a la V5.3)

STOP CHL() MODE(TERMINATE)

V5.3



Enlaces fiables - Notas

N

En el conjunto de productos distribuidos, todas las funciones del gestor de cola son en realidad llevadas a cabo por un proceso diferente de la aplicación conocida como agente. Cada llamada API se envía al proceso de agente a través de algún mecanismo de IPC. Esto se realiza por motivos de aislamiento. Una aplicación no tiene acceso directo a los recursos o a la memoria del gestor de cola. No obstante, cada llamada MQI es un conmutador de proceso que tiene un impacto sobre el rendimiento. Para un mensaje no permanente, alrededor del 60% del coste de la llamada radica en el conmutador de proceso.

O

Todas las aplicaciones pueden optar por ejecutarse en modalidad 'fiable', lo cual significa que las funciones del gestor de cola se ejecutarán directamente en línea con la aplicación. Se recomienda que las aplicaciones en general no se ejecuten en esta modalidad. Los canales, no obstante, son código de producto y, por lo tanto, pueden ejecutarse como aplicaciones 'fiables'. Por omisión, se ejecutarán como aplicaciones normales pero estableciendo MQIBindType=FASTPATH en la sección de los canales en el archivo QM.INI, todos los canales se ejecutarán de forma fiable.

T

Se deberá prestar atención a la rutina de salida de los canales. Dado que las rutinas de salida de canal se ejecutan en línea con MCA, las rutinas de salida ahora estarán en el mismo espacio de memoria que el agente de gestor de cola. Se deberán probar detenidamente las rutinas de salidas de canal antes de ejecutarlas en esta modalidad.

E

S

Se deberá tener cuidado al utilizar el mandato STOP MODE(FORCE) en los releases anteriores a V5.3, y en el release V5.3 se deberá tener cuidado al utilizar el mandato STOP MODE(TERMINATE), ambos podrían dañar o dejar inactivo el gestor de cola ya que es posible que la limpieza no se haya llevado a cabo completamente cuando el canal ha terminado.

Referencia de parámetros

Canal	Descripción	
HBINT	Intervalo de pulsaciones	
BATCHSZ	Número máximo de mensajes en un lote	
BATCHINT	Duración mínima de un lote de mensajes	
DISCINT	Intervalo después del cual el canal finalizará	
SHORTRTY, SHORTTMR LONGRTY, LONGTMR	Contadores y temporizadores de reintentos de canal	
MRRTY, MRTMR MREXIT, MRDATA	Contadores y temporizadores de reintentos de mensaje y salida (no en z/OS)	
QM.INI - Sección de canal	XPARMS	Descripción
AdoptNewMCA	ADOPTMCA	Adoptar tipos de canales
AdoptNewMCATimeout		Adoptar el tiempo de espera de desactivación
AdoptNewMCACheck	ADOPTCHK	Adoptar la comprobación de parámetros
KeepAlive	TCPKEEP	Uso de la función Keepalive
MQIBindType		Enlace de FASTPATH del canal
Variables de entorno	XPARMS	Descripción
MQRCVBLKTO	RCVTIME	Tiempo de espera de recepción
MQRCVBLKMIN	RCVTMIN	Tiempo de espera mínimo de recepción



Referencias sobre parámetros - Notas

En esta tabla se listan aquellos atributos de un canal que afectan a las funciones de rendimiento y recuperación.

N

Algunos atributos forman parte de la auténtica definición de canal y otros se establecen en otras ubicaciones, como por ejemplo en un archivo INI.

En el manual InterCommunication podrá encontrar más detalles sobre los parámetros en la definición de canal.

O

En las plataformas distribuidas, esos parámetros se encuentran en el archivo QM.INI en la sección de los canales o, en Windows, se implementan en el Registro

Clave : HKEY_LOCAL_MACHINE/SOFTWARE/IBM/MQSeries/QueueManager/<QMNAME>

o utilice los servicios de MQSeries

T

Los parámetros de iniciador de canal ofrecen algunos de los mismos parámetros que en z/OS se pueden encontrar en el archivo QM.INI en las plataformas distribuidas. Los XPARMS se generan con la macro CSQ6CHIP.

E

S

