9.3

IBM MQ Panoramica tecnica

Nota

Prima di utilizzare queste informazioni e il prodotto che supportano, leggere le informazioni in “Informazioni particolari” a pagina 313.

Questa edizione si applica alla versione 9 release 3 di IBM® MQ e a tutte le successive release e modifiche se non diversamente indicato nelle nuove edizioni.

Indice

IBM MQ - Sommario tecnico

Utilizzare IBM MQ per collegare le applicazioni e gestire la distribuzione delle informazioni nell'organizzazione.

IBM MQ consente ai programmi di comunicare tra loro attraverso una rete di componenti diversi (processori, sistemi operativi, sottosistemi e protocolli di comunicazione) utilizzando un'API (application programming interface) coerente. Le applicazioni progettate e scritte utilizzando questa interfaccia sono note come applicazioni di accodamento messaggi.

Utilizzare i seguenti argomenti secondari per informazioni sull'accodamento dei messaggi e su altre funzioni fornite da IBM MQ.

Concetti correlati

Introduzione a IBM MQ

Dove trovare i requisiti del prodotto e le informazioni di supporto

Attività correlate

Pianificazione di un'architettura IBM MQ

Riferimenti correlati

“Caratteristiche principali e vantaggi dell'accodamento dei messaggi” a pagina 7

Queste informazioni evidenziano alcune funzioni e vantaggi dell'accodamento dei messaggi. Descrive funzioni quali la sicurezza e l'integrità dei dati dell'accodamento dei messaggi.

Introduzione all'accodamento dei messaggi

I prodotti IBM MQ consentono ai programmi di comunicare tra loro attraverso una rete di componenti diversi (processori, sistemi operativi, sottosistemi e protocolli di comunicazione) utilizzando un'API (application programming interface) coerente.

Le applicazioni progettate e scritte utilizzando questa interfaccia sono note come applicazioni di *accodamento messaggi* , poiché utilizzano lo stile di *messaggistica* e *accodamento* :

- La messaggistica significa che i programmi comunicano inviando reciprocamente i dati nei messaggi piuttosto che chiamandosi direttamente.
- L'accodamento significa che i messaggi vengono inseriti nelle code nella memoria, consentendo ai programmi di essere eseguiti indipendentemente l'uno dall'altro, a velocità e tempi diversi, in ubicazioni diverse e senza avere una connessione logica tra loro.

L'accodamento dei messaggi è stato utilizzato nell'elaborazione dei dati per molti anni. È più comunemente usato oggi nella posta elettronica. Senza accodamento, l'invio di un messaggio elettronico su lunghe distanze richiede che ogni nodo dell'instradamento sia disponibile per l'inoltro dei messaggi e che i destinatari siano collegati e consapevoli del fatto che si sta tentando di inviare loro un messaggio. In un sistema di accodamento, i messaggi vengono memorizzati su nodi intermedi finché il sistema è pronto ad inoltrarli. Alla loro destinazione finale sono memorizzati in una casella di posta elettronica fino a quando il destinatario è pronto a leggerli.

Anche così, molte transazioni aziendali complesse vengono elaborate oggi senza accodare. In una rete di grandi dimensioni, il sistema potrebbe mantenere molte migliaia di connessioni in uno stato di pronto utilizzo. Se una parte del sistema subisce un problema, molte parti del sistema diventano inutilizzabili.

È possibile pensare all'accodamento di messaggi come posta elettronica per programmi. In un ambiente di accodamento messaggi, ogni programma che fa parte di una suite di applicazioni esegue una funzione ben definita e autonoma in risposta a una specifica richiesta. Per comunicare con un altro programma, un programma deve inserire un messaggio su una coda predefinita. L'altro programma richiama il messaggio dalla coda ed elabora le richieste e le informazioni contenute nel messaggio. Quindi l'accodamento dei messaggi è uno stile di comunicazione tra programmi.

L'accodamento è il meccanismo con cui i messaggi vengono conservati fino a quando un'applicazione non è pronta per elaborarli. L'accodamento consente di:

- Comunicare tra i programmi (che potrebbero essere in esecuzione in ambienti differenti) senza dover scrivere il codice di comunicazione.
- Selezionare l'ordine in cui un programma elabora i messaggi.
- Bilanciare i caricamenti su un sistema facendo in modo che più di un programma servano una coda quando il numero di messaggi supera una soglia.
- Aumentare la disponibilità delle applicazioni disponendo un sistema alternativo per servire le code se il sistema primario non è disponibile.

Cos' è una coda messaggi?

Una coda messaggi, nota semplicemente come coda, è una destinazione denominata a cui possono essere inviati i messaggi. I messaggi si accumulano sulle code fino a quando non vengono richiamate dai programmi che gestiscono tali code.

Le code si trovano in, e sono gestite da, un gestore code (consultare “Terminologia di accodamento messaggi” a pagina 9). La natura fisica di una coda dipende dal sistema operativo su cui è in esecuzione il gestore code. Una coda può essere un'area di buffer volatile nella memoria di un computer o un dataset su una periferica di archiviazione permanente (come un disco). La gestione fisica delle code è responsabilità del gestore code e non è resa evidente ai programmi di applicazione partecipanti.

I programmi accedono alle code solo tramite i servizi esterni del gestore code. Possono aprire una coda, inserire messaggi, ricevere messaggi da essa e chiudere la coda. Possono anche impostare e analizzare gli attributi delle code.

Diversi stili di accodamento dei messaggi

Point-to-point

Un messaggio viene inserito nella coda e un'applicazione riceve quel messaggio.

Nella messaggistica point - to - point, un'applicazione di invio deve conoscere le informazioni sull'applicazione di ricezione prima di poter inviare un messaggio a tale applicazione. Ad esempio, l'applicazione mittente potrebbe dover conoscere il nome della coda a cui inviare le informazioni e potrebbe anche specificare un nome gestore code.

Pubblicazione/Sottoscrizione

Una copia di ogni messaggio pubblicato da un'applicazione di pubblicazione viene consegnata a ogni applicazione interessata. Potrebbero essere presenti molte, una o nessuna applicazione interessata. Nella pubblicazione / sottoscrizione un'applicazione interessata è nota come sottoscrittore e i messaggi vengono accodati su una coda identificata da una sottoscrizione.

La messaggistica di pubblicazione / sottoscrizione consente di disaccoppiare il fornitore di informazioni dai consumatori di tali informazioni. L'applicazione mittente e l'applicazione ricevente non hanno bisogno di sapere molto l'una sull'altra per le informazioni da inviare e ricevere. Per ulteriori informazioni, consultare “Pubblicazione/sottoscrizione della messaggistica” a pagina 63.

Vantaggi dell'accodamento dei messaggi allo sviluppatore e allo sviluppatore dell'applicazione

IBM MQ consente ai programmi di applicazione di utilizzare l' *accodamento dei messaggi* per partecipare all'elaborazione basata sui messaggi. I programmi applicativi possono comunicare tra diverse piattaforme utilizzando i prodotti software di accodamento dei messaggi appropriati. Ad esempio, le applicazioni z/OS possono comunicare tramite IBM MQ for z/OS. Le applicazioni sono protette dalla meccanica delle comunicazioni sottostanti. Alcuni degli altri vantaggi dell'accodamento dei messaggi sono:

- È possibile progettare applicazioni utilizzando piccoli programmi che è possibile condividere tra molte applicazioni.

- È possibile creare rapidamente nuove applicazioni riutilizzando questi blocchi di creazione.
- Le applicazioni scritte per utilizzare le tecniche di accodamento dei messaggi non sono influenzate dalle modifiche apportate al funzionamento dei gestori code.
- Non è necessario utilizzare alcun protocollo di comunicazione. Il gestore code gestisce tutti gli aspetti della comunicazione.
- Non è necessario che i programmi che ricevono messaggi siano in esecuzione nel momento in cui vengono loro inviati. I messaggi vengono conservati nelle code.

I progettisti possono ridurre il costo delle applicazioni perché lo sviluppo è più rapido, sono necessari meno sviluppatori e le richieste di capacità di programmazione sono inferiori rispetto a quelle per le applicazioni che non utilizzano l'accodamento dei messaggi.

IBM MQ implementa un'API (application programming interface) comune nota come *interfaccia coda messaggi* (o MQI) ovunque vengano eseguite le applicazioni. In questo modo, è più semplice trasferire i programmi applicativi da una piattaforma all'altra.

Per i dettagli su MQI, vedere Panoramica di Message Queue Interface.

Caratteristiche principali e vantaggi dell'accodamento dei messaggi

Queste informazioni evidenziano alcune funzioni e vantaggi dell'accodamento dei messaggi. Descrive funzioni quali la sicurezza e l'integrità dei dati dell'accodamento dei messaggi.

Le funzioni principali delle applicazioni che utilizzano le tecniche di accodamento dei messaggi sono:

Nota: Quando si considerano client e server IBM MQ , non è necessario modificare un'applicazione server per supportare ulteriori IBM MQ MQI clients su nuove piattaforme. Allo stesso modo, IBM MQ MQI client può, senza modifiche, funzionare con ulteriori tipi di server.

Nessuna connessione diretta tra programmi

L'accodamento dei messaggi è una tecnica di comunicazione indiretta da programma a programma. Può essere utilizzato all'interno di qualsiasi applicazione in cui i programmi comunicano tra loro. La comunicazione viene eseguita da un programma che immette i messaggi su una coda (di proprietà di un gestore code) e da un altro programma che riceve i messaggi dalla coda.

I programmi possono ricevere messaggi che sono stati inseriti in una coda da altri programmi. Gli altri programmi possono essere connessi allo stesso gestore code del programma di ricezione o a un altro gestore code. Questo altro gestore code potrebbe trovarsi su un altro sistema, su un altro sistema di computer o anche all'interno di un'azienda o di un'azienda diversa.

Non esistono connessioni fisiche tra i programmi che comunicano utilizzando le code messaggi. Un programma invia messaggi a una coda di proprietà di un gestore code e un altro programma richiama i messaggi dalla coda (consultare Figura 1 a pagina 8).

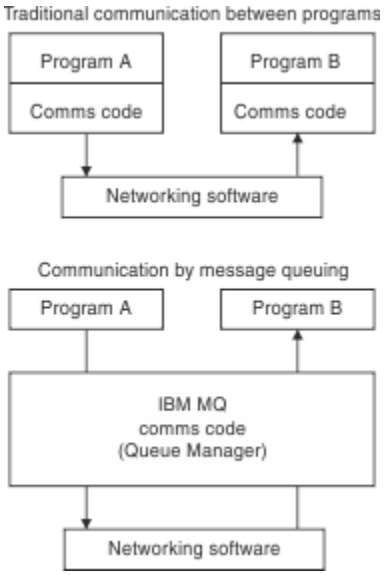


Figura 1. Accodamento messaggi rispetto alla comunicazione tradizionale

Come per la posta elettronica, i singoli messaggi che fanno parte di una transazione viaggiano attraverso una rete su un store - and - forwardbase. Se un collegamento tra i nodi ha esito negativo, il messaggio viene conservato fino a quando il collegamento non viene ripristinato oppure l'operatore o il programma reindirizza il messaggio.

Il meccanismo mediante il quale un messaggio viene spostato dalla coda alla coda è nascosto dai programmi. Pertanto i programmi sono più semplici.

Comunicazione indipendente dal tempo

I programmi che richiedono ad altri di lavorare non devono attendere la risposta ad una richiesta. Possono eseguire altre operazioni ed elaborare la risposta quando arriva o in un secondo momento. Quando si scrive un'applicazione di messaggistica, non è necessario sapere (o preoccuparsi) quando un programma invia un messaggio o quando la destinazione è in grado di ricevere il messaggio. Il messaggio non viene perso; viene conservato dal gestore code finché la destinazione non è pronta per elaborarlo. Il messaggio rimane nella coda fino a quando non viene rimosso dal programma. Ciò significa che i programmi applicativi di invio e ricezione vengono disaccoppiati; il mittente può continuare l'elaborazione senza attendere che il destinatario confermi la ricezione del messaggio. L'applicazione di destinazione non deve essere in esecuzione quando il messaggio viene inviato. Può richiamare il messaggio dopo che è stato avviato.

Piccoli programmi

L'accodamento dei messaggi consente di utilizzare i vantaggi dell'utilizzo di programmi piccoli e autonomi. Invece di un singolo, grande programma che esegue tutte le parti di un lavoro in modo sequenziale, è possibile distribuire il lavoro su diversi programmi più piccoli e indipendenti. Il programma richiedente invia messaggi a ciascuno dei programmi separati, chiedendo loro di eseguire la loro funzione; quando ogni programma è completo, i risultati vengono inviati indietro come uno o più messaggi.

Elaborazione basata sui messaggi

Quando i messaggi arrivano su una coda, possono avviare automaticamente un'applicazione utilizzando il *trigger*. Se necessario, le applicazioni possono essere arrestate quando il messaggio (o i messaggi) sono stati elaborati.

Elaborazione basata sugli eventi

I programmi possono essere controllati in base allo stato delle code. Ad esempio, è possibile organizzare l'avvio di un programma non appena un messaggio arriva su una coda oppure specificare che il programma non viene avviato fino a quando, ad esempio, non vi sono 10 messaggi al di sopra di una determinata priorità sulla coda o 10 messaggi di qualsiasi priorità sulla coda.

Priorità messaggio

Un programma può assegnare una priorità a un messaggio quando inserisce il messaggio su una coda. Ciò determina la posizione nella coda in cui viene aggiunto il nuovo messaggio.

I programmi possono richiamare i messaggi da una coda nell'ordine in cui i messaggi si trovano nella coda oppure ottenendo un messaggio specifico. (Un programma potrebbe voler ottenere un messaggio specifico se sta cercando la risposta a una richiesta che ha inviato in precedenza.)

Sicurezza

Vengono fornite funzioni di protezione, inclusa l'autenticazione delle applicazioni quando utilizzano un gestore code, i controlli di autorizzazione quando utilizzano risorse come una coda sul gestore code e la codifica dei dati dei messaggi quando viaggiano sulla rete e quando risiedono sulle code. Per ulteriori informazioni sulla sicurezza, consultare Panoramica della sicurezza.

Integrità dei dati

L'integrità dei dati è fornita dalle unità di lavoro. La sincronizzazione dell'inizio e della fine delle unità di lavoro è completamente supportata come opzione su ogni MQGET o MQPUT, consentendo il commit o il rollback dei risultati dell'unità di lavoro. Il supporto del punto di sincronizzazione opera internamente o esternamente a IBM MQ in base al formato di coordinamento del punto di sincronizzazione selezionato per l'applicazione.

Supporto di ripristino

Per consentire il ripristino, vengono registrati tutti gli aggiornamenti IBM MQ persistenti. Se il ripristino è necessario, vengono ripristinati tutti i messaggi persistenti, viene eseguito il rollback di tutte le transazioni in corso e qualsiasi commit e backout del punto di sincronizzazione viene gestito nel modo normale del gestore del punto di sincronizzazione in controllo. Per ulteriori informazioni sui messaggi persistenti, consultare Persistenza messaggio.

Terminologia di accodamento messaggi

Queste informazioni forniscono informazioni su alcuni termini utilizzati nell'accodamento dei messaggi.

Tra di essi ricordiamo:

- Canali
- Cluster
- IBM MQ MQI client
- z/OS Accodamento all'interno del gruppo
- Messaggio
- Agent del canale messaggi
- descrittore messaggi
- Point-to-point
- Pubblicazione/sottoscrizione
- Coda
- Gestore code

- z/OS Gruppo di condivisione code
- z/OS coda condivisa
- abbonamento
- Argomento

Canali

I canali vengono utilizzati per spostare i messaggi da un gestore code a un altro e proteggono le applicazioni dai protocolli di comunicazione sottostanti. I gestori code potrebbero esistere sullo stesso sistema, su sistemi differenti sulla stessa piattaforma o su piattaforme differenti. I messaggi inviati possono avere origine da molti luoghi:

- Programmi di applicazione scritti dall'utente che trasferiscono i dati da un nodo all'altro.
- Applicazioni di gestione scritte dall'utente che utilizzano i comandi PCF o MQAI.
- Il IBM MQ Explorer.
- Gestori code che inviano messaggi di evento di strumentazione ad un altro gestore code.
- Gestori code che inviano comandi di gestione remoti ad un altro gestore code. Ad esempio, utilizzando i comandi MQSC o administrative REST API.

Per ulteriori informazioni sui canali, consultare “Definizioni canale” a pagina 32.

Cluster

Un *cluster* è una rete di gestori code associati logicamente in qualche modo.

In una rete IBM MQ che utilizza l'accodamento distribuito senza cluster, ogni gestore code è indipendente. Se un gestore code deve inviare messaggi a un altro, deve aver definito una coda di trasmissione e un canale per il gestore code remoto.

Esistono due diversi motivi per utilizzare i cluster: per ridurre la gestione del sistema e migliorare la disponibilità e il bilanciamento del carico di lavoro.

Non appena si stabilisce anche il cluster più piccolo, si beneficia di una gestione semplificata del sistema. I gestori code che fanno parte di un cluster hanno bisogno di un numero minore di definizioni e quindi il rischio di commettere un errore nelle definizioni è ridotto.

Per ulteriori informazioni sul cluster, consultare Cluster.

IBM MQ MQI client

I IBM MQ client *MQI* sono componenti installabili in modo indipendente di IBM MQ. Un client MQI consente di eseguire applicazioni IBM MQ con un protocollo di comunicazione, di interagire con uno o più server MQI (Message Queue Interface) su altre piattaforme e di connettersi ai relativi gestori code.

Per i dettagli completi su come installare e utilizzare i componenti IBM MQ MQI client , consultare i seguenti argomenti:

- AIX Installazione di un client di IBM MQ su AIX
- Linux Installazione di un client di IBM MQ su Linux®
- Windows Installazione di un client di IBM MQ su Windows
- IBM i Installazione di un client di IBM MQ su IBM i

e Configurazione delle connessioni tra il server e client.

Accodamento all'interno del gruppo

z/OS

I gestori code in un gruppo di condivisione code possono comunicare utilizzando canali normali oppure è possibile utilizzare una tecnica denominata *IGQ (intra - group queuing*), che consente di eseguire un trasferimento messaggi rapido senza definire i canali. Ciò si applica solo a IBM MQ for z/OS.

Per ulteriori informazioni sull'accodamento all'interno del gruppo, consultare “Accodamento all'interno del gruppo” a pagina 217.

Messaggio

Nell'accodamento dei messaggi, un messaggio è una raccolta di dati inviati da un programma e destinati a un altro programma. Vedere IBM MQ messaggi.

Per informazioni sui tipi di messaggio, vedere Tipi di messaggio.

Agent del canale messaggi

Un MCA (message channel agent) è un'estremità di un canale. Una coppia di agent del canale dei messaggi, uno che invia e uno che riceve, crea un canale e sposta i messaggi da un gestore code all'altro.

Per informazioni su come vengono utilizzati gli agent del canale dei messaggi, consultare Introduzione alla gestione delle code distribuite.

Descrittore messaggio

Un messaggio IBM MQ è costituito da informazioni di controllo e dati dell'applicazione.

Le informazioni di controllo sono definite in una struttura del descrittore del messaggio (MQMD) e contengono elementi quali:

- Il tipo di messaggio
- Un identificatore per il messaggio
- La priorità per la consegna del messaggio

La struttura e il contenuto dei dati dell'applicazione sono determinati dai programmi partecipanti, non da IBM MQ.

Per ulteriori informazioni, vedere MQMD.

Messaggistica point-to-point

Nella messaggistica point-to-point, ogni messaggio passa da un'applicazione di produzione a un'applicazione di consumo. I messaggi vengono trasferiti tramite l'applicazione di produzione inserendo i messaggi in una coda e l'applicazione di consumo li richiama da tale coda.

Pubblicazione/sottoscrizione della messaggistica

Nella messaggistica di pubblicazione / sottoscrizione, una copia di ciascun messaggio pubblicato da un'applicazione di pubblicazione viene consegnata a ciascuna applicazione interessata. Potrebbero essere presenti molte, una o nessuna applicazione interessata. Nella pubblicazione / sottoscrizione un'applicazione interessata è nota come sottoscrittore e i messaggi vengono accodati su una coda identificata da una sottoscrizione.

Per ulteriori informazioni, consultare “Pubblicazione/sottoscrizione della messaggistica” a pagina 63.

Coda

Una destinazione denominata a cui possono essere inviati i messaggi. I messaggi si accumulano sulle code fino a quando non vengono richiamate dai programmi che gestiscono tali code.

Per ulteriori informazioni, consultare “Code” a pagina 20.

Gestore code

Un *gestore code* è un programma di sistema che fornisce servizi di accodamento alle applicazioni.

Fornisce un'interfaccia di programmazione dell'applicazione in modo che i programmi possano inserire messaggi e ricevere messaggi dalle code. Un gestore code fornisce funzioni supplementari in modo che gli amministratori possano creare nuove code, modificare le proprietà delle code esistenti e controllare il funzionamento del gestore code.

Perché i servizi di accodamento messaggi IBM MQ siano disponibili su un sistema, è necessario che sia in esecuzione un gestore code. È possibile avere più di un gestore code in esecuzione su un singolo sistema (ad esempio, per separare un sistema di test da un sistema *attivo*). Per un'applicazione, ogni gestore code è identificato da un *handle di connessione* (*Hconn*).

Molte applicazioni differenti possono utilizzare i servizi del gestore code allo stesso tempo e queste applicazioni possono essere completamente non correlate. Affinché un programma utilizzi i servizi di un gestore code, è necessario stabilire una connessione a tale gestore code.

Per consentire alle applicazioni di inviare messaggi alle applicazioni connesse ad altri gestori code, è necessario che i gestori code siano in grado di comunicare tra loro. IBM MQ implementa un protocollo *store - and - forward* per assicurare la consegna sicura dei messaggi tra tali applicazioni.

Per ulteriori informazioni, consultare “Gestori code” a pagina 29.

Gruppo di condivisione code

z/OS

I gestori code che possono accedere alla stessa serie di code condivise formano un gruppo denominato *gruppo di condivisione code* (QSG). Comunicano tra loro con una CF (coupling facility) che memorizza le code condivise. Ciò si applica solo a IBM MQ for z/OS.

Per ulteriori informazioni, consultare “Code condivise e gruppi di condivisione code” a pagina 171.

Coda condivisa

z/OS

Una *coda condivisa* è un tipo di coda locale con messaggi a cui possono accedere uno o più gestori code che si trovano in un sysplex. Non è la stessa coda condivisa da più di un'applicazione, utilizzando lo stesso gestore code. Ciò si applica solo a IBM MQ for z/OS.

Sottoscrizione

Un'applicazione di pubblicazione / sottoscrizione può registrare un interesse nei messaggi su argomenti specifici. Quando un'applicazione esegue questa operazione, è nota come sottoscrittore e il termine sottoscrizione definisce in che modo i messaggi corrispondenti vengono accodati per l'elaborazione.

Una sottoscrizione contiene informazioni sull'identità del sottoscrittore e sull'identità della coda di destinazione in cui devono essere collocate le pubblicazioni. Contiene inoltre informazioni sul modo in cui una pubblicazione deve essere inserita nella coda di destinazione.

Per ulteriori informazioni, consultare “Sottoscrittori e sottoscrizioni” a pagina 66.

Argomento

Un argomento è una stringa di caratteri che descrive l'oggetto dell'informazione pubblicata in un messaggio di pubblicazione/sottoscrizione.

Gli argomenti sono fondamentali per una consegna riuscita dei messaggi in un sistema di pubblicazione/sottoscrizione. Invece di includere un indirizzo di destinazione specifico in ciascun messaggio, un

publisher assegna un argomento a ciascun messaggio. Il gestore code fa corrispondere l'argomento a un elenco di sottoscrittori che hanno sottoscritto tale argomento e consegna il messaggio a ciascuno di questi sottoscrittori.

Per ulteriori informazioni, consultare “Argomenti” a pagina 69.

Messaggi e code

I messaggi e le code sono i componenti di base di un sistema di accodamento messaggi.

Cos' è un messaggio?

Un *messaggio* è una stringa di byte significativa per le applicazioni che lo utilizzano. I messaggi vengono utilizzati per trasferire le informazioni da un programma applicativo ad un altro (o tra diverse parti della stessa applicazione). Le applicazioni possono essere in esecuzione sulla stessa piattaforma o su piattaforme differenti.

Un messaggio IBM MQ è composto da:

- *I dati dell'applicazione.*Il contenuto e la struttura dei dati dell'applicazione sono definiti dai programmi applicativi che li utilizzano.
- *Un descrittore di messaggi.*Il descrittore del messaggio identifica il messaggio e contiene ulteriori informazioni di controllo, ad esempio il tipo di messaggio e la priorità assegnata al messaggio dall'applicazione mittente.

  Il formato del descrittore del messaggio è definito da IBM MQ. Per una descrizione completa del descrittore del messaggio, vedere MQMD - Descrittore del messaggio.
- *Proprietà del messaggio.* Metadati relativi al messaggio. Il contenuto delle proprietà del messaggio viene definito dai programmi applicativi che le utilizzano. Per ulteriori informazioni, consultare Proprietà del messaggio.

Lunghezze dei messaggi

La lunghezza massima predefinita del messaggio è 4 MB, anche se è possibile aumentarla a una lunghezza massima di 100 MB (dove 1 MB equivale a 1 048 576 byte). In pratica, la lunghezza del messaggio potrebbe essere limitata da:

- La lunghezza massima del messaggio definita per la coda di ricezione
- La lunghezza massima del messaggio definita per il gestore code
- La lunghezza massima del messaggio definita dalla coda
- La lunghezza massima del messaggio definita dall'applicazione mittente o ricevente
- La quantità di memoria disponibile per il messaggio

Potrebbero essere necessari diversi messaggi per inviare tutte le informazioni richieste da un'applicazione.

In che modo le applicazioni inviano e ricevono messaggi?

I programmi applicativi inviano e ricevono messaggi utilizzando **chiamate MQI**.

Ad esempio, per inserire un messaggio in una coda, un'applicazione:

1. Apre la coda richiesta emettendo una chiamata MQI `MQOPEN`
2. Emette una chiamata MQI `MQPUT` per inserire il messaggio nella coda

Un'altra applicazione può richiamare il messaggio dalla stessa coda emettendo una chiamata MQI `MQGET`

Per ulteriori informazioni sulle chiamate MQI, vedi Chiamate MQI.

Cos' è una coda?

Una *coda* è una struttura di dati utilizzata per memorizzare i messaggi.

Ogni coda è di un *gestore code*. Il gestore code è responsabile della gestione delle code di sua proprietà e della memorizzazione di tutti i messaggi ricevuti nelle code appropriate. I messaggi potrebbero essere inseriti nella coda dai programmi di applicazione o da un gestore code come parte della normale operazione.

Code predefinite e code dinamiche

Le code possono essere caratterizzate dal modo in cui vengono create:

- Le **code predefinite** vengono create da un amministratore utilizzando i comandi MQSC o PCF appropriati.Le code predefinite sono permanenti; esistono indipendentemente dalle applicazioni che le utilizzano e sopravvivono ai riavvii di IBM MQ .
- Le **code dinamiche** vengono create quando un'applicazione emette una richiesta MQOPEN specificando il nome di una *coda modello*. La coda creata è basata su una *definizione di coda modello*, denominata coda modello.È possibile creare una coda modello utilizzando il comando MQSC DEFINE QMODEL. Gli attributi di una coda modello (ad esempio, il numero massimo di messaggi che possono essere memorizzati su di essa) vengono ereditati da qualsiasi coda dinamica creata da essa.

  Le code modello hanno un attributo che specifica se la coda dinamica deve essere permanente o temporanea. Le code permanenti sopravvivono al riavvio dell'applicazione e del gestore code; le code temporanee vengono perse al riavvio.

Richiamo dei messaggi dalle code

Le applicazioni opportunamente autorizzate possono richiamare i messaggi da una coda in base ai seguenti algoritmi di recupero:

- FIFO (First - in - first - out).
- Priorità del messaggio, come definito nel descrittore del messaggio. I messaggi con la stessa priorità vengono richiamati su base FIFO.
- Una richiesta di programma per un messaggio specifico.

La richiesta MQGET dall'applicazione determina il metodo utilizzato.

Oggetti IBM MQ

I gestori code definiscono le proprietà degli oggetti IBM MQ . I valori di queste proprietà influenzano il modo in cui IBM MQ elabora questi oggetti. Gli oggetti vengono creati e gestiti utilizzando le interfacce e i comandi IBM MQ . Dalle proprie applicazioni, si utilizza MQI (Message Queue Interface) per controllare gli oggetti. Gli oggetti sono identificati da un MQOD (IBM MQ *object descriptor*) quando vengono indirizzati da un programma.

Gestione oggetti

La gestione degli oggetti include le seguenti attività:

- Avvio e arresto dei gestori code.
- Creazione di oggetti, in particolare code, per applicazioni.
- Visualizzazione o modifica degli attributi degli oggetti.
- Eliminazione oggetti.
- Utilizzo dei canali per creare percorsi di comunicazione per i gestori code su altri sistemi (remoti).
- Creazione di *cluster* di gestori code per semplificare il processo di gestione generale e bilanciare il carico di lavoro.

Ad eccezione delle code dinamiche, gli oggetti devono essere definiti nel gestore code prima di poterli utilizzare.

Quando si utilizza un comando IBM MQ per eseguire un'operazione di gestione degli oggetti, il gestore code verifica di disporre del livello di autorizzazione richiesto per eseguire l'operazione. Allo stesso modo, quando un'applicazione utilizza la chiamata MQOPEN per aprire un oggetto, il gestore code controlla che l'applicazione disponga del livello di autorizzazione richiesto prima di consentire l'accesso a tale oggetto. Le verifiche vengono effettuate sul nome dell'oggetto da aprire.

È possibile definire e gestire gli oggetti utilizzando i seguenti metodi:

- I comandi PCF descritti in Programmable command formats reference e Automating administration tasks
- I comandi MQSC descritti in Comandi MQSC
- z/OS Le operazioni e i pannelli di controllo IBM MQ for z/OS , descritti in Amministrazione IBM MQ for z/OS
- Linux Windows IBM MQ Explorer (solo Windows e Linux per sistemi Intel). Per ulteriori informazioni, consultare Introduzione a MQ Explorer.

È anche possibile gestire gli oggetti utilizzando i seguenti metodi:

- Comandi di controllo, immessi da una tastiera. Consultare Amministrazione IBM MQ for Multiplatforms utilizzando i comandi di controllo.
- MQAI (IBM MQ Administration Interface) richiama un programma. Vedere IBM MQ Administration Interface (MQAI).

ALW Per le sequenze di comandi IBM MQ su AIX, Linux, and Windows, è possibile utilizzare la funzionalità MQSC per eseguire una serie di comandi contenuti in un file. Per ulteriori informazioni, consultare Amministrazione IBM MQ utilizzando i comandi MQSC.

IBM i Per le sequenze di comandi IBM MQ per IBM i che si utilizzano regolarmente è possibile scrivere programmi CL. Per ulteriori informazioni, consultare Gestione IBM MQ for IBM i utilizzando i comandi CL.

z/OS Per sequenze di comandi IBM MQ for z/OS che si utilizzano regolarmente, è possibile scrivere programmi di gestione che creano messaggi contenenti comandi e che inseriscono tali messaggi nella coda di input dei comandi del sistema. Il gestore code elabora i messaggi su questa coda nello stesso modo in cui elabora i comandi immessi dalla riga comandi o dalle operazioni e dai pannelli di controllo. Questa tecnica è descritta in Scrittura di programmi per la gestione di IBM MQe illustrata nell'applicazione di esempio di Mail Manager fornita con IBM MQ for z/OS. Per una descrizione di questo esempio, vedere Programmi di esempio per IBM MQ for z/OS .

Attributi oggetto

Le proprietà di un oggetto sono definite dai suoi attributi. Alcuni possono essere specificati, altri possono essere solo visualizzati.

Ad esempio, la lunghezza massima del messaggio che una coda può contenere è definita dal relativo attributo **MaxMsgLength** ; è possibile specificare questo attributo quando viene creata una coda. L'attributo **DefinitionType** specifica come è stata creata la coda; è possibile visualizzare solo questo attributo.

In IBM MQ, esistono due modi per fare riferimento ad un attributo:

- Utilizzando il nome PCF, ad esempio, **MaxMsgLength**.
- Utilizzando il nome del comando MQSC, ad esempio MAXMSGL.

Gruppi di condivisione code

z/OS

I gestori code che possono accedere alla stessa serie di code condivise formano un gruppo denominato *gruppo di condivisione code* (QSG) e comunicano tra loro utilizzando una CF (coupling facility) che memorizza le code condivise. Si noti che un QSG non è un oggetto.

Una coda condivisa è un tipo di coda locale con messaggi a cui possono accedere uno o più gestori code che si trovano in un gruppo di condivisione code. Non è la stessa coda condivisa da più di un'applicazione, utilizzando lo stesso gestore code.

I gruppi di condivisione code hanno un nome composto da un massimo di quattro caratteri. Il nome deve essere univoco nella rete e diverso da qualsiasi altro nome di gestore code.

Importante: Le code condivise e i gruppi di condivisione code sono supportati solo su IBM MQ for z/OS.

Per ulteriori informazioni, consultare “Code condivise e gruppi di condivisione code” a pagina 171.

Oggetti predefiniti del sistema

Gli *oggetti predefiniti di sistema* sono una serie di definizioni di oggetti che vengono create automaticamente ogni volta che viene creato un gestore code. È possibile copiare e modificare una qualsiasi di queste definizioni di oggetto da utilizzare nelle applicazioni durante l'installazione.

I nomi oggetto predefiniti hanno la radice SYSTEM; ad esempio, la coda locale predefinita è SYSTEM.DEFAULT.LOCAL.QUEUEe il canale ricevente predefinito è SYSTEM.DEF.RECEIVER. Non è possibile ridenominare questi oggetti; sono richiesti gli oggetti predefiniti di questi nomi.

Quando si definisce un oggetto, tutti gli attributi non specificati esplicitamente vengono copiati dall'oggetto predefinito appropriato. Ad esempio, se si definisce una coda locale, gli attributi non specificati vengono presi dalla coda predefinita SYSTEM.DEFAULT.LOCAL.QUEUE.

Per ulteriori informazioni, consultare Oggetti di sistema e predefiniti .

Tipi di oggetto

Molte delle attività di gestione riguardano la manipolazione di vari tipi di IBM MQ *oggetti*.

Per informazioni sulla denominazione di oggetti IBM MQ , consultare “Denominazione degli oggetti IBM MQ” a pagina 37.

Per informazioni sugli oggetti predefiniti creati su un gestore code, consultare “Oggetti predefiniti del sistema” a pagina 16.

Per informazioni sui diversi tipi di oggetti IBM MQ , consultare quanto segue:

Oggetti delle informazioni di autenticazione

Un oggetto delle informazioni di autenticazione fornisce le definizioni richieste per eseguire il controllo della revoca del certificato.

L'oggetto delle informazioni di autenticazione del gestore code fa parte del supporto IBM MQ per TLS (Transport Layer Security). Fornisce le definizioni necessarie per controllare i certificati revocati. Le autorità di certificazione revocano i certificati che non sono più attendibili.

È possibile utilizzare il comando MQSC **DEFINE AUTHINFO** per definire un oggetto delle informazioni di autenticazione. Per ulteriori informazioni sugli attributi degli oggetti delle informazioni di autenticazione, consultare **DEFINE AUTHINFO**.

È possibile utilizzare i comandi di controllo IBM MQ seguenti con un oggetto delle informazioni di autenticazione:

- **setmqaut** (concessione o revoca dell'autorizzazione)
- **dspmqaut** (visualizza autorizzazione oggetto)

- **dmpmqaut** (autorizzazioni dump)
- **rcrmqobj** (ricrea oggetto)
- **rcdmqimg** (registrazione immagine supporto)
- **dspmqfls** (visualizza nomi file)

Per una panoramica di TLS e sull'utilizzo degli oggetti delle informazioni di autenticazione, consultare Transport Layer Security (TLS) concepts e TLS security protocols in IBM MQ.

Canali

I canali sono oggetti che forniscono un percorso di comunicazione da un gestore code all'altro.

Per ulteriori informazioni, consultare “Canali” a pagina 30.

Oggetti informazioni di comunicazione

Il multicast IBM MQ offre un'affidabile messaggistica multicast a bassa latenza ed elevato fanout. È necessario un oggetto Informazioni di comunicazione (COMMINFO) per utilizzare la trasmissione multicast.

Per ulteriori informazioni, consultare “IBM MQ Multicast” a pagina 109.

Un oggetto COMMINFO è un oggetto IBM MQ che contiene gli attributi associati alla trasmissione multicast. Per ulteriori informazioni su questi attributi, consultare DEFINE COMMINFO. Per ulteriori informazioni sulla creazione di un oggetto COMMINFO, consultare Introduzione al multicast.

Listener

I *listener* sono processi che accettano le richieste di rete da altri gestori code o applicazioni client e avviano i canali associati.

I *processi listener* possono essere avviati utilizzando il comando di controllo **runmqlsr** .

I *listener* sono IBM MQ oggetti che consentono di gestire l'avvio e l'arresto dei processi listener dall'ambito di un gestore code. Definendo gli attributi di un oggetto listener si effettua quanto segue:

- Configurare il processo listener.
- Specificare se il processo listener viene avviato e arrestato automaticamente quando il gestore code viene avviato e arrestato.

Importante: z/OS Gli oggetti listener non sono supportati su IBM MQ for z/OS. Per ulteriori informazioni su come IBM MQ for z/OS implementa l'ascolto, utilizzando l'iniziatore di canali, consultare “L'iniziatore di canali su z/OS” a pagina 167.

Elenchi nomi

Un *elenco nomi* è un oggetto IBM MQ che contiene un elenco di nomi cluster, nomi coda o nomi oggetto delle informazioni di autenticazione. In un cluster, può essere utilizzato per identificare un elenco di cluster per cui il gestore code detiene i repository.

Un elenco nomi è un oggetto IBM MQ che contiene un elenco di altri oggetti IBM MQ . Di solito, gli elenchi nomi sono utilizzati dalle applicazioni, quali i monitor del trigger, dove sono utilizzati per identificare un gruppo delle code. Il vantaggio di utilizzare un elenco nomi è che viene gestito indipendentemente dalle applicazioni; può essere aggiornato senza arrestare alcuna delle applicazioni che lo utilizzano. Inoltre, se si verifica un malfunzionamento di un'applicazione, l'elenco nomi non viene influenzato e le altre applicazioni possono continuare a utilizzarlo.

Gli elenchi di nomi vengono utilizzati anche con i cluster del gestore code per gestire un elenco di cluster a cui fanno riferimento più oggetti IBM MQ .

È possibile definire e modificare gli elenchi nomi utilizzando i comandi MQSC DEFINE NAMELIST e ALTER NAMELIST .

Nota: z/OS In alternativa, su z/OS, è possibile utilizzare le operazioni IBM MQ for z/OS e i pannelli di controllo

I programmi possono utilizzare MQI per scoprire quali code sono incluse in questi elenchi nomi. L'organizzazione degli elenchi nomi è responsabilità del progettista dell'applicazione e dell'amministratore di sistema.

Per un elenco degli attributi dell'elenco nomi disponibili da utilizzare, consultare Attributi per gli elenchi nomi,

Definizioni dei processi

Gli oggetti di definizione del processo consentono alle applicazioni di essere avviate senza la necessità di un intervento dell'operatore definendo gli attributi dell'applicazione per l'utilizzo da parte del gestore code.

L'oggetto di definizione del processo definisce un'applicazione che inizia in risposta a un evento trigger su un gestore code IBM MQ . Gli attributi di definizione del processo includono l'ID applicazione, il tipo di applicazione e i dati specifici dell'applicazione. Per ulteriori informazioni, consultare *Code di iniziazione* in “Code utilizzate per scopi specifici da parte di IBM MQ” a pagina 27.

Per consentire l'avvio di un'applicazione senza la necessità di intervento dell'operatore, come descritto in Avvio di applicazioni IBM MQ utilizzando i trigger, gli attributi dell'applicazione devono essere noti al gestore code. Questi attributi sono definiti in un *oggetto definizione processo*.

L'attributo **ProcessName** è fisso quando l'oggetto viene creato. Tuttavia, è possibile modificare altri attributi utilizzando i comandi IBM MQ .

Nota: z/OS In alternativa, su z/OS, è possibile utilizzare le operazioni IBM MQ for z/OS e i pannelli di controllo.

È possibile analizzare i valori di tutti gli attributi utilizzando MQINQ - Interroga attributi oggetto.

Per un elenco degli attributi di definizione del processo disponibili per l'uso, consultare Attributi per le definizioni del processo.

Code

Una IBM MQ *coda* è un oggetto denominato su cui le applicazioni possono inserire messaggi e da cui le applicazioni possono ricevere messaggi.

Per ulteriori informazioni, consultare “Code” a pagina 20.

Gestori code

I gestori code IBM MQ forniscono i servizi di accodamento alle applicazioni e gestiscono le code che vi appartengono.

Per ulteriori informazioni, consultare “Gestori code” a pagina 29.

Servizi

Gli oggetti *Servizio* sono un metodo per definire i programmi da eseguire quando un gestore code viene avviato o arrestato.

I programmi possono essere uno dei seguenti tipi:

Server
Un *server* è un oggetto di servizio che ha il parametro SERVTYPE specificato come SERVER. Un oggetto servizio server è la definizione di un programma che verrà eseguito quando viene avviato un

gestore code specificato. È possibile eseguire contemporaneamente una sola istanza di un processo server. Durante l'esecuzione, lo stato di un processo server può essere monitorato utilizzando il comando MQSC, DISPLAY SVSTATUS. In genere gli oggetti di servizio del server sono definizioni di programmi come gestori di messaggi non recapitabili o controlli trigger, tuttavia i programmi che possono essere eseguiti non sono limitati a quelli forniti con IBM MQ. Inoltre, è possibile definire un oggetto servizio server per includere un comando che verrà eseguito quando il gestore code specificato viene arrestato per terminare il programma.

Comandi
Un *comando* è un oggetto servizio che ha il parametro SERVTYPE specificato come COMMAND. Un oggetto servizio comandi è la definizione di un programma che verrà eseguito quando viene avviato o arrestato un gestore code specificato. Più istanze di un processo di comando possono essere eseguite contemporaneamente. Gli oggetti servizio comando differiscono dagli oggetti servizio server in quanto, una volta eseguito il programma, il gestore code non monitorerà il programma. In genere, gli oggetti servizio comandi sono definizioni di programmi di breve durata che eseguono un'attività specifica, ad esempio l'avvio di una o più altre attività.

Importante: z/OS Gli oggetti di servizio non sono supportati su IBM MQ for z/OS.

Per ulteriori informazioni, vedi Utilizzo dei servizi.

Classi di memoria

z/OS

Una classe di memoria associa una o più code ad una serie di pagine.

Ciò significa che i messaggi per tale coda vengono memorizzati (in base al buffer) su tale serie di pagine.

Le classi di memoria sono supportate solo su IBM MQ for z/OS.

Per ulteriori informazioni sulle classi di archiviazione, consultare Pianificazione dell'ambiente IBM MQ su z/OS.

Oggetti argomento

Un *oggetto argomento* è un oggetto IBM MQ che permette di assegnare specifici attributi non predefiniti agli argomenti.

Un *argomento* è definito da un'applicazione che pubblica o sottoscrive una particolare *stringa argomento*. Una stringa di argomenti può specificare una gerarchia di argomenti separandoli con una barra (/). Ciò può essere visualizzato da un *albero degli argomenti*. Ad esempio, se un'applicazione esegue la pubblicazione nelle stringhe argomento `/Sport/American Football` e `/Sport/Soccer`, viene creata una struttura ad albero degli argomenti che ha un nodo parent `Sport` con due child, `American Football`e `Soccer`.

Gli argomenti ereditano i relativi attributi dal primo nodo di gestione parent trovato nella relativa struttura ad albero degli argomenti. Se non esistono nodi di argomenti di gestione in una particolare struttura ad albero degli argomenti, tutti gli argomenti ereditano i relativi attributi dall'oggetto argomento di base, SYSTEM.BASE.TOPIC.

È possibile creare un oggetto argomento in qualsiasi nodo in una struttura ad albero degli argomenti specificando la stringa argomento del nodo nell'attributo TOPICSTR dell'oggetto argomento. È anche possibile definire altri attributi per il nodo argomento di amministrazione. Per ulteriori informazioni su questi attributi, consultare I comandi MQSCo Gestione automatica mediante comandi PCF. Ciascun oggetto argomento, per impostazione predefinita, eredita gli attributi dal nodo argomento di gestione principale più vicino.

Gli oggetti argomento possono essere utilizzati anche per nascondere l'intera struttura ad albero degli argomenti agli sviluppatori di applicazioni. Se un oggetto argomento denominato `FOOTBALL.US` viene creato per l'argomento `/Sport/American Football`, un'applicazione può pubblicare o sottoscrivere l'oggetto denominato `FOOTBALL.US` invece della stringa `/Sport/American Football` con lo stesso risultato.

Se si immette un carattere #, +, / o * all'interno di una stringa di argomenti su un oggetto argomento, il carattere viene considerato come un carattere normale all'interno della stringa e viene considerato parte della stringa di argomenti associata a un oggetto argomento.

Per ulteriori informazioni sugli oggetti argomento, consultare “Pubblicazione/sottoscrizione della messaggistica” a pagina 63.

Concetti correlati

“Introduzione all'accodamento dei messaggi” a pagina 5
I prodotti IBM MQ consentono ai programmi di comunicare tra loro attraverso una rete di componenti diversi (processori, sistemi operativi, sottosistemi e protocolli di comunicazione) utilizzando un'API (application programming interface) coerente.

Riferimenti correlati

Comandi MQSC

Code

Introduzione alle code e agli attributi della coda IBM MQ .

I messaggi vengono memorizzati su una coda, in modo che se l'applicazione di inserimento è in attesa di una risposta al suo messaggio, è libera di eseguire altre operazioni mentre è in attesa di tale risposta. Le applicazioni accedono ad una coda utilizzando MQI (Message Queue Interface), descritto in Panoramica di Message Queue Interface.

Prima che un messaggio possa essere inserito in una coda, la coda deve essere già stata creata. Una coda è di proprietà di un gestore code e tale gestore code può possedere molte code. Tuttavia, ogni coda deve avere un nome univoco all'interno del gestore code.

Una coda viene gestita tramite un gestore code. Nella maggior parte dei casi, ogni coda è gestita fisicamente dal relativo gestore code, ma ciò non è evidente per un programma applicativo. IBM MQ for z/OS le code condivise possono essere gestite da qualsiasi gestore code nel gruppo di condivisione code.

Per creare una coda è possibile utilizzare i comandi IBM MQ (MQSC), i comandi PCF o le interfacce specifiche della piattaforma. Ad esempio, i pannelli di controllo e le operazioni IBM MQ for z/OS sono specifici della piattaforma.

È possibile creare code locali per lavori temporanei *dinamicamente* dall'applicazione. Ad esempio, è possibile creare code *reply - to* (che non sono necessarie una volta terminata un'applicazione). Per ulteriori informazioni, consultare “Code dinamiche e modello” a pagina 25.

Prima di utilizzare una coda, è necessario aprire la coda, specificando le operazioni da eseguire. Ad esempio, è possibile aprire una coda per:

- Solo messaggi di esplorazione (non richiamandoli)
- Richiamo dei messaggi (e condivisione dell'accesso con altri programmi o con accesso esclusivo)
- Inserimento di messaggi nella coda
- Richiesta di informazioni sugli attributi della coda
- Impostazione degli attributi della coda

Per un elenco completo delle opzioni che è possibile specificare quando si apre una coda, consultare MQOPEN-Open object.

Attributi delle code

Alcuni degli attributi di una coda vengono specificati quando la coda viene definita e non possono essere modificati successivamente (ad esempio, il tipo di coda). Gli altri attributi delle code possono essere raggruppati in quelli che possono essere modificati:

- Dal gestore code durante l'elaborazione della coda (ad esempio, la grandezza corrente di una coda)
- Solo per comandi (ad esempio, la descrizione testuale della coda)

- Dalle applicazioni, utilizzando la chiamata MQSET (ad esempio, se le operazioni di inserimento sono consentite sulla coda)

È possibile trovare i valori di tutti gli attributi utilizzando la chiamata MQINQ.

Gli attributi comuni a più di un tipo di coda sono:

QName
Il nome della coda.

QTYPE
Il tipo della coda.

QDesc
Descrizione testuale della coda.

InhibitGet
Indica se ai programmi è consentito richiamare i messaggi dalla coda. Tuttavia, non è possibile ottenere messaggi dalle code remote.

InhibitPut
Indica se ai programmi è consentito inserire messaggi nella coda.

DefPriority
Priorità predefinita per i messaggi inseriti nella coda.

DefPersistence
Persistenza predefinita per i messaggi inseriti nella coda

Ambito
Controlla se una voce per questa coda esiste anche in un servizio nomi.

z/OS L'attributo **Scope** non è supportato su z/OS .

Per una descrizione completa di questi attributi, vedere Attributi per le code.

Modi di definire le code

È possibile definire le code in IBM MQ utilizzando il comando MQSC DEFINE o il comando PCF Crea coda . I comandi specificano il tipo di coda e i suoi attributi. Ad esempio, un oggetto coda locale dispone di attributi che specificano cosa accade quando le applicazioni fanno riferimento a tale coda nelle chiamate MQI. Esempi di attributi sono:

- Se le applicazioni possono richiamare i messaggi dalla coda (GET abilitato)
- Indica se le applicazioni possono inserire messaggi nella coda (PUT abilitato)
- Se l'accesso alla coda è esclusivo per un'applicazione o condiviso tra le applicazioni
- Il numero massimo di messaggi che possono essere memorizzati nella coda contemporaneamente (profondità massima della coda)
- La lunghezza massima dei messaggi che possono essere inseriti nella coda

Esistono anche diverse interfacce specifiche della piattaforma che è possibile utilizzare per definire le code.

Concetti correlati

“Code cluster” a pagina 59
Una coda cluster è una coda ospitata da un gestore code cluster e resa disponibile ad altri gestori code del cluster.

“Code di messaggi non recapitabili” a pagina 50
La coda di messaggi non recapitabili (o coda di messaggi non recapitati) è la coda a cui vengono inviati i messaggi se non possono essere instradati alla destinazione corretta. Ogni gestore code generalmente ha una coda di messaggi non recapitabili.

Automatizzazione della gestione mediante comandi PCF
Gestione delle code in IBM MQ Console

Attività correlate

Amministrazione di IBM MQ utilizzando i comandi MQSC

Creazione e configurazione di gestori code e oggetti con MQ Explorer

IBM i Gestione di IBM MQ for IBM i utilizzando i comandi CL

z/OS Origini da cui è possibile emettere comandi MQSC e PCF su IBM MQ for z/OS

Riferimenti correlati

“Confronto tra code condivise e code cluster” a pagina 59

Queste informazioni sono progettate per confrontare le code condivise e le code cluster e decidere quali potrebbero essere più adatte al proprio sistema.

Informazioni correlate

“Cos' è una coda condivisa?” a pagina 171

Code locali

Le code di trasmissione, iniziazione, lettera non recapitabile, comando, valore predefinito, canale e eventi sono tipi di coda locale.

Una coda è nota a un programma come *locale* se è di proprietà del gestore code al quale è connesso il programma. È possibile ricevere messaggi dalle code locali e inserire messaggi nelle code locali.

L'oggetto di definizione della coda contiene le informazioni di definizione della coda, nonché messaggi fisici inseriti nella coda.

Ogni gestore code può avere alcune code locali che utilizza per scopi speciali:

Code di trasmissione

Quando un'applicazione invia un messaggio a una coda remota, il gestore code locale memorizza il messaggio in una speciale coda locale, denominata *coda di trasmissione*. Le applicazioni possono inserire i messaggi direttamente su una coda di trasmissione o indirettamente tramite una definizione di coda remota.

Quando un gestore code invia messaggi a un gestore code remoto, identifica la coda di trasmissione utilizzando la seguente sequenza:

1. La coda di trasmissione denominata nell'attributo XMITQ della definizione locale di una coda remota.
2. Una coda di trasmissione con lo stesso nome del gestore code remoto. Questo valore è il valore predefinito su XMITQ della definizione locale di una coda remota.
3. La coda di trasmissione denominata nell'attributo DEFXMITQ del gestore code locale.

Un *agent del canale dei messaggi* è un programma del canale associato alla coda di trasmissione e consegna il messaggio alla destinazione successiva. La destinazione successiva è il Gestore code a cui è collegato il canale dei messaggi. Non è necessariamente lo stesso gestore code della destinazione finale del messaggio. Quando il messaggio viene consegnato alla destinazione successiva, viene eliminato dalla coda di trasmissione. Il messaggio potrebbe dover passare attraverso molti gestori code durante il viaggio verso la destinazione finale. È necessario definire una coda di trasmissione in ogni gestore code lungo l'instradamento, ogni messaggio in attesa di essere trasmesso alla destinazione successiva. Una coda di trasmissione normale contiene i messaggi per la destinazione successiva, anche se i messaggi potrebbero avere destinazioni diverse. Una coda di trasmissione cluster contiene i messaggi per più destinazioni. Il `correlID` di ogni messaggio identifica il canale su cui viene inserito il messaggio per trasferirlo alla sua successiva destinazione.

È possibile definire diverse code di trasmissione in un gestore code. È possibile definire diverse code di trasmissione per la stessa destinazione, ognuna delle quali viene utilizzata per una classe di servizio differente. Ad esempio, è possibile che si desideri creare code di trasmissione diverse per messaggi di piccole dimensioni e messaggi di grandi dimensioni che vanno alla stessa destinazione. È quindi possibile trasferire i messaggi utilizzando canali di messaggi differenti, in modo che i messaggi di grandi dimensioni non trattengano i messaggi più piccoli. Tutti i messaggi per le

code cluster o gli argomenti cluster vengono posizionati nella singola coda di trasmissione cluster SYSTEM.CLUSTER.TRANSMIT.QUEUE, per impostazione predefinita. Come opzione, è possibile modificare il valore predefinito e separare il traffico di messaggi che va a gestori code di cluster differenti in code di trasmissione cluster differenti. Se si imposta l'attributo DEFCLXQ del gestore code su CHANNEL, ogni canale mittente del cluster crea una coda di trasmissione del cluster separata. In alternativa, è possibile definire manualmente le code di trasmissione del cluster per i canali mittente del cluster da utilizzare.

Le code di trasmissione possono attivare un MCA (message channel agent) per inviare i messaggi in avanti; consultare Avvio delle applicazioni IBM MQ utilizzando i trigger.

z/OS Su IBM MQ for z/OS, se si sta utilizzando l'accodamento all'interno del gruppo, la coda di trasmissione viene servita da un *agent di accodamento all'interno del gruppo*. Una coda di trasmissione condivisa viene utilizzata quando si utilizza l'accodamento all'interno del gruppo su IBM MQ for z/OS.

Code di iniziazione

Una *coda di iniziazione* è una coda locale in cui il gestore code inserisce un messaggio trigger quando si verifica un evento trigger su una coda dell'applicazione.

Un evento trigger è un evento che ha lo scopo di far sì che un programma avvii l'elaborazione di una coda. Ad esempio, un evento potrebbe essere più di 10 messaggi in arrivo. Per ulteriori informazioni su come funziona il trigger, consulta Avvio delle applicazioni IBM MQ utilizzando i trigger.

Coda di messaggi non recapitati (messaggio non consegnato)

Una *coda di messaggi non recapitabili (non recapitati)* è una coda locale in cui il gestore code inserisce i messaggi che non può consegnare.

Quando il gestore code inserisce un messaggio nella coda di messaggi non recapitabili, aggiunge un'intestazione al messaggio. Le informazioni di intestazione includono il motivo per cui il gestore code ha inserito il messaggio nella coda di messaggi non instradabili. Contiene anche la destinazione del messaggio originale, la data e l'ora in cui il gestore code ha inserito il messaggio nella coda di messaggi non recapitabili.

Le applicazioni possono anche utilizzare la coda per i messaggi che non possono consegnare. Per ulteriori informazioni, consultare Utilizzo della coda di messaggi non recapitabili.

Coda comandi di sistema

La *coda dei comandi di sistema* è una coda a cui le applicazioni opportunamente autorizzate possono inviare comandi IBM MQ . Queste code ricevono i comandi PCF, MQSC e CL, come supportato sulla piattaforma, in modo che il gestore code possa eseguirli.

z/OS Su IBM MQ for z/OS la coda è denominata SYSTEM.COMMAND.INPUT ; su altre piattaforme è denominato SYSTEM.ADMIN.COMMAND.QUEUE. I comandi accettati variano per piattaforma. Consultare Riferimento per i formati dei comandi programmabili per i dettagli.

Code predefinite di sistema

Le *code predefinite del sistema* contengono le definizioni iniziali delle code per il proprio sistema. Quando si crea una definizione di coda, il gestore code copia la definizione dalla coda predefinita di sistema appropriata. La creazione di una definizione di coda è diversa dalla creazione di una coda dinamica. La definizione della coda dinamica si basa sulla coda modello scelta come modello per la coda dinamica.

Code eventi

Le *code eventi* contengono messaggi di eventi. Questi messaggi vengono notificati dal gestore code o da un canale.

Code remote

Per un programma, una coda è *remota* se è di proprietà di un gestore code diverso rispetto a quello a cui è connesso il programma.

Se è stato stabilito un collegamento di comunicazione, un programma può inviare un messaggio a una coda remota. Un programma non può mai ricevere un messaggio da una coda remota.

L'oggetto di definizione della coda, creato quando si definisce una coda remota, contiene solo le informazioni necessarie al gestore code locale per individuare la coda a cui si desidera inviare il messaggio. Questo oggetto è noto come *definizione locale di una coda remota*. Tutti gli attributi della coda remota sono conservati dal gestore code che ne è proprietario, poiché è una coda locale per tale gestore code.

Quando si apre una coda remota, per identificare la coda è necessario specificare uno dei seguenti:

- Il nome della definizione locale che definisce la coda remota. Dal punto di vista di una applicazione, ciò equivale all'apertura di una coda locale. Non è necessario che un'applicazione sappia se una coda è locale o remota.

  Per creare una definizione locale di una coda remota su tutte le piattaforme tranne IBM i, utilizzare il comando DEFINE QREMOTE .

  IBM i Su IBM i, utilizzare il comando CRTMQMQ .
- Il nome del gestore code remoto e il nome della coda così come è noto al gestore code remoto.

Le definizioni locali delle code remote hanno tre attributi in aggiunta agli attributi comuni descritti in “Attributi delle code” a pagina 20. Questi tre attributi sono:

RemoteQName
Il nome con cui il gestore code proprietario della coda lo riconosce.

RemoteQmgrName
Il nome del gestore code proprietario.

XmitQName
Il nome della coda di trasmissione locale utilizzata durante l'inoltro dei messaggi ad altri gestori code.

Per ulteriori informazioni su questi attributi, consultare Attributi per le code.

Se si utilizza la chiamata MQINQ rispetto alla definizione locale di una coda remota, il gestore code restituisce solo gli attributi della definizione locale, ovvero il nome della coda remota, il nome del gestore code remoto e il nome della coda di trasmissione, non gli attributi della coda locale corrispondente nel sistema remoto.

Vedere anche Code di trasmissione.

Code alias

Una *coda alias* è un oggetto IBM MQ che è possibile utilizzare per accedere a un'altra coda o a un altro argomento. Ciò significa che più di un programma può gestire la stessa coda, accedendo ad essa utilizzando nomi diversi.

La coda risultante dalla risoluzione di un nome alias, noto come coda di base, può essere uno dei seguenti tipi di code, come supportato dalla piattaforma:

- Una coda locale
- La definizione locale di una coda remota.
- z/OS Una coda condivisa, che è un tipo di coda locale disponibile solo su IBM MQ for z/OS.
- Una coda predefinita
- Una coda dinamica

Un nome alias può anche essere risolto in un argomento. Se un'applicazione attualmente inserisce messaggi in una coda, è possibile effettuare la pubblicazione in un argomento rendendo il nome della coda un alias per l'argomento. Non è necessaria alcuna modifica al codice dell'applicazione.

Nota: Un alias non può essere risolto direttamente in un altro alias sullo stesso gestore code.

Un esempio di utilizzo delle code alias è per un responsabile di sistema per fornire autorizzazioni di accesso differenti al nome coda di base (ovvero, la coda in cui l'alias si risolve) e al nome coda alias. Ciò significa che un programma o un utente può essere autorizzato ad utilizzare la coda alias, ma non la coda di base.

In alternativa, l'autorizzazione può essere impostata per impedire le operazioni di inserimento per il nome alias, ma per consentire la coda di base.

In alcune applicazioni, l'utilizzo di code alias significa che gli amministratori di sistema possono modificare facilmente la definizione di un oggetto coda alias senza dover modificare l'applicazione.

IBM MQ effettua i controlli di autorizzazione rispetto al nome alias quando i programmi tentano di utilizzare tale nome. Non controlla che il programma sia autorizzato ad accedere al nome su cui si risolve l'alias. Un programma può quindi essere autorizzato ad accedere ad un nome coda alias, ma non al nome coda risolto.

Oltre agli attributi generali della coda descritti in “Code” a pagina 20, le code alias hanno un attributo **BaseQName** . Questo è il nome della coda di base in cui viene risolto il nome alias. Per una descrizione più completa di questo attributo, consultare BaseQName (MQCHAR48).

Gli attributi *InhibitGet* e **InhibitPut** (consultare “Code” a pagina 20) delle code alias appartengono al nome alias. Ad esempio, se il nome della coda alias ALIAS1 si risolve nel nome della coda di base BASE, le inibizioni su ALIAS1 influiscono solo su ALIAS1 e BASE non è inibito. Tuttavia, le inibizioni su BASE influenzano anche ALIAS1.

Gli attributi *DefPriority* e **DefPersistence** appartengono anche al nome alias. Quindi, ad esempio, è possibile assegnare diverse priorità predefinite a diversi alias della stessa coda base. Inoltre, è possibile modificare queste priorità senza dover modificare le applicazioni che utilizzano gli alias.

Code dinamiche e modello

Queste informazioni forniscono informazioni dettagliate sulle code dinamiche, sulle proprietà delle code dinamiche temporanee e permanenti, sugli utilizzi delle code dinamiche, su alcune considerazioni relative all'uso delle code dinamiche e sulle code modello.

Quando un programma applicativo emette una chiamata MQOPEN per l'apertura di una coda modello, il gestore code crea dinamicamente un'istanza di una coda locale con gli stessi attributi della coda modello. In base al valore del campo *DefinitionType* della coda modello, il gestore code crea una coda dinamica temporanea o permanente (consultare Creazione di code dinamiche).

Proprietà delle code dinamiche temporanee

Le *code dinamiche temporanee* hanno le seguenti proprietà:

- z/OS Non possono essere code condivise, accessibili dai gestori code in un gruppo di condivisione code.

  Notare che i gruppi di condivisione code sono disponibili solo su IBM MQ for z/OS.
- Contengono solo messaggi non persistenti.
- Sono irrecuperabili.
- Vengono eliminati quando il gestore code viene avviato.
- Vengono eliminati quando l'applicazione che ha emesso la chiamata MQOPEN che ha creato la coda chiude la coda o termina.
  - Se sulla coda sono presenti messaggi di cui è stato eseguito il commit, vengono eliminati.
  - Se sono presenti chiamate MQGET, MQPUT o MQPUT1 non sottoposte a commit in attesa rispetto alla coda in questo momento, la coda viene contrassegnata come eliminata logicamente e viene eliminata fisicamente (dopo che è stato eseguito il commit) come parte dell'elaborazione di chiusura o quando l'applicazione termina.

- Se la coda è in uso in questo momento (dalla creazione o da un'altra applicazione), la coda viene contrassegnata come logicamente eliminata e viene eliminata fisicamente solo quando viene chiusa dall'ultima applicazione che utilizza la coda.
- I tentativi di accedere a una coda eliminata logicamente (non per chiuderla) hanno esito negativo con codice motivo MQRC_Q_DELETED.
- MQCO_NONE, MQCO_DELETE e MQCO_DELETE_PURGE vengono tutti trattati come MQCO_NONE quando vengono specificati su una chiamata MQCLOSE per la chiamata MQOPEN corrispondente che ha creato la coda.

Proprietà delle code dinamiche permanenti

Le *code dinamiche permanenti* hanno le seguenti proprietà:

- Contengono messaggi persistenti o non persistenti.
- Sono recuperabili in caso di errori di sistema.
- Vengono eliminati quando un'applicazione (non necessariamente quella che ha emesso la chiamata MQOPEN che ha creato la coda) chiude correttamente la coda utilizzando l'opzione MQCO_DELETE o MQCO_DELETE_PURGE.
  - Una richiesta di chiusura con l'opzione MQCO_DELETE ha esito negativo se sulla coda sono ancora presenti messaggi (di cui è stato eseguito il commit o di cui non è stato eseguito il commit). Una richiesta di chiusura con l'opzione MQCO_DELETE_PURGE ha esito positivo anche se ci sono messaggi di cui è stato eseguito il commit sulla coda (i messaggi vengono eliminati come parte della chiusura), ma ha esito negativo se sono presenti chiamate MQGET, MQPUT o MQPUT1 non sottoposte a commit in sospeso sulla coda.
  - Se la richiesta di eliminazione ha esito positivo, ma la coda è in uso (dalla creazione o da un'altra applicazione), la coda viene contrassegnata come eliminata logicamente e viene eliminata fisicamente solo quando viene chiusa dall'ultima applicazione che utilizza la coda.
- Non vengono eliminati se chiusi da un'applicazione che non è autorizzata a eliminare la coda, a meno che l'applicazione di chiusura non abbia emesso la chiamata MQOPEN che ha creato la coda. I controlli di autorizzazione vengono eseguiti sull'identificativo utente (o sull'identificativo utente alternativo se è stato specificato MQOO_ALTERNATE_USER_AUTHORITY) utilizzato per convalidare la chiamata MQOPEN corrispondente.
- Possono essere eliminati allo stesso modo di una coda normale.

Utilizzi delle code dinamiche

È possibile utilizzare code dinamiche per:

- Le applicazioni che non richiedono code da conservare dopo la chiusura dell'applicazione.
- Applicazioni che richiedono che le risposte ai messaggi vengano elaborate da un'altra applicazione. Tali applicazioni possono creare in modo dinamico una coda di risposta aprendo una coda modello. Ad esempio, un'applicazione client può:
  1. Creare una coda dinamica.
  2. Fornire il suo nome nel campo **ReplyToQ** della struttura descrittore del messaggio di richiesta.
  3. Inserire la richiesta su una coda elaborata da un server.

Il server può quindi posizionare il messaggio di risposta sulla coda di risposta. Infine, il client potrebbe elaborare la risposta e chiudere la coda di risposta con l'opzione di eliminazione.

Considerazioni sull'utilizzo di code dinamiche

Considerare i seguenti punti quando si utilizzano le code dinamiche:

- In un modello client - server, ogni client deve creare e utilizzare la propria coda di risposta dinamica. Se una coda di risposta dinamica è condivisa tra più di un client, l'eliminazione della coda di risposta potrebbe essere ritardata a causa della presenza di un'attività di cui non è stato eseguito il commit in sospeso sulla coda o perché la coda è utilizzata da un altro client. Inoltre, la coda potrebbe essere

contrassegnata come eliminata logicamente e inaccessibile per le successive richieste API (diverse da MQCLOSE).

- Se l'ambiente dell'applicazione richiede che le code dinamiche siano condivise tra le applicazioni, assicurarsi che la coda sia chiusa (con l'opzione di eliminazione) solo quando è stato eseguito il commit di tutta l'attività rispetto alla coda. Deve essere l'ultimo utente. Ciò garantisce che l'eliminazione della coda non venga ritardata e riduce al minimo il periodo in cui la coda è inaccessibile perché è stata contrassegnata come logicamente eliminata.

Code modello

Una *coda modello* è un template di una definizione di coda che si utilizza quando si crea una coda dinamica.

È possibile creare dinamicamente una coda locale da un programma IBM MQ , denominando la coda modello che si desidera utilizzare come modello per gli attributi della coda. A quel punto è possibile modificare alcuni attributi della nuova coda. Tuttavia, non è possibile modificare **DefinitionType**. Se, ad esempio, si richiede una coda permanente, selezionare una coda modello con il tipo di definizione impostato su permanente. Alcune applicazioni di conversazione possono utilizzare code dinamiche per conservare le risposte alle loro query, poiché probabilmente non hanno bisogno di gestire queste code dopo aver elaborato le risposte.

Specificare il nome di una coda modello nel MQOD (*object descriptor*) della chiamata MQOPEN. Utilizzando gli attributi della coda modello, il gestore code crea dinamicamente una coda locale per l'utente.

È possibile specificare un nome (completo) per la coda dinamica o la radice di un nome (ad esempio, ABC) e consentire al gestore code di aggiungere una parte univoca oppure è possibile consentire al gestore code di assegnare un nome univoco completo. Se il gestore code assegna il nome, lo inserisce nella struttura MQOD.

Non è possibile emettere una chiamata MQPUT1 direttamente a una coda modello, ma è possibile emettere una MQPUT1 per la coda dinamica creata aprendo una coda modello.

MQSET e MQINQ non possono essere emessi rispetto a una coda modello. L'apertura di una coda modello con MQOO_INQUIRE o MQOO_SET comporta l'esecuzione di chiamate MQINQ e MQSET successive rispetto alla coda creata dinamicamente.

Gli attributi di una coda modello sono un sottoinsieme di quelli di una coda locale. Per una descrizione più completa, consultare Attributi per le code.

Code utilizzate per scopi specifici da parte di IBM MQ

IBM MQ utilizza alcune code locali per scopi specifici correlati al suo funzionamento.

È necessario definire queste code prima che IBM MQ possa utilizzarle.

Code di iniziazione

Le code di iniziazione sono code utilizzate nel trigger. Un gestore code inserisce un messaggio trigger su una coda di iniziazione quando si verifica un evento trigger. Un evento trigger è una combinazione logica di condizioni rilevate da un gestore code. Ad esempio, un evento trigger potrebbe essere generato quando il numero di messaggi su una coda raggiunge una profondità predefinita. Questo evento fa sì che il gestore code inserisca un messaggio trigger su una coda di avvio specificata. Questo messaggio trigger viene richiamato da un *controllo trigger*, un'applicazione speciale che controlla una coda di iniziazione. Il controllo trigger, quindi, avvia il programma applicativo specificato nel messaggio trigger.

Se un gestore code deve utilizzare l'attivazione, è necessario definire almeno una coda di avvio per tale gestore code. Consultare Gestione degli oggetti per l'attivazione di, runmqtrme Avvio delle applicazioni IBM MQ mediante i trigger

Code di trasmissione

Le code di trasmissione sono code che memorizzano temporaneamente i messaggi destinati a un gestore code remoto. È necessario definire almeno una coda di trasmissione per ogni gestore code

remoto a cui il gestore code locale deve inviare direttamente i messaggi. Queste code vengono utilizzate anche nell'amministrazione remota; consultare Gestione remota da un gestore code locale. Per informazioni sull'utilizzo delle code di trasmissione nell'accodamento distribuito, consultare IBM MQ tecniche di accodamento distribuito.

Ogni gestore code può avere una coda di trasmissione predefinita. Se un gestore code che non fa parte di un cluster inserisce un messaggio in una coda remota, l'azione predefinita consiste nell'utilizzare la coda di trasmissione predefinita. Se è presente una coda di trasmissione con lo stesso nome del gestore code di destinazione, il messaggio viene inserito in tale coda di trasmissione. Se esiste una definizione di alias del gestore code, in cui il parametro **RQMNAME** corrisponde al gestore code di destinazione e viene specificato il parametro **XMITQ** , il messaggio viene posizionato nella coda di trasmissione denominata da **XMITQ**. Se non esiste alcun parametro **XMITQ** , il messaggio viene inserito nella coda locale indicata nel messaggio.

Code di trasmissione cluster
Ogni gestore code all'interno di un cluster ha una coda di trasmissione cluster denominata SYSTEM.CLUSTER.TRANSMIT.QUEUEe una coda di trasmissione cluster modello, SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE. Le definizioni di queste code vengono create per impostazione predefinita quando si definisce un gestore code. Se l'attributo del gestore code, **DEFCLXQ**, è impostato su CHANNEL, viene creata automaticamente una coda di trasmissione del cluster dinamica permanente per ogni canale mittente del cluster creato. Le code sono denominate SYSTEM.CLUSTER.TRANSMIT. *ChannelName*. È anche possibile definire manualmente le code di trasmissione del cluster.

Un gestore code che fa parte del cluster invia messaggi su una di tali code ad altri gestori code che si trovano nello stesso cluster.

Durante la risoluzione dei nomi, una coda di trasmissione del cluster ha la precedenza sulla coda di trasmissione predefinita e una specifica coda di trasmissione del cluster ha la precedenza su SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Code di messaggi non recapitabili
Una coda di messaggi non recapitabili (undelivered-message) è una coda che memorizza i messaggi che non possono essere instradati alle destinazioni corrette. Un messaggio non può essere instradato quando, ad esempio, la coda di destinazione è piena. La coda di messaggi non instradabili fornita è denominata SYSTEM.DEAD.LETTER.QUEUE.

Per l'accodamento distribuito, definire una coda di messaggi non instradabili su ogni gestore code coinvolto.

Code comandi
La coda comandi, SYSTEM.ADMIN.COMMAND.QUEUE, è una coda locale a cui le applicazioni adeguatamente autorizzate possono inviare comandi MQSC per l'elaborazione. Questi comandi vengono quindi richiamati da un componente IBM MQ denominato server dei comandi. Il server dei comandi convalida i comandi, passa quelli validi per l'elaborazione da parte del gestore code e restituisce tutte le risposte alla coda di risposta appropriata.

Una coda comandi viene creata automaticamente per ogni gestore code quando viene creato tale gestore code.

Code di risposta
Quando un'applicazione invia un messaggio di richiesta, l'applicazione che riceve il messaggio può restituire un messaggio di risposta all'applicazione mittente. Questo messaggio viene inserito in una coda, denominata coda di risposta, che normalmente è una coda locale all'applicazione di invio. Il nome della coda di risposta è specificato dall'applicazione mittente come parte del descrittore del messaggio.

Code eventi
Gli eventi di strumentazione possono essere utilizzati per monitorare i gestori code indipendentemente dalle applicazioni MQI.

Quando si verifica un evento di strumentazione, il gestore code inserisce un messaggio evento su una coda eventi. Questo messaggio può quindi essere letto da un'applicazione di controllo, che potrebbe informare un amministratore o avviare alcune azioni correttive se l'evento indica un problema.

Nota: Gli eventi trigger sono differenti dagli eventi di strumentazione. Gli eventi trigger non sono causati dalle stesse condizioni e non generano messaggi di evento.

Per ulteriori informazioni sugli eventi di strumentazione, vedi Eventi di strumentazione.

Gestori code

Introduzione ai *gestori code* e ai servizi di accodamento che forniscono alle applicazioni.

Un programma deve avere una connessione a un gestore code prima di poter utilizzare i servizi di tale gestore code. Un programma può effettuare questa connessione in modo esplicito (utilizzando la chiamata MQCONN o MQCONNX) oppure la connessione potrebbe essere effettuata in modo implicito (ciò dipende dalla piattaforma e dall'ambiente in cui il programma è in esecuzione).

Un gestore code IBM MQ assicura le seguenti azioni:

- Gli attributi dell'oggetto vengono modificati in base ai comandi ricevuti.
- Gli eventi speciali come gli eventi trigger o gli eventi di strumentazione vengono generati quando vengono soddisfatte le condizioni appropriate.
- I messaggi vengono inseriti sulla coda corretta, come richiesto dall'applicazione che effettua la chiamata MQPUT . L'applicazione viene informata se ciò non è possibile e viene fornito un codice di errore appropriato.

Ciascuna coda appartiene a un singolo gestore code e viene detta *coda locale* per tale gestore code. Il gestore code a cui è connessa un'applicazione viene detto *gestore code locale* per tale applicazione. Per l'applicazione, le code che appartengono al gestore code locale sono code locali.

Una *coda remota* è una coda che appartiene a un altro gestore code. Un *gestore code remoto* è un gestore code diverso dal gestore code locale. Un gestore code remoto può esistere su una macchina remota nella rete o sulla stessa macchina del gestore code locale. IBM MQ supporta più gestori code sulla stessa macchina.

Un oggetto gestore code può essere utilizzato in alcune chiamate MQI. Ad esempio, è possibile richiedere informazioni sugli attributi dell'oggetto gestore code utilizzando la chiamata MQI MQINQ.

Attributi dei gestori code

A ciascun gestore code è associata una serie di attributi (o proprietà) che ne definiscono le caratteristiche.Alcuni degli attributi di un gestore code vengono corretti quando viene creato; è possibile modificarne altri utilizzando i comandi IBM MQ . È possibile richiedere informazioni sui valori di tutti gli attributi, eccetto quelli utilizzati per la crittografia TLS (Transport Layer Security), utilizzando la chiamata MQINQ.

Gli attributi fissi includono:

- Il nome del gestore code
- La piattaforma su cui viene eseguito il gestore code (ad esempio, Windows)
- Il livello di comandi di controllo del sistema supportato dal gestore code
- La priorità massima che è possibile assegnare ai messaggi elaborati dal gestore code
- Il nome della coda a cui i programmi possono inviare comandi IBM MQ
- La lunghezza massima dei messaggi che il gestore code può elaborare z/OS (fissa solo in IBM MQ for z/OS)
- Se il gestore code supporta il punto di sincronizzazione quando i programmi inserono e ricevono i messaggi

Gli attributi *modificabili* includono:

- Una descrizione testuale del gestore code
- L'identificativo della serie di caratteri che il gestore code utilizza per le stringhe di caratteri quando elabora le chiamate MQI
- L'intervallo di tempo utilizzato dal gestore code per limitare il numero di messaggi trigger
- z/OS L'intervallo di tempo utilizzato dal gestore code per determinare la frequenza con cui le code devono essere sottoposte a scansione per i messaggi scaduti (solo IBM MQ for z/OS)
- Il nome della coda di messaggi non recapitabili (messaggi non recapitati) del gestore code
- Il nome della coda di trasmissione predefinita del gestore code
- Il numero massimo di handle aperti per una connessione
- Abilitazione e disabilitazione di varie categorie di report di eventi
- Il numero massimo di messaggi di cui non è stato eseguito il commit all'interno di un'unità di lavoro

Gestori code e gestione del carico di lavoro

È possibile impostare un cluster di gestori code con più di una definizione per la stessa coda (ad esempio, i gestori code nel cluster potrebbero essere cloni l'uno dell'altro). I messaggi per una particolare coda possono essere gestiti da qualsiasi gestore code che ospita un'istanza della coda. Un algoritmo di gestione del carico di lavoro decide quale gestore code gestisce il messaggio e quindi distribuisce il carico di lavoro tra i gestori code; per ulteriori informazioni, fare riferimento a Algoritmo di gestione del carico di lavoro del cluster .

Canali

Un *canale* è un collegamento di comunicazione logico, utilizzato dai gestori code distribuiti, tra un server IBM MQ MQI client e un server IBM MQ o tra due server IBM MQ .

I canali vengono utilizzati per spostare i messaggi da un gestore code a un altro e proteggono le applicazioni dai protocolli di comunicazione sottostanti. I gestori code potrebbero esistere sullo stesso sistema, su sistemi differenti sulla stessa piattaforma o su piattaforme differenti. I messaggi inviati possono avere origine da molti luoghi:

- Programmi di applicazione scritti dall'utente che trasferiscono i dati da un nodo all'altro.
- Applicazioni di gestione scritte dall'utente che utilizzano i comandi PCF o MQAI.
- Il IBM MQ Explorer.
- Gestori code che inviano messaggi di evento di strumentazione ad un altro gestore code.
- Gestori code che inviano comandi di gestione remoti ad un altro gestore code. Ad esempio, utilizzando i comandi MQSC o administrative REST API.

Un canale ha due definizioni: una ad ogni estremità della connessione. Affinché i gestori code comunichino tra loro, è necessario definire un oggetto canale sul gestore code che deve inviare i messaggi e un altro, complementare, sul gestore code che deve riceverli. Lo stesso *nome canale* deve essere utilizzato ad ogni estremità della connessione e il tipo di canale utilizzato deve essere compatibile.

Ci sono tre categorie di canali in IBM MQ, con diversi tipi di canali all'interno di queste categorie:

- I canali di messaggi, che sono unidirezionali, e trasferiscono i messaggi da un gestore code all'altro.
- I canali MQI, che sono bidirezionali e trasferiscono le chiamate MQI da un IBM MQ MQI client a un gestore code e le risposte da un gestore code a un client IBM MQ .
- I canali AMQP, che sono bidirezionali e collegano il client AMQP a un gestore code su una macchina server. IBM MQ utilizza canali AMQP per trasferire chiamate e risposte AMQP tra applicazioni AMQP e gestori code

Canali dei messaggi

Lo scopo di un canale di messaggi è di trasferire i messaggi da un gestore code all'altro. I canali di messaggi non sono richiesti dall'ambiente del server client.

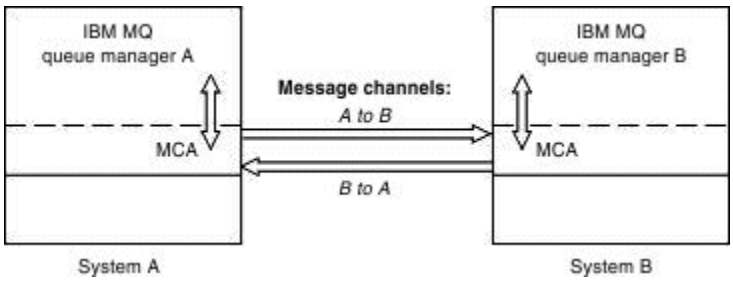


Figura 2. Canali di messaggi tra due gestori code

Un canale di messaggi è un collegamento unidirezionale. Se si desidera che un gestore code remoto risponda ai messaggi inviati da un gestore code locale, è necessario impostare un secondo canale per inviare le risposte al gestore code locale.

Un canale dei messaggi connette due gestori code utilizzando gli MCA (*message channel agent*). Esiste un agente del canale dei messaggi ad ogni estremità di un canale. È possibile consentire a un MCA di trasferire i messaggi utilizzando più thread. Questo processo è noto come *pipelining*. Pipelining consente all'MCA di trasferire i messaggi in modo più efficiente, migliorando le prestazioni del canale. Per ulteriori informazioni sulla pipeline, vedi Attributi dei canali.

Per ulteriori informazioni sui canali, consultare Chiamate di uscita del canale e strutture datie “Componenti di accodamento distribuiti” a pagina 47.

Canali MQI

Un canale MQI (Message Queue Interface) connette un IBM MQ MQI client a un gestore code su una macchina server e viene stabilito quando si emette una chiamata `MQCONN` o `MQCONNX` da un'applicazione IBM MQ MQI client .

Si tratta di un collegamento bidirezionale e viene utilizzato solo per il trasferimento di chiamate e risposte MQI, incluse le chiamate `MQPUT` che contengono dati di messaggio e le chiamate `MQGET` che risultano nella restituzione dei dati di messaggio. Esistono diversi modi per creare e utilizzare le definizioni di canale (vedere Definizione dei canali MQI).

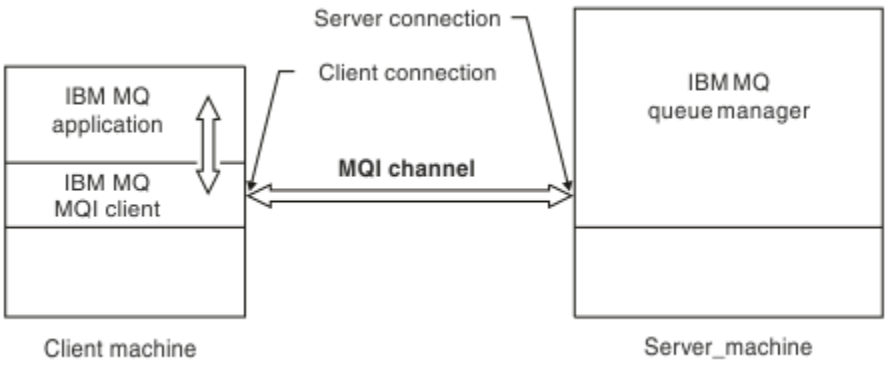


Figura 3. Connessione client e connessione server su un canale MQI

z/OS Un canale MQI può essere utilizzato per collegare un client a un singolo gestore code o a un gestore code che fa parte di un gruppo di condivisione code (consultare Connessione di un client a un gruppo di condivisione code).

Ci sono due tipi di canale per definizioni di canale MQI. Definiscono il canale MQI bidirezionale.

Canale connessione client
Questo tipo è per IBM MQ MQI client.

Canale di connessione server
Questo tipo è per il server che esegue il gestore code, con cui l'applicazione IBM MQ , in esecuzione in un ambiente IBM MQ MQI client , deve comunicare.

Canali AMQP

Esiste un solo tipo di canale AMQP.

Il canale è utilizzato per connettere un'applicazione di messaggistica AMQP a un gestore code, abilitando l'applicazione allo scambio di messaggi con le applicazioni IBM MQ. Un canale AMQP consente di sviluppare un'applicazione utilizzando MQ Light e di distribuirla quindi come un'applicazione enterprise, avvalendosi delle funzioni di livello enterprise fornite da IBM MQ.

Canali di connessione client

I *canali di connessione client* sono oggetti che forniscono un percorso di comunicazione da un IBM MQ MQI client a un gestore code.

I canali di connessione client vengono utilizzati nell'accodamento distribuito per spostare i messaggi tra un gestore code e un client. Proteggono le applicazioni dai protocolli di comunicazione sottostanti. Il client potrebbe esistere sulla stessa piattaforma o su una piattaforma diversa dal gestore code.

Definizioni canale

Consultare “Definizioni canale” a pagina 32 per descrizioni di ciascun tipo di canale.

Concetti correlati

“Accodamento distribuito e cluster” a pagina 43
L'accodamento distribuito significa inviare messaggi da un gestore code ad un altro. Il gestore code di ricezione può trovarsi sulla stessa macchina o su un'altra; nelle vicinanze o dall'altra parte del mondo. Può essere in esecuzione sulla stessa piattaforma del gestore code locale o su una qualsiasi delle piattaforme supportate da IBM MQ. È possibile definire manualmente tutte le connessioni in un ambiente di accodamento distribuito oppure è possibile creare un cluster e consentire a IBM MQ di definire gran parte dei dettagli di connessione.

Panoramica su Interfaccia coda messaggi

Attività correlate

Gestione di oggetti IBM MQ remoti
Arresto dei canali MQI
Configurazione delle connessioni tra il server e il client

Riferimenti correlati

Chiamate di uscita canale e strutture dati
“Comunicazione” a pagina 36
IBM MQ MQI clients utilizzare i canali MQI per comunicare con il server.

Definizioni canale

Tabelle che descrivono i diversi tipi di canali di messaggi e canali MQI utilizzati da IBM MQ .

Quando si parla di canali di messaggi, la parola canale spesso viene utilizzata come sinonimo per la definizione di canale. Di solito il contesto aiuta a capire se si parla di una canale completo, che ha due estremità, o di una definizione di canale, che ha una estremità sola.

Canali dei messaggi

Le definizioni dei canali di messaggi possono essere di uno dei seguenti tipi:

Tipo di definizione di canali dei messaggi	Descrizione
Mittente	Un canale mittente è un canale dei messaggi che il gestore code utilizza per inviare i messaggi ad altri gestori code. Per inviare i messaggi mediante un canale mittente, è necessario creare, sull'altro gestore code, un canale ricevente con lo stesso nome del canale mittente. È anche possibile utilizzare i canali del mittente con i canali del richiedente, se si implementa un meccanismo di "callback".
Server	Un canale server è un canale di messaggi che il gestore code utilizza per inviare i messaggi ad altri gestori code. Per inviare i messaggi mediante un canale server, è anche necessario creare, sull'altro gestore code, un canale ricevente con lo stesso nome del canale server. I canali server possono essere utilizzati anche con i canali richiedenti. In questo caso, la definizione di canale richiedente richiede l'avvio della definizione di canale server all'altra estremità del canale. Il server invia i messaggi al richiedente. Quindi, il server inizia la comunicazione, attiva fino a che il nome della connessione del canale partner è noto.
Ricevente	Un canale ricevente è un canale di messaggi utilizzato dal gestore code per ricevere i messaggi dagli altri gestori code. Per ricevere i messaggi mediante un canale ricevente, è anche necessario creare, sull'altro gestore code, un canale mittente o server con lo stesso nome di questo canale ricevente.
Richiedente	Un canale Richiedente è un canale di messaggi utilizzato dal gestore code per ricevere messaggi da altri gestori code. Un canale RIchiedente può richiedere che sia avviato il canale partner definito all'estremità remota. Se il canale partner è un canale Server, il canale Server accetta la richiesta di avvio e inizia a inviare messaggi, dalla coda di trasmissione identificata nella definizione di canale Server, al canale Richiedente. Se il canale partner è un canale Mittente, il canale Mittente accetta la richiesta di avvio ma chiude quindi la connessione con il Richiedente. Il canale Mittente viene quindi avviato, negozia una sessione con il canale Richiedente partner e inizia a inviare messaggi dalla coda di trasmissione identificata nella definizione del canale Mittente. Quest'ultimo caso fornisce fondamentalmente un meccanismo di richiamo, dal momento che il canale Richiedente richiede al canale Mittente di richiamare.

Tipo di definizione di canali dei messaggi	Descrizione
Mittente del cluster	Una definizione di canale mittente del cluster (CLUSSDR) definisce l'estremità di invio su cui il gestore code di un cluster può inviare le informazioni sul cluster a uno dei repository completi. Il canale mittente del cluster viene utilizzato per notificare al repository le modifiche apportate allo stato del gestore code, ad esempio nel caso dell'aggiunta o della rimozione di una coda. Essa viene utilizzata anche per trasmettere i messaggi. I gestori code con repository completo hanno canali mittente cluster che fanno riferimento l'uno all'altro. Tali canali vengono utilizzati per comunicare le modifiche allo stato del cluster. Non è particolarmente importante conoscere a quale repository completo una definizione del canale CLUSSDR del gestore code fa riferimento. Una volta stabilito il contatto iniziale, vengono definiti automaticamente ulteriori oggetti del gestore code del cluster in modo che il gestore code possa inviare le informazioni sul cluster a ogni repository completo e i messaggi a ogni gestore code.
Ricevente del cluster	Una definizione di canale ricevente del cluster (CLUSRCVR) definisce l'estremità di ricezione su cui il gestore code di un cluster può ricevere le informazioni sul cluster da altri gestori code. Un canale ricevente del cluster trasporta inoltre le informazioni sul destinate al repository. Definendo il canale ricevente del cluster, il gestore code indica agli altri gestori code che è disponibile per la ricezione di messaggi. È necessario disporre di almeno un canale ricevente del cluster per ogni gestore code del cluster.

Per ogni canale, è necessario definire entrambe le estremità in modo da avere una definizione di canale per ognuna di esse. Le due estremità del canale devono essere di tipo compatibile.

È possibile avere le seguenti combinazioni di definizioni di canali:

- Mittente-Ricevente
- Server-Ricevente
- Richiedente-Server
- Richiedente-Mittente (callback)
- Mittente del cluster-Ricevente del cluster

Agente canale dei messaggi (MCA, message channel agent)

Ogni definizione di canale che viene creata appartiene a un determinato gestore code. Un gestore code può avere diversi canali, dello stesso tipo o di tipo differente. A ogni estremità del canale è presente un programma, un MCA (message channel agent), ovvero un agent dei messaggi dei canali. A un'estremità del canale, l'MCA chiamante prende i messaggi dalla coda di trasmissione e li inoltra sul canale. All'altra estremità del canale, l'MCA rispondente riceve i messaggi e li consegna al gestore code remoto.

Un MCA chiamante può essere associato a un canale mittente, server o richiedente. Un MCA rispondente può essere associato a qualsiasi tipo di canale.

IBM MQ supporta le seguenti combinazioni di tipi di canale alle due estremità di una connessione:

Chiamante		Direzione del flusso di messaggi	Rispondente	
Tipo di canale	**È necessario il listener?**		**È necessario il listener?**	**Tipo di canale**
Mittente	No	Dal chiamante al rispondente	Sì	Ricevente
Server	No	Dal chiamante al rispondente	Sì	Ricevente
Server	No	Dal chiamante al rispondente	Sì	Richiedente
Richiedente	No	Dal rispondente al chiamante	Sì	Server
Richiedente	Sì	Dal rispondente al chiamante	Sì	Mittente

Canali MQI

I canali MQI possono essere di uno dei seguenti tipi:

Tipo canale MQI	Descrizione
Connessione server	Un canale di connessione server è un canale MQI bidirezionale utilizzato per connettere un client IBM MQ a un server IBM MQ. Il canale di connessione server è l'estremità server del canale.
Connessione client	Un canale di connessione client è un canale MQI bidirezionale utilizzato per connettere un client IBM MQ a un server IBM MQ. IBM MQ Explorer utilizza anche le connessioni client per connettersi ai gestori code remoti. Il canale di connessione client è l'estremità client del canale. Quando si crea un canale di connessione client, viene creato un file sul computer su cui è presente il gestore code. È necessario quindi copiare il file di connessione client al computer client IBM MQ.

Multi ***Supporto thread multipli - pipelining***

Facoltativamente, è possibile consentire a un MCA (message channel agent) di trasferire i messaggi utilizzando più thread. Questo processo, denominato *pipelining*, consente all'MCA di trasferire i messaggi in modo più efficiente, con un minor numero di stati di attesa, migliorando le prestazioni del canale. Ogni MCA è limitato a un massimo di due thread.

Si controlla la pipeline con il parametro *PipeLineLength* nel file qm.ini . Questo parametro viene aggiunto alla sezione Canali.

Nota: Il pipelining è efficace solo per i canali TCP/IP.

Quando si utilizza la pipeline, i gestori code su entrambe le estremità del canale devono essere configurati in modo che *PipeLineLength* sia maggiore di 1.

Considerazioni sull'uscita del canale

La pipeline può causare il malfunzionamento di alcuni programmi di uscita, perché:

- Le uscite potrebbero non essere richiamate in modo seriale.

- Le uscite possono essere richiamate alternativamente da thread differenti.

Verificare la progettazione dei programmi di uscita prima di utilizzare la pipeline:

- Le uscite devono essere rientranti in tutte le fasi della loro esecuzione.
- Quando si utilizzano chiamate MQI, ricordare che non è possibile utilizzare la stessa gestione MQI quando l'exit viene richiamata da thread differenti.

Considerare un'uscita del messaggio che apre una coda e ne utilizza l'handle per le chiamate MQPUT su tutte le chiamate successive dell'uscita. L'operazione non riesce in modalità pipeline perché l'uscita viene richiamata da thread differenti. Per evitare questo errore, mantenere un handle di coda per ciascun thread e controllare l'ID thread ogni volta che viene richiamata l'uscita.

Comunicazione

IBM MQ MQI clients utilizzare i canali MQI per comunicare con il server.

Una definizione di canale deve essere creata sia all'estremità IBM MQ MQI client che al server della connessione. Come creare le definizioni di canale è spiegato in Definizione dei canali MQI.

I protocolli di trasmissione possibili sono mostrati nella seguente tabella:

Tabella 1. Protocolli di trasmissione per canali MQI

Piattaforma client	LU 6.2	TCP/IP	NetBIOS	SPX
IBM i IBM i		Sì		
Linux AIX SistemiAIX and Linux	Sì[1]	Sì		
Windows Windows	Sì	Sì	Sì	Sì

Nota:

1. Linux LU6.2 non è supportato sulle piattaforme seguenti:
   - Linux (piattaforma POWER)
   - Linux (piattaformax86-64)
   - Linux (piattaformazSeries s390x)

Protocolli di trasmissione - combinazione di IBM MQ MQI client e piattaforme server mostra le possibili combinazioni di IBM MQ MQI client e piattaforme server, utilizzando questi protocolli di trasmissione.

Un'applicazione IBM MQ su IBM MQ MQI client può utilizzare tutte le chiamate MQI nello stesso modo di quando il gestore code è locale. **MQCONN** o **MQCONNX** associa l'applicazione IBM MQ al gestore code selezionato, creando un *handle di connessione*. Le altre chiamate che utilizzano tale handle di connessione vengono quindi elaborate dal gestore code connesso. La comunicazione IBM MQ MQI client richiede una connessione attiva tra il client e il server, a differenza della comunicazione tra i gestori code, che è indipendente dalla connessione e dal tempo.

Il protocollo di trasmissione viene specificato utilizzando la definizione di canale e non influisce sull'applicazione. Ad esempio, un'applicazione Windows può connettersi a un gestore code su TCP/IP e a un altro gestore code su NetBIOS.

Considerazioni sulle prestazioni

Il protocollo di trasmissione utilizzato potrebbe influire sulle prestazioni del sistema client e server IBM MQ . In alcune situazioni in cui la trasmissione è lenta, è possibile utilizzare la compressione del canale IBM MQ .

Denominazione degli oggetti IBM MQ

La convenzione di denominazione adottata per gli oggetti IBM MQ dipende dall'oggetto. Anche il nome delle macchine e gli ID utente utilizzati con IBM MQ sono soggetti ad alcune limitazioni di denominazione.

Ciascuna istanza di un gestore code è nota con il relativo nome. Questo nome deve essere univoco nella rete di gestori code interconnessi, in modo che un gestore code possa identificare in modo non ambiguo il gestore code di destinazione a cui viene inviato un determinato messaggio.

Per gli altri tipi di oggetti, ogni oggetto ha un nome associato e può essere indicato con tale nome. Questi nomi devono essere univoci all'interno di un gestore code e di un tipo di oggetto. Ad esempio, è possibile avere una coda e un processo con lo stesso nome, ma non è possibile avere due code con lo stesso nome.

In IBM MQ, i nomi possono avere un massimo di 48 caratteri, ad eccezione di *canali* che hanno un massimo di 20 caratteri. Per ulteriori informazioni sulla denominazione degli oggetti IBM MQ , consultare “Regole per la denominazione degli oggetti IBM MQ” a pagina 37.

Anche il nome delle macchine e gli ID utente utilizzati con IBM MQ sono soggetti ad alcune limitazioni di denominazione:

- Assicurarsi che il nome della macchina non contenga spazi. IBM MQ non supporta nomi di macchine che contengono spazi. Se si installa IBM MQ su tale macchina, non è possibile creare alcun gestore code.
- Per le autorizzazioni IBM MQ , i nomi degli ID utente e dei gruppi non devono superare i 20 caratteri (gli spazi non sono consentiti).
- Windows Un server di IBM MQ for Windows non supporta la connessione di un IBM MQ MQI client se il client è in esecuzione con un ID utente che contiene il carattere @, ad esempio, abc@d.

Concetti correlati

“IBM MQNOMI DI FILE” a pagina 40
Ogni oggetto gestore code, coda, definizione del processo, elenco nomi, canale, canale di connessione client, listener, servizio e informazioni di autenticazione di IBM MQ è rappresentato da un file. Poiché i nomi oggetto non sono necessariamente nomi file validi, il gestore code converte il nome oggetto in un nome file valido, se necessario.

Riferimenti correlati

“Regole per la denominazione degli oggetti IBM MQ” a pagina 37
I nomi oggetto IBM MQ hanno lunghezze massime e sono sensibili al maiuscolo / minuscolo. Non tutti i caratteri sono supportati per ogni tipo di oggetto e molti oggetti hanno regole relative all'unicità dei nomi.

Regole per la denominazione degli oggetti IBM MQ

I nomi oggetto IBM MQ hanno lunghezze massime e sono sensibili al maiuscolo / minuscolo. Non tutti i caratteri sono supportati per ogni tipo di oggetto e molti oggetti hanno regole relative all'unicità dei nomi.

Esistono molti tipi diversi di oggetti IBM MQ e gli oggetti di ogni tipo possono avere lo stesso nome perché esistono in spazi dei nomi oggetto separati: ad esempio, una coda locale e un canale mittente possono avere entrambi lo stesso nome. Tuttavia, un oggetto non può avere lo stesso nome di un altro oggetto nello stesso spazio dei nomi. Ad esempio, una coda locale non può avere lo stesso nome di una coda modello, mentre un canale mittente non può avere lo stesso nome di un canale ricevente.

I seguenti oggetti IBM MQ esistono in spazi dei nomi oggetto separati:

- Informazioni di autenticazione
- Canale
- Canale client
- Listener
- Elenco nomi
- Processo
- Coda

- Servizio
- Classe di memoria
- Sottoscrizione
- Argomento

Lunghezza carattere dei nomi oggetto

In generale, i nomi oggetto IBM MQ possono avere una lunghezza massima di 48 caratteri. Questa regola si applica ai seguenti oggetti:

- Informazioni di autenticazione
- Cluster
- Listener
- Elenco nomi
- Definizione di processo
- Coda
- Gestore code
- Servizio
- Sottoscrizione
- Argomento

Esistono delle limitazioni:

1. z/OS Sui sistemi z/OS , i gestori code devono avere una lunghezza massima di 4 caratteri e devono contenere solo caratteri maiuscoli e numerici.
2. La lunghezza massima dei nomi oggetto canale e dei nomi canale di connessione client è 20 caratteri. Consultare Definizione dei canali per ulteriori informazioni sui canali.
3. Le stringhe argomento possono essere un massimo di 10240 byte. Tutti i nomi oggetto IBM MQ sono sensibili al maiuscolo / minuscolo.
4. I nomi delle sottoscrizioni possono avere una lunghezza massima di 10240 byte e possono contenere spazi.
5. La lunghezza massima dei nomi delle classi di memoria è di 8 caratteri.
6. La lunghezza massima dei nomi della struttura CF è 12 caratteri.

Caratteri nei nomi oggetto

I caratteri validi per i nomi oggetto IBM MQ sono:

Caratteri	Limitazioni
Maiuscolo A - Z	• Nessuna

Caratteri	Limitazioni
Minuscolo a - z	• Negli script MQSC, i nomi con caratteri minuscoli devono essere racchiusi tra virgolette singole. Ciò impedisce che i caratteri minuscoli vengano ripiegati in maiuscolo. • I sistemi che utilizzano EBCDIC Katakana non possono utilizzare caratteri a - z minuscoli nei nomi oggetto. • z/OS Potrebbero esserci delle limitazioni quando si utilizzano caratteri minuscoli sui sistemi z/OS , ad esempio, i nomi dei gestori code non possono contenere caratteri minuscoli. • IBM i Su sistemi IBM i quando si utilizzano comandi CL, i nomi con caratteri minuscoli devono essere racchiusi tra virgolette singole. Ciò impedisce che i caratteri minuscoli vengano ripiegati in maiuscolo.
Numeri 0-9	• Nessuna
Punto (.)	• Nessuna
Trattino basso (_)	• Multi Nessuna • z/OS Evitare di utilizzare nomi con caratteri di sottolineatura iniziali o finali perché non possono essere gestiti dalle operazioni e dai pannelli di controllo IBM MQ for z/OS .
Barra (/)	• Windows Sui sistemi Windows , il primo carattere di un nome gestore code non può essere una barra. • IBM i Su sistemi IBM i quando si utilizzano comandi CL, i nomi contenenti una barra devono essere racchiusi tra virgolette singole. • z/OS Nessuna
Segno percentuale (%)	• ALW Nessuna • z/OS Se si utilizza RACF come gestore della sicurezza esterno per IBM MQ for z/OS, non utilizzare% nei nomi oggetto perché i nomi non sono inclusi nelle verifiche di sicurezza quando si utilizzano i profili generici RACF . • IBM i Su sistemi IBM i quando si utilizzano comandi CL, i nomi che contengono un segno di percentuale devono essere racchiusi tra virgolette singole.

Ci sono anche alcune regole generali riguardanti i caratteri sui nomi degli oggetti:

1. Non sono consentiti spazi vuoti iniziali o centrali.
2. I caratteri della lingua nazionale non sono consentiti.
3. Qualsiasi nome inferiore alla lunghezza del campo completo può essere riempito a destra con spazi. Tutti i nomi brevi restituiti dal gestore code vengono sempre riempiti a destra con spazi.

Nomi coda

Il nome di una coda è composto da due parti:

- Il nome di un gestore code
- Il nome locale della coda così come è noto a tale gestore code

Ogni parte del nome della coda è lunga 48 caratteri.

Per fare riferimento a una coda locale, è possibile omettere il nome del gestore code (sostituendolo con caratteri vuoti o utilizzando un carattere null iniziale). Tuttavia, tutti i nomi coda restituiti a un programma da IBM MQ contengono il nome del gestore code.

z/OS Una coda condivisa, accessibile a qualsiasi gestore code nel relativo gruppo di condivisione code, non può avere lo stesso nome di una coda locale non condivisa nello stesso gruppo di condivisione code. Questa limitazione evita la possibilità che un'applicazione apra erroneamente una coda condivisa quando intendeva aprire una coda locale o viceversa. Le code condivise e i gruppi di condivisione code sono disponibili solo su IBM MQ for z/OS.

Per fare riferimento ad una coda remota, un programma deve includere il nome del gestore code nel nome completo della coda oppure deve esistere una definizione locale della coda remota.

Quando un'applicazione utilizza un nome coda, tale nome può essere il nome di una coda locale (o un alias a uno) o il nome di una definizione locale di una coda remota, ma l'applicazione non ha bisogno di sapere quale, a meno che non abbia bisogno di richiamare un messaggio dalla coda (quando la coda deve essere locale). Quando l'applicazione apre l'oggetto coda, la chiamata MQOPEN esegue una funzione di risoluzione dei nomi per stabilire su quale coda eseguire le operazioni successive. Il significato di ciò è che l'applicazione non ha alcuna dipendenza integrata su code particolari definite in posizioni particolari in una rete di gestori code. Pertanto, se un amministratore di sistema riposiziona le code nella rete e ne modifica le definizioni, non è necessario modificare le applicazioni che utilizzano tali code.

Nomi oggetto riservati

I nomi oggetto che iniziano con SYSTEM. sono riservati agli oggetti definiti dal gestore code. È possibile utilizzare i comandi **Alter**, **Define**e **Replace** per modificare queste definizioni di oggetti in modo da adattarle alla propria installazione. I nomi definiti per IBM MQ sono elencati in modo completo in Nomi coda.

z/OS Su IBM MQ for z/OS, il nome della struttura dell'applicazione Coupling Facility CSQSYSAPPL è riservato.

Concetti correlati

Nome installazione su AIX, Linux, and Windows

IBM MQNOMI DI FILE

Ogni oggetto gestore code, coda, definizione del processo, elenco nomi, canale, canale di connessione client, listener, servizio e informazioni di autenticazione di IBM MQ è rappresentato da un file. Poiché i nomi oggetto non sono necessariamente nomi file validi, il gestore code converte il nome oggetto in un nome file valido, se necessario.

Il percorso predefinito per una directory del gestore code è il seguente:

- Un prefisso, definito nelle informazioni di configurazione IBM MQ :

– Linux AIX Su AIX and Linux, il prefisso predefinito è `/var/mqm`. Questo è configurato nella stanza `DefaultPrefix` del file di configurazione mqs.ini .

– Windows Su sistemi Windows a 32 bit, il prefisso predefinito è `C:\Program Files (x86)\IBM\WebSphere MQ`. Su sistemi Windows a 64 bit, il prefisso predefinito è `C:\Program Files\IBM\MQ`. Per le installazioni a 32 bit e a 64 bit, le directory di dati sono installate in `C:\ProgramData\IBM\MQ`. Questo è configurato nella stanza `DefaultPrefix` del file di configurazione mqs.ini .

Laddove disponibile, il prefisso può essere modificato utilizzando la pagina delle proprietà IBM MQ in Esplora risorse di IBM MQ , altrimenti modificare manualmente il file di configurazione `mqs.ini` .

- Il nome del gestore code viene trasformato in un nome directory valido. Ad esempio, il gestore code:

```
queue.manager
```

sarà rappresentato come:

```
queue!manager
```

Questo processo viene definito *trasformazione del nome*.

In IBM MQ, è possibile assegnare a un gestore code un nome contenente fino a 48 caratteri.

Ad esempio, è possibile denominare un gestore code:

```
QUEUE.MANAGER.ACCOUNTING.SERVICES
```

Tuttavia, ogni gestore code è rappresentato da un file e vi sono limitazioni sulla lunghezza massima di un nome file e sui caratteri che possono essere utilizzati nel nome. Di conseguenza, i nomi dei file che rappresentano gli oggetti vengono trasformati automaticamente per soddisfare i requisiti del file system.

Le regole che gestiscono la trasformazione di un nome gestore code sono le seguenti:

1. Trasforma singoli caratteri:
   - Da. a!
   - Da / a &
2. Se il nome non è ancora valido:
   a. Tronca a otto caratteri
   b. Accoda un suffisso numerico a tre caratteri

Ad esempio, supponendo il prefisso predefinito e un gestore code con il nome `queue.manager`:

- Windows Su Windows con NTFS o FAT32, il nome gestore code diventa:

```
C:\Program Files\IBM\MQ\qmgrs\queue!manager
```

- Windows Su Windows con FAT, il nome del gestore code diventa:

```
C:\Program Files\IBM\MQ\qmgrs\queue!ma
```

- Linux AIX Su AIX and Linux, il nome gestore code diventa:

```
/var/mqm/qmgrs/queue!manager
```

L'algoritmo di trasformazione distingue anche tra nomi che differiscono solo in maiuscolo / minuscolo su file system che non sono sensibili al maiuscolo / minuscolo.

Trasformazione nome oggetto

I nomi oggetto non sono necessariamente nomi file system validi. Potrebbe essere necessario trasformare i nomi oggetto. Il metodo utilizzato è diverso da quello utilizzato per i nomi dei gestori code poiché, sebbene vi siano solo pochi nomi di gestori code su ciascuna macchina, è possibile che vi sia un numero elevato di altri oggetti per ciascun gestore code. Code, definizioni di processi, elenchi nomi, canali, canali di connessione client, listener, servizi e oggetti delle informazioni di autenticazione sono rappresentati nel file system.

Quando il processo di trasformazione genera un nuovo nome, non esiste una semplice relazione con il nome oggetto originale. È possibile utilizzare il comando **dspmqfls** per eseguire la conversione tra nomi di oggetti reali e trasformati.

Riferimenti correlati

dspmqfls (nomi file di visualizzazione)

Informazioni correlate

Stanza AllQueueManagers del file mqs.ini

IBM i Nomi oggetto su IBM i

Un gestore code ha una libreria del gestore code associata che ha un nome univoco. Potrebbe essere necessario trasformare i nomi dei gestori code e dei nomi oggetto per soddisfare i requisiti dell'IFS (Integrated File System) IBM i Integrated File System .

Quando viene creato un gestore code, IBM MQ associa una libreria del gestore code. A questa libreria del gestore code viene assegnato un nome univoco, non più lungo di 10 caratteri, basato in gran parte sul nome del gestore code definito dall'utente. Sia il gestore code che la libreria del gestore code vengono collocati in una directory che si basa anche sul nome del gestore code con il prefisso `/QIBM/UserData/mqm`. Di seguito è riportato un esempio di gestore code, libreria del gestore code e directory:

Nome del gestore code	ARANCIO
Nome libreria gestore code	QMORANGE
Directory	`/QIBM/UserData/mqm/ORANGE`

Tutti i nomi dei gestori code e i nomi delle librerie dei gestori code vengono scritti nelle stanze nel file `/QIBM/UserData/mqm/mqs.ini`.

IBM MQ File e directory IFS

L'IFS (Integrated File System) IBM i Integrated File System viene utilizzato ampiamente da IBM MQ per memorizzare i dati. Per ulteriori informazioni su IFS, consultare *Integrated File System Introduzione*.

Ogni oggetto IBM MQ , ad esempio un canale o un gestore code, è rappresentato da un file. Poiché i nomi oggetto non sono necessariamente nomi file validi, il gestore code converte il nome oggetto in un nome file valido, se necessario.

Il percorso di una directory del gestore code è formato da:

- Un prefisso, definito nel file di configurazione del gestore code, `qm.ini`. Il prefisso predefinito è `/QIBM/UserData/mqm`.
- Una costante letterale, `qmgrs`.
- Un nome gestore code codificato, che è il nome del gestore code trasformato in un nome directory valido. Ad esempio, il gestore code `queue/manager` è rappresentato da `queue&manager`.

Questo processo viene definito trasformazione del nome.

Trasformazione nome gestore code IFS

In IBM MQ, è possibile assegnare a un gestore code un nome contenente fino a 48 caratteri.

Ad esempio, è possibile denominare un gestore code `QUEUE/MANAGER/ACCOUNTING/SERVICES`. Nello stesso modo in cui viene creata una libreria per ciascun gestore code, ogni gestore code è rappresentato anche da un file. A causa dei punti di codice delle varianti in EBCDIC, esistono delle limitazioni ai caratteri che possono essere utilizzati nel nome. Di conseguenza, i nomi dei file IFS che rappresentano gli oggetti vengono trasformati automaticamente per soddisfare i requisiti del file system.

Utilizzando l'esempio di un gestore code con il nome `queue/manager`, trasformando il carattere `/` in `&e` assumendo il prefisso predefinito, il nome del gestore code in IBM MQ for IBM i diventa `/QIBM/UserData/mqm/qmgrs/queue&manager`.

Trasformazione nome oggetto

I nomi oggetto non sono necessariamente nomi file system validi, quindi i nomi oggetto potrebbero dover essere trasformati. Il metodo utilizzato è diverso da quello per i nomi dei gestori code perché, sebbene vi siano solo pochi nomi di gestori code per ciascuna macchina, è possibile che vi sia un numero elevato di altri oggetti per ciascun gestore code. Solo le definizioni dei processi, le code e gli elenchi nomi sono rappresentati nel file system; i canali non sono influenzati da queste considerazioni.

Quando il processo di trasformazione genera un nuovo nome, non esiste una semplice relazione con il nome oggetto originale. È possibile utilizzare il comando DSPMQMOBJN per visualizzare i nomi trasformati per gli oggetti IBM MQ .

Accodamento distribuito e cluster

L'accodamento distribuito significa inviare messaggi da un gestore code ad un altro. Il gestore code di ricezione può trovarsi sulla stessa macchina o su un'altra; nelle vicinanze o dall'altra parte del mondo. Può essere in esecuzione sulla stessa piattaforma del gestore code locale o su una qualsiasi delle piattaforme supportate da IBM MQ. È possibile definire manualmente tutte le connessioni in un ambiente di accodamento distribuito oppure è possibile creare un cluster e consentire a IBM MQ di definire gran parte dei dettagli di connessione.

accodamento distribuito

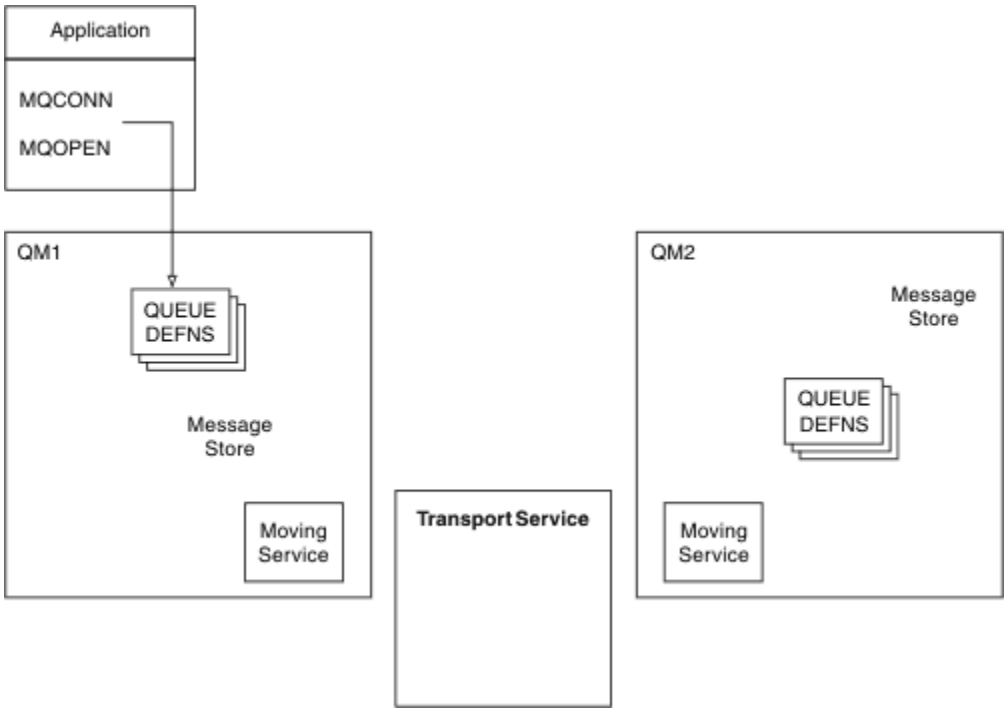


Figura 4. Panoramica dei componenti dell'accodamento distribuito

Nella figura precedente:

- Un'applicazione utilizza la chiamata MQCONN per connettersi ad un gestore code. L'applicazione utilizza quindi la chiamata MQOPEN per aprire una coda in modo che possa inserire i messaggi nella coda.
- Ogni gestore code ha una definizione per ciascuna delle sue code. Può avere definizioni di *code locali* (ossia, ospitate da questo gestore code) e definizioni di *code remote* (ossia, ospitate da altri gestori code).
- Se i messaggi sono destinati a una coda remota, il gestore code locale li conserva su una coda di trasmissione , che li conserva in un archivio messaggi, fino a quando non possono essere inoltrati al gestore code remoto.
- Ogni gestore code contiene software di comunicazioni, noto come *servizio di spostamento*, che il gestore code utilizza per comunicare con altri gestori code.
- Il *servizio di trasporto* è indipendente dal gestore code e può essere uno dei seguenti (a seconda della piattaforma):
  - SNA APPC (Systems Network Architecture Advanced Program - to Program Communication)
  - Transmission Control Protocol/Internet Protocol (TCP/IP)
  - Sistema di input / output di base della rete (NetBIOS)
  - SPX (Sequenced Packet Exchange)

Componenti necessari per inviare un messaggio

Se un messaggio deve essere inviato a un gestore code remoto, il gestore code locale necessita di definizioni per una *coda di trasmissione* e un *canale*. Un canale è un collegamento di comunicazione unidirezionale tra due gestori code. Può trasportare messaggi destinati a qualsiasi numero di code sul gestore code remoto.

Ogni estremità di un canale ha una definizione separata, definendolo, ad esempio, come l'estremità di invio o l'estremità di ricezione. Un canale semplice è costituito da una definizione di canale *mittente* nel gestore code locale e una definizione di canale *destinatario* nel gestore code remoto. Queste due definizioni devono avere lo stesso nome e insieme costituiscono un canale.

Il software che gestisce l'invio e la ricezione di messaggi è denominato *MCA (Message Channel Agent)* . Esiste un MCA (*message channel agent*) ad ogni estremità di un canale.

Ogni gestore code deve avere una *coda di messaggi non recapitabili* (nota anche come *coda di messaggi non recapitati*). I messaggi vengono inseriti in questa coda se non possono essere consegnati alla loro destinazione.

La seguente figura mostra la relazione tra gestori code, code di trasmissione, canali e MCA:

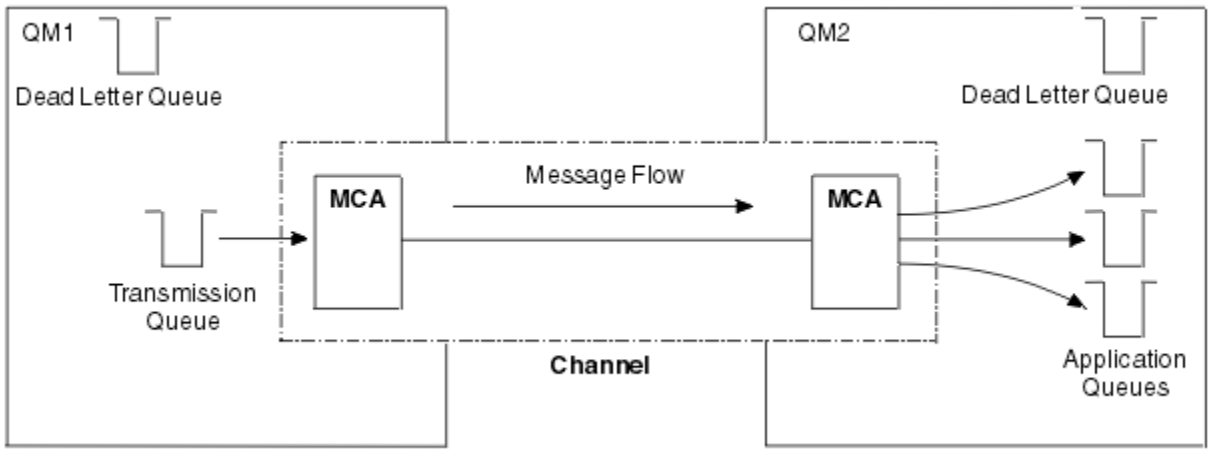


Figura 5. invio di messaggi

Componenti necessari per restituire un messaggio

Se l'applicazione richiede che i messaggi vengano restituiti dal gestore code remoto, è necessario definire un altro canale, da eseguire nella direzione opposta tra i gestori code, come mostrato nella seguente figura:

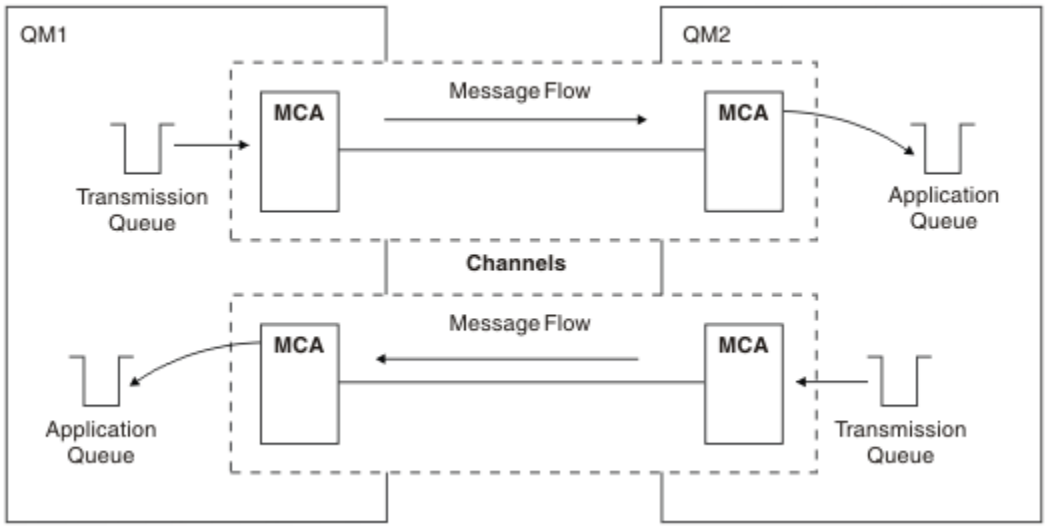


Figura 6. Invio di messaggi in entrambe le direzioni

Cluster

Anziché definire manualmente tutte le connessioni in un ambiente di accodamento distribuito, è possibile raggruppare una serie di gestori code in un cluster. Quando si esegue questa operazione, i gestori code possono rendere le code che ospitano disponibili ad altri gestori code nel cluster senza la necessità di definizioni di canale esplicite, definizioni di coda remota o code di trasmissione per ciascuna destinazione. Ogni gestore code in un cluster ha una singola coda di trasmissione che trasmette i messaggi a qualsiasi altro gestore code nel cluster. Per ogni gestore code, è necessario solo definire un canale ricevente del

cluster e un canale mittente del cluster; eventuali canali aggiuntivi vengono gestiti automaticamente dal cluster.

Un IBM MQ client può connettersi a un gestore code che fa parte di un cluster, proprio come può connettersi a qualsiasi altro gestore code. Come con l'accodamento distribuito configurato manualmente, utilizzare la chiamata MQPUT per inserire un messaggio in una coda su qualsiasi gestore code. Utilizzare la chiamata MQGET per richiamare i messaggi da una coda locale.

I gestori code su piattaforme che supportano cluster non devono necessariamente far parte di un cluster. È possibile continuare a configurare manualmente l'accodamento distribuito e / o utilizzare i cluster.

Vantaggi dell'utilizzo dei cluster

Il clustering fornisce due vantaggi principali:

- I cluster semplificano la gestione delle reti IBM MQ , che di solito richiedono la configurazione di molte definizioni di oggetti per canali, code di trasmissione e code remote. Questa situazione è particolarmente vera nelle reti di grandi dimensioni, potenzialmente in fase di modifica, in cui molti gestori code devono essere interconnessi. Questa architettura è particolarmente difficile da configurare e mantenere attivamente.
- I cluster possono essere utilizzati per distribuire il carico di lavoro del traffico di messaggi tra code e gestori code nel cluster. Tale distribuzione consente la distribuzione del carico di lavoro dei messaggi di una singola coda tra istanze equivalenti di quella coda ubicata su più gestori code. Questa distribuzione del carico di lavoro può essere utilizzata per ottenere una maggiore resilienza agli errori di sistema e per migliorare le prestazioni di scalabilità dei flussi di messaggi particolarmente attivi in un sistema. In un ambiente di questo tipo, ciascuna delle istanze delle code distribuite utilizza le applicazioni che elaborano i messaggi. Per ulteriori informazioni, consultare Utilizzo dei cluster per la gestione del carico di lavoro.

Modalità di instradamento dei messaggi in un cluster

È possibile considerare un cluster come una rete di gestori code gestiti da un amministratore di sistema coscienzioso. Ogni volta che si definisce una coda cluster, l'amministratore di sistema crea automaticamente le definizioni di coda remota corrispondenti, in base alle esigenze, sugli altri gestori code.

Non è necessario creare definizioni di code di trasmissione perché IBM MQ fornisce una coda di trasmissione su ciascun gestore code del cluster. Questa singola coda di trasmissione può essere utilizzata per trasportare i messaggi a qualsiasi altro gestore code nel cluster. Non si è limitati all'utilizzo di una singola coda di trasmissione. Un gestore code può utilizzare più code di trasmissione per separare i messaggi diretti a ciascun gestore code in un cluster. Generalmente, un gestore code utilizza una singola coda di trasmissione cluster. È possibile modificare l'attributo del gestore code DEFCLXQ, in modo che un gestore code utilizzi una diversa coda di trasmissione cluster per ciascun gestore code in un cluster. È anche possibile definire manualmente le code di trasmissione del cluster.

Tutti i gestori code che si uniscono a un cluster accettano di lavorare in questo modo. Inviano informazioni su se stessi e sulle code che ospitano e ricevono informazioni sugli altri membri del cluster.

Per garantire che non vengano perse informazioni quando un gestore code diventa non disponibile, specificare due gestori code nel cluster che fungano da *repository completi*. Questi gestori code memorizzano una serie completa di informazioni su tutti i gestori code e le code nel cluster. Tutti gli altri gestori code nel cluster memorizzano solo le informazioni su tali gestori code e code con cui scambiano messaggi. Questi gestori code sono noti come *repository parziali*. Per ulteriori informazioni, consultare “Repository cluster” a pagina 57.

Per far parte di un cluster, un gestore code deve avere due canali: un canale mittente del cluster e un canale ricevente del cluster:

- Un canale mittente cluster è un canale di comunicazione come un canale mittente. È necessario creare manualmente un canale mittente del cluster su un gestore code per connetterlo a un repository completo già membro del cluster.

- Un canale ricevente cluster è un canale di comunicazione come un canale ricevente. È necessario creare manualmente un canale ricevente del cluster. Il canale funge da meccanismo per il gestore code per ricevere le comunicazioni cluster.

Tutti gli altri canali necessari per la comunicazione tra questo gestore code e gli altri membri del cluster vengono quindi creati automaticamente.

La seguente figura mostra i componenti di un cluster denominato CLUSTER:

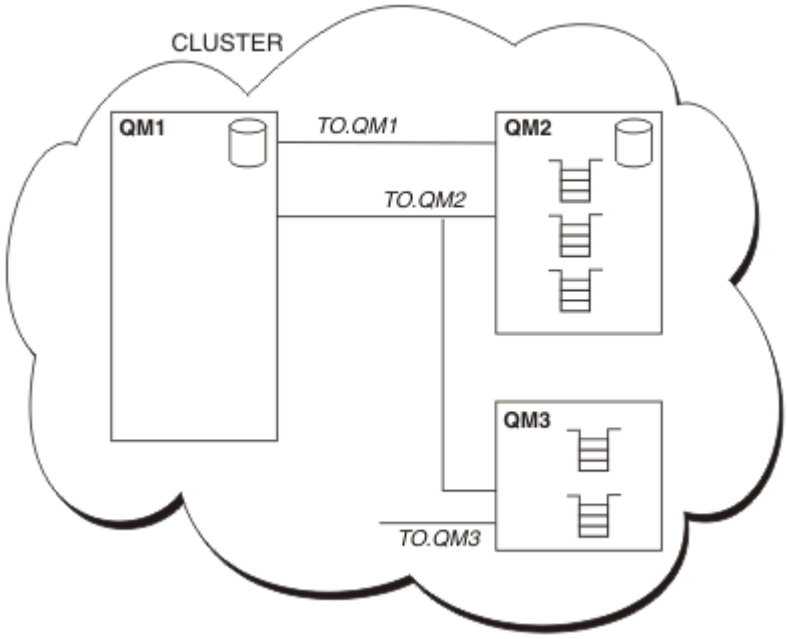


Figura 7. Un cluster di gestori code

- CLUSTER contiene tre gestori code, QM1, QM2e QM3.
- QM1 e QM2 ospitano repository completi di informazioni sui gestori code e sulle code nel cluster.
- QM2 e QM3 ospitano alcune code cluster, ossia code accessibili a qualsiasi altro gestore code nel cluster.
- Ogni gestore code dispone di un canale ricevente del cluster denominato TO.qmgr su cui è possibile ricevere messaggi.
- Ogni gestore code ha anche un canale mittente del cluster su cui può inviare informazioni a uno dei gestori code del repository.
- QM1 e QM3 inviano al repository in QM2 e QM2 invia al repository in QM1.

Componenti di accodamento distribuiti

I componenti dell'accodamento distribuito sono i canali di messaggi, gli agenti dei canali di messaggi, le code di trasmissione, i listener e gli iniziatori dei canali e i programmi di uscita dei canali. La definizione di ciascuna estremità di un canale di messaggi può essere di diversi tipi.

I canali dei messaggi sono i canali che trasportano i messaggi da un gestore code all'altro. Non confondere i canali dei messaggi con quelli MQI. Esistono due tipi di canale MQI, SVRCONN (server - connection) e CLNTCONN (client - connection). Per ulteriori informazioni, vedi Canali.

La definizione di ciascuna estremità di un canale di messaggi può essere uno dei seguenti tipi:

- Mittente (SDR)
- Ricevitore (RCVR)
- Server (SVR)

- Richiedente (RQSTR)
- Mittente cluster (CLUSSDR)
- Ricevitore cluster (CLUSRCVR)

Un canale di messaggi viene definito utilizzando uno di questi tipi definiti ad un'estremità e un tipo compatibile all'altra estremità. Le combinazioni possibili sono:

- Mittente-Ricevente
- Richiedente-Server
- Richiedente-Mittente (callback)
- Server-Ricevente
- Mittente cluster - ricevente cluster

Istruzioni dettagliate per la creazione di un canale mittente - destinatario sono incluse in Definizione dei canali. Per esempi dei parametri necessari per impostare i canali mittente - destinatario, consultare Informazioni di configurazione di esempio applicabili alla propria piattaforma. Per i parametri necessari per definire un canale di qualsiasi tipo, consultare DEFINE CHANNEL.

Canali mittente-ricevente

Un mittente in un sistema avvia il canale in modo che possa inviare messaggi all'altro sistema. Il mittente richiede l'avvio del destinatario all'altra estremità del canale. Il mittente invia i messaggi dalla coda di trasmissione al ricevitore. Il destinatario inserisce i messaggi nella coda di destinazione. Figura 8 a pagina 48 illustra questo.

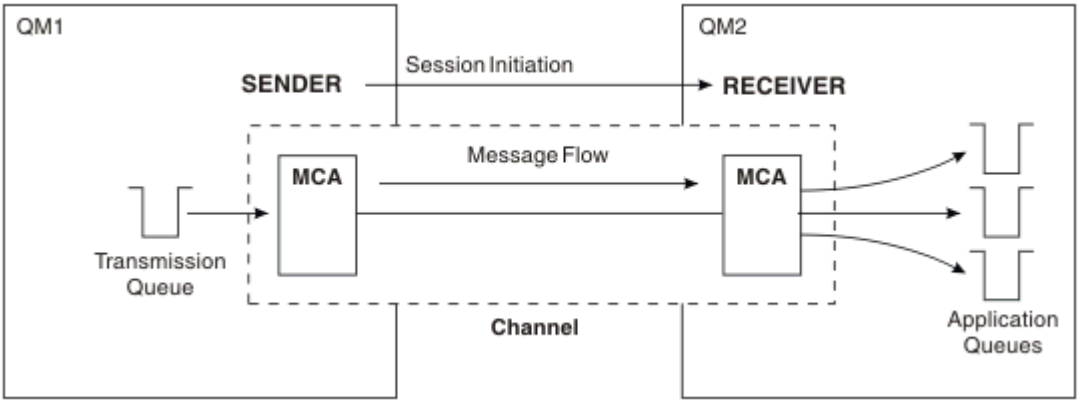


Figura 8. Un canale mittente - destinatario

Canali richiedente-server

Un richiedente in un sistema avvia il canale in modo che possa ricevere messaggi dall'altro sistema. Il richiedente richiede l'avvio del server all'altra estremità del canale. Il server invia i messaggi al richiedente dalla coda di trasmissione definita nella definizione del canale.

Un canale server può anche avviare la comunicazione e inviare messaggi a un richiedente. Ciò si applica solo ai server *completi* , ossia i canali server che hanno il nome connessione del partner specificato nella definizione del canale. Un server completo può essere avviato da un richiedente o può avviare una comunicazione con un richiedente.

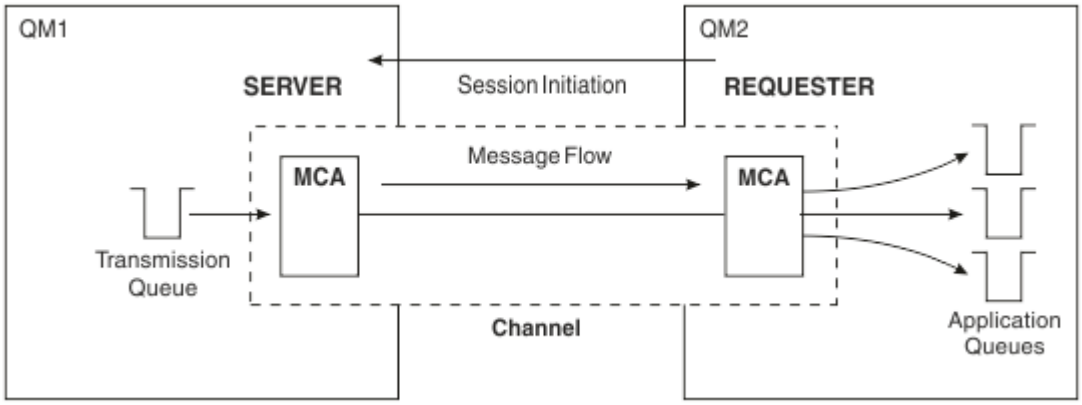


Figura 9. Un canale richiedente - server

Canali richiedente-mittente

Il richiedente avvia il canale e il mittente termina la chiamata. Il mittente riavvia quindi la comunicazione in base alle informazioni nella relativa definizione di canale (nota come *callback*). Invia i messaggi dalla coda di trasmissione al richiedente.

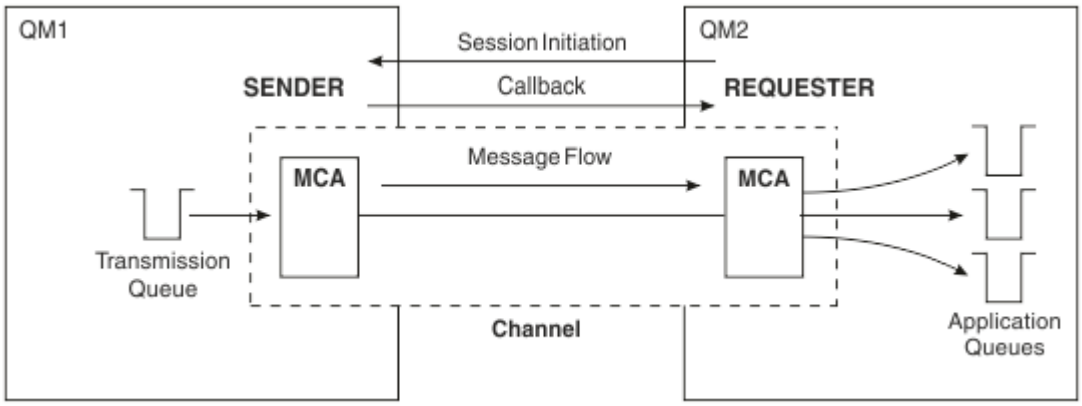


Figura 10. Un canale richiedente - mittente

Canali riceventi del server

È come il mittente - destinatario, ma si applica solo ai server *completi* , ossia i canali server che hanno il nome connessione del partner specificato nella definizione del canale. L'avvio del canale deve essere avviato all'estremità del server del collegamento. L'illustrazione è come l'illustrazione in Figura 8 a pagina 48.

Canali mittenti del cluster

In un cluster, ogni gestore code dispone di un canale mittente del cluster su cui è possibile inviare le informazioni del cluster a uno dei gestori code del repository completo. I gestori code possono anche inviare messaggi ad altri gestori code sui canali mittenti del cluster.

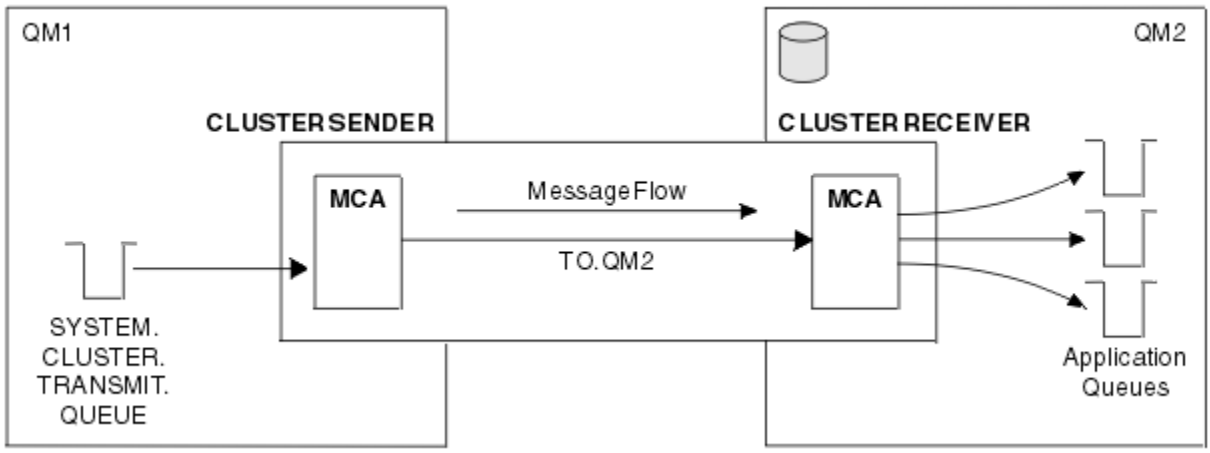


Figura 11. Un canale mittente del cluster

Canali riceventi del cluster

In un cluster, ogni gestore code ha un canale ricevente del cluster su cui può ricevere messaggi e informazioni sul cluster. L'illustrazione è come l'illustrazione in Figura 11 a pagina 50.

Code di messaggi non recapitabili

La coda di messaggi non recapitabili (o coda di messaggi non recapitati) è la coda a cui vengono inviati i messaggi se non possono essere instradati alla destinazione corretta. Ogni gestore code generalmente ha una coda di messaggi non recapitabili.

Una *coda di messaggi non recapitabili* (DLQ), a volte indicata come *coda di messaggi non recapitati*, è una coda di attesa per i messaggi che non possono essere consegnati alle relative code di destinazione, ad esempio perché la coda non esiste o perché è piena. Le code di messaggi non recapitabili vengono utilizzate anche all'estremità di invio di un canale, per errori di conversione dati. Ogni gestore code in una rete generalmente ha una coda locale da utilizzare come coda di messaggi non recapitabili in modo che i messaggi che non possono essere consegnati alla destinazione corretta possano essere memorizzati per un successivo richiamo.

I messaggi possono essere inseriti nella DLQ da gestori code, MCA (message channel agent) e applicazioni. Tutti i messaggi sulla DLQ devono avere come prefisso una struttura *dead-letter header* , MQDLH. Il campo *Motivo* della struttura MQDLH contiene un codice motivo che identifica il motivo per cui il messaggio si trova sulla DLQ.

Generalmente, è necessario definire una coda di messaggi non instradabili per ogni gestore code. In caso contrario, e l'MCA non è in grado di inserire un messaggio, questo viene lasciato sulla coda di trasmissione e il canale viene arrestato. Inoltre, se i messaggi sono rapidi, non persistenti (consultare Messaggi rapidi, non persistenti) non può essere consegnato e non esiste alcuna coda di messaggi non instradabili sul sistema di destinazione, questi messaggi vengono eliminati.

Tuttavia, l'utilizzo di code di messaggi non recapitabili può influenzare la sequenza in cui i messaggi vengono consegnati, pertanto è possibile scegliere di non utilizzarle.

Attività correlate

Gestione delle code di messaggi non recapitabili

Risoluzione dei problemi dei messaggi non recapitati

Riferimenti correlati

runmqdlq (esecuzione gestore code di messaggi non instradabili)

Definizioni di coda remota

Le definizioni di coda remota sono definizioni per le code di proprietà di un altro gestore code.

Mentre le applicazioni possono richiamare i messaggi solo dalle code locali, possono inserire i messaggi sulle code locali o remote. Pertanto, oltre a una definizione per ognuna delle proprie code locali, un gestore code può avere *definizioni di code remote*. Il vantaggio delle definizioni della coda remota è che consentono a un'applicazione di inserire un messaggio in una coda remota senza dover specificare il nome della coda remota o del gestore code remoto o il nome della coda di trasmissione. Le definizioni della coda remota forniscono l'indipendenza dell'ubicazione.

Esistono altri usi per le definizioni di coda remota, descritti in seguito.

Come accedere al gestore code remoto

È possibile che non si disponga sempre di un canale tra ciascun gestore code di origine e di destinazione. Ci sono una serie di altri modi di collegamento tra i due, tra cui multi - hopping, canali di condivisione, utilizzando canali diversi e il clustering.

Multi - hopping

Se non vi è alcun collegamento di comunicazione diretto tra il gestore code di origine e il gestore code di destinazione, è possibile passare attraverso uno o più *gestori code intermedi* nel percorso verso il gestore code di destinazione. Questo è noto come *multi - hop*.

È necessario definire i canali tra tutti i gestori code e le code di trasmissione sui gestori code intermedi. Viene mostrato in Figura 12 a pagina 51.

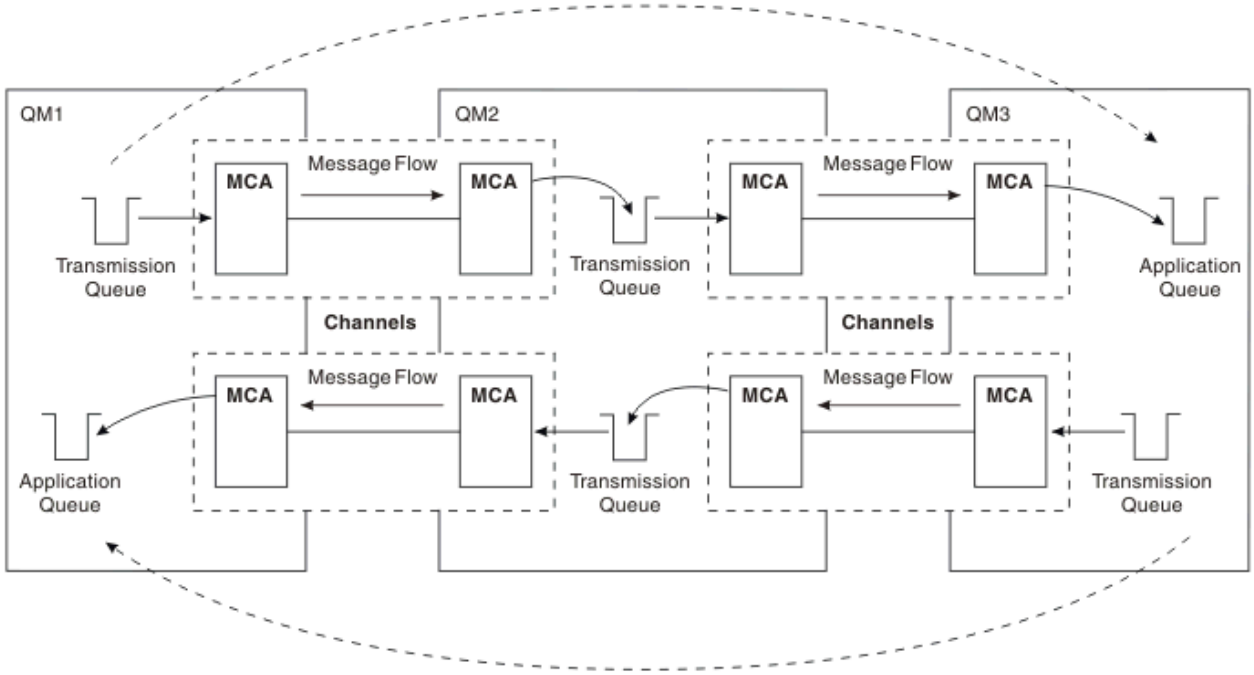


Figura 12. Passaggio attraverso gestori code intermedi

Condivisione dei canali

In qualità di progettista dell'applicazione, è possibile forzare le applicazioni a specificare il nome del gestore code remoto insieme al nome della coda oppure creare una *definizione della coda remota* per ogni coda remota. Questa definizione contiene il nome del gestore code remoto, il nome della coda e il nome della coda di trasmissione. In entrambi i casi, tutti i messaggi di tutte le applicazioni che indirizzano le code nella stessa ubicazione remota vengono inviati attraverso la stessa coda di trasmissione. Viene mostrato in Figura 13 a pagina 52.

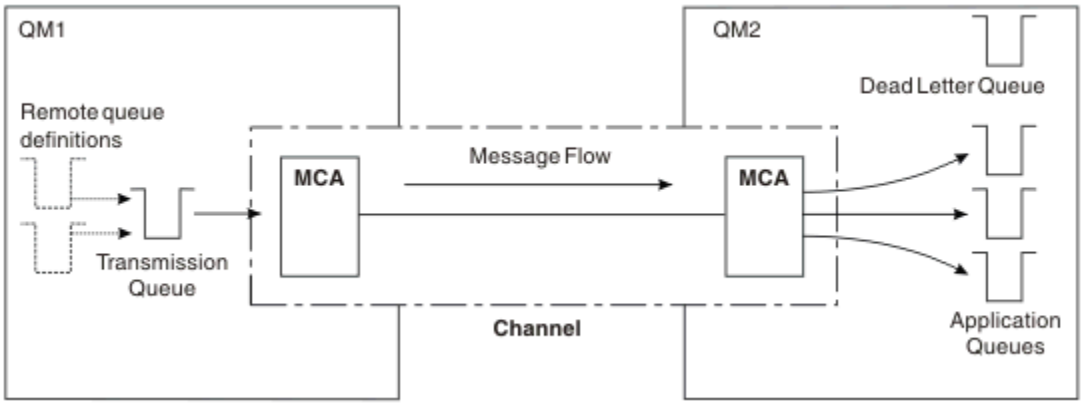


Figura 13. Condivisione di una coda di trasmissione

Figura 13 a pagina 52 illustra che i messaggi da più applicazioni a più code remote possono utilizzare lo stesso canale.

Utilizzo di canali diversi

Se si dispone di messaggi di tipo diverso da inviare tra due gestori code, è possibile definire più di un canale tra i due. Ci sono momenti in cui hai bisogno di canali alternativi, magari per motivi di sicurezza, o per scambiare la velocità di consegna contro la maggior parte del traffico di messaggi.

Per impostare un secondo canale, è necessario definire un altro canale e un'altra coda di trasmissione e creare una definizione della coda remota specificando l'ubicazione e il nome della coda di trasmissione. Le applicazioni possono quindi utilizzare entrambi i canali, ma i messaggi vengono ancora consegnati alle stesse code di destinazione. Viene mostrato in Figura 14 a pagina 52.

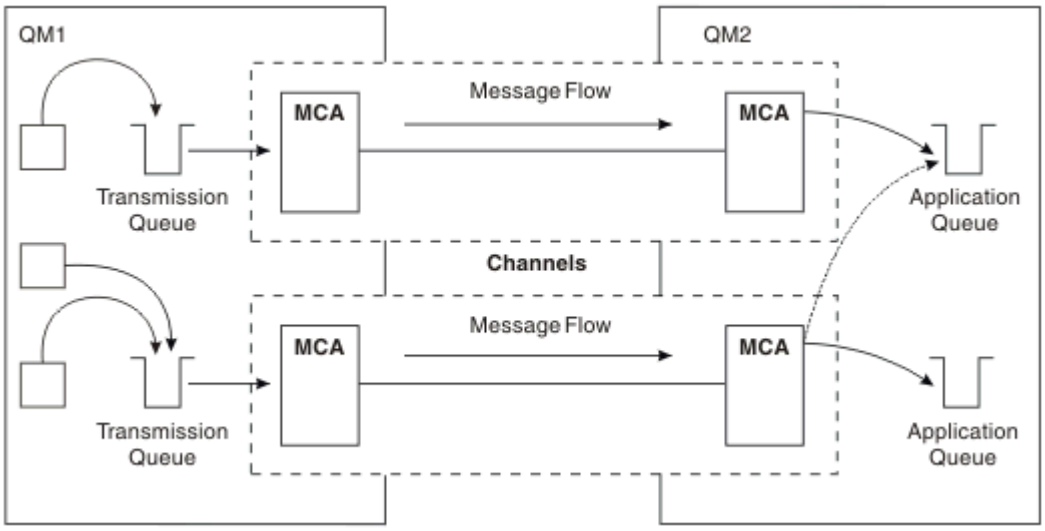


Figura 14. Utilizzo di più canali

Quando si utilizzano le definizioni della coda remota per specificare una coda di trasmissione, le applicazioni **non** devono specificare l'ubicazione (ossia, il gestore code di destinazione). In tal caso, il gestore code non utilizza le definizioni della coda remota. Le definizioni della coda remota forniscono l'indipendenza dell'ubicazione. Le applicazioni possono inserire messaggi in una coda *logica* senza sapere dove si trova la coda ed è possibile modificare la coda *fisica* senza dover modificare le proprie applicazioni.

Utilizzo del clustering

Ogni gestore code all'interno di un cluster definisce un canale ricevente del cluster. Quando un altro gestore code desidera inviare un messaggio a tale gestore code, definisce automaticamente il canale mittente del cluster corrispondente. Ad esempio, se c'è più di un'istanza di una coda in un cluster, il canale mittente del cluster potrebbe essere definito per qualsiasi gestore code che ospita la coda. IBM MQ utilizza un algoritmo di gestione del carico di lavoro che utilizza una routine round robin per selezionare un gestore code disponibile a cui instradare un messaggio. Per ulteriori informazioni, consultare Cluster.

Informazioni sull'indirizzamento

Quando un'applicazione inserisce messaggi destinati a un gestore code remoto, il gestore code locale aggiunge loro un'intestazione di trasmissione prima di inserirli nella coda di trasmissione. Questa intestazione contiene il nome della coda di destinazione e del gestore code, ovvero le *informazioni di indirizzamento*.

In un singolo ambiente del gestore code, l'indirizzo di una coda di destinazione viene stabilito quando un'applicazione apre una coda per l'inserimento dei messaggi. Poiché la coda di destinazione si trova sullo stesso gestore code, non è necessaria alcuna informazione di indirizzamento.

In un ambiente di accodamento distribuito, il gestore code deve conoscere non solo il nome della coda di destinazione, ma anche l'ubicazione di tale coda (ossia, il nome del gestore code) e l'instradamento a tale ubicazione remota (ossia, la coda di trasmissione). Queste informazioni di indirizzamento sono contenute nell'intestazione di trasmissione. Il canale di ricezione rimuove l'intestazione di trasmissione e utilizza le informazioni in essa contenute per individuare la coda di destinazione.

È possibile evitare che le applicazioni debbano specificare il nome del gestore code di destinazione se si utilizza una definizione di coda remota. Questa definizione specifica il nome della coda remota, il nome del gestore code remoto a cui sono destinati i messaggi e il nome della coda di trasmissione utilizzata per trasportare i messaggi.

Cosa sono gli alias?

Gli alias vengono utilizzati per fornire una qualità del servizio per i messaggi. L'alias del gestore code consente all'amministratore di sistema di modificare il nome di un gestore code di destinazione senza dover modificare le applicazioni. Inoltre, consente all'amministratore di sistema di modificare l'instradamento a un gestore code di destinazione o di impostare un instradamento che implica il passaggio attraverso un numero di altri gestori code (multi - hopping). L'alias della coda di risposta fornisce una qualità del servizio per le risposte.

Gli alias del gestore code e gli alias della coda di risposta vengono creati utilizzando una definizione di coda remota con un RNAME vuoto. Queste definizioni non definiscono code reali; vengono utilizzate dal gestore code per risolvere i nomi delle code fisiche, i nomi dei gestori code e le code di trasmissione.

Le definizioni alias sono caratterizzate da un RNAME vuoto.

Risoluzione nome coda

La risoluzione del nome della coda si verifica in ogni gestore code ogni volta che una coda viene aperta. Il suo scopo è identificare la coda di destinazione, il gestore code di destinazione (che potrebbe essere locale) e l'instradamento a tale gestore code (che potrebbe essere null). Il nome risolto è composto da tre parti: il nome del gestore code, il nome della coda e, se il gestore code è remoto, la coda di trasmissione.

Quando esiste una definizione di coda remota, non si fa riferimento a nessuna definizione di alias. Il nome coda fornito dall'applicazione viene risolto nel nome della coda di destinazione, del gestore code remoto e della coda di trasmissione specificati nella definizione della coda remota. Per informazioni più dettagliate sulla risoluzione dei nomi delle code, consultare Risoluzione dei nomi delle code.

Se non è presente alcuna definizione di coda remota e viene specificato un nome gestore code o viene risolto dal servizio dei nomi, il gestore code controlla se esiste una definizione alias del gestore code che corrisponde al nome del gestore code fornito. In questo caso, le informazioni vengono utilizzate per

risolvere il nome del gestore code nel nome del gestore code di destinazione. La definizione alias del gestore code può essere utilizzata anche per determinare la coda di trasmissione al gestore code di destinazione.

Se il nome della coda risolta non è una coda locale, sia il nome del gestore code che il nome della coda sono inclusi nell'intestazione di trasmissione di ogni messaggio inserito dall'applicazione alla coda di trasmissione.

La coda di trasmissione utilizzata in genere ha lo stesso nome del gestore code risolto, a meno che non venga modificata da una definizione di coda remota o da una definizione di alias del gestore code. Se non è stata definita una coda di trasmissione di questo tipo ma è stata definita una coda di trasmissione predefinita, viene utilizzata.

z/OS I nomi dei gestori code in esecuzione su z/OS sono limitati a quattro caratteri.

Definizioni alias gestore code

Le definizioni degli alias del gestore code si applicano quando un'applicazione che apre una coda per inserire un messaggio, specifica il nome della coda **e** il nome del gestore code.

Le definizioni alias del gestore code hanno tre utilizzi:

- Quando si inviano messaggi, riassociare il nome del gestore code
- Durante l'invio di messaggi, la modifica o la specifica della coda di trasmissione
- Quando si ricevono i messaggi, determinando se il gestore code locale è la destinazione prevista per tali messaggi

Messaggi in uscita - rimappatura del nome gestore code

Le definizioni di alias del gestore code possono essere utilizzate per riassociare il nome gestore code specificato in una chiamata MQOPEN. Ad esempio, una chiamata MQOPEN specifica un nome coda THISQ e un nome gestore code YOURQM. Nel gestore code locale, è presente una definizione alias del gestore code simile al seguente esempio:

```
DEFINE QREMOTE (YOURQM) RQMNAME(REALQM)
```

Mostra che il gestore code reale da utilizzare, quando un'applicazione inserisce i messaggi nel gestore code YOURQM, è REALQM. Se il gestore code locale è REALQM, inserisce i messaggi nella coda THISQ, che è una coda locale. Se il gestore code locale non è denominato REALQM, instrada il messaggio ad una coda di trasmissione denominata REALQM. Il gestore code modifica l'intestazione di trasmissione in REALQM invece di YOURQM.

Messaggi in uscita - modifica o specifica della coda di trasmissione

La Figura 15 a pagina 55 mostra uno scenario in cui i messaggi arrivano al gestore code QM1 con intestazioni di trasmissione che mostrano i nomi delle code sul gestore code QM3. In questo scenario, QM3 è raggiungibile passando attraverso QM2.

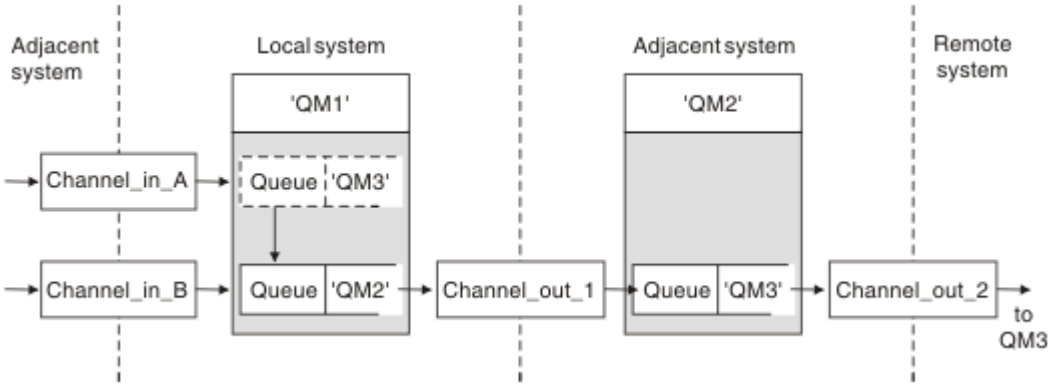


Figura 15. Alias gestore code

Tutti i messaggi per QM3 vengono acquisiti in QM1 con un alias del gestore code. L'alias del gestore code è denominato QM3 e contiene la definizione QM3 tramite la coda di trasmissione QM2. La definizione è simile al seguente esempio:

```
DEFINE QREMOTE (QM3) RNAME(' ') RQMNAME(QM3) XMITQ(QM2)
```

Il gestore code inserisce i messaggi nella coda di trasmissione QM2 ma non modifica l'intestazione della coda di trasmissione perché il nome del gestore code di destinazione, QM3, non cambia.

Anche tutti i messaggi che arrivano a QM1 e che mostrano un'intestazione di trasmissione contenente un nome coda in QM2 vengono inseriti sulla coda trasmissione QM2 . In questo modo, i messaggi con destinazioni differenti vengono raccolti su una coda di trasmissione comune a un sistema adiacente appropriato, per la trasmissione in avanti alle relative destinazioni.

Messaggi in ingresso - determinazione della destinazione

Un MCA di ricezione apre la coda a cui si fa riferimento nell'intestazione di trasmissione. Se esiste una definizione di alias del gestore code con lo stesso nome del gestore code a cui si fa riferimento, il nome del gestore code ricevuto nell'intestazione di trasmissione viene sostituito con RQMNAME da tale definizione.

Questo processo ha due usi:

- Indirizzamento di messaggi a un altro gestore code
- Modifica del nome del gestore code in modo che sia uguale al gestore code locale

Definizioni alias coda di risposta

Una definizione dell'alias della coda di risposta specifica nomi alternativi per le informazioni di risposta nel descrizione del messaggio. Il vantaggio è che è possibile modificare il nome di una coda o di un gestore code senza dover modificare le applicazioni.

Risoluzione nome coda

Quando un'applicazione risponde a un messaggio, utilizza i dati nel *descrittore del messaggio* del messaggio ricevuto per individuare il nome della coda a cui rispondere. L'applicazione di invio indica dove vengono inviate le risposte e allega queste informazioni ai relativi messaggi. Questo concetto deve essere coordinato come parte della progettazione dell'applicazione.

La risoluzione del nome della coda avviene all'estremità di invio dell'applicazione, prima che il messaggio venga inserito in una coda. La risoluzione del nome della coda si verifica quindi prima dell'interazione con l'applicazione remota a cui viene inviato il messaggio. Questa è l'unica situazione in cui la risoluzione dei nomi si verifica in un momento in cui una coda non viene aperta.

Risoluzione del nome coda utilizzando un alias del gestore code

Normalmente un'applicazione specifica una coda di risposta e lascia vuoto il nome del gestore code di risposta. Il gestore code completa il proprio nome al momento dell'inserimento. Questo metodo funziona bene tranne quando si desidera utilizzare un canale alternativo per le repliche, ad esempio, un canale che utilizzi la coda di trasmissione `QM1_relief` invece del canale di ritorno predefinito che utilizza la coda di trasmissione `QM1`. In questa situazione, i nomi dei gestore code specificati nelle intestazioni delle code di trasmissione non corrispondono ai nomi dei gestori code "reali", ma vengono rispecificati utilizzando le definizioni di alias dei gestori code. Per restituire le risposte lungo instradamenti alternativi, è necessario associare anche i dati della coda reply - to, utilizzando le definizioni degli alias della coda reply - to.

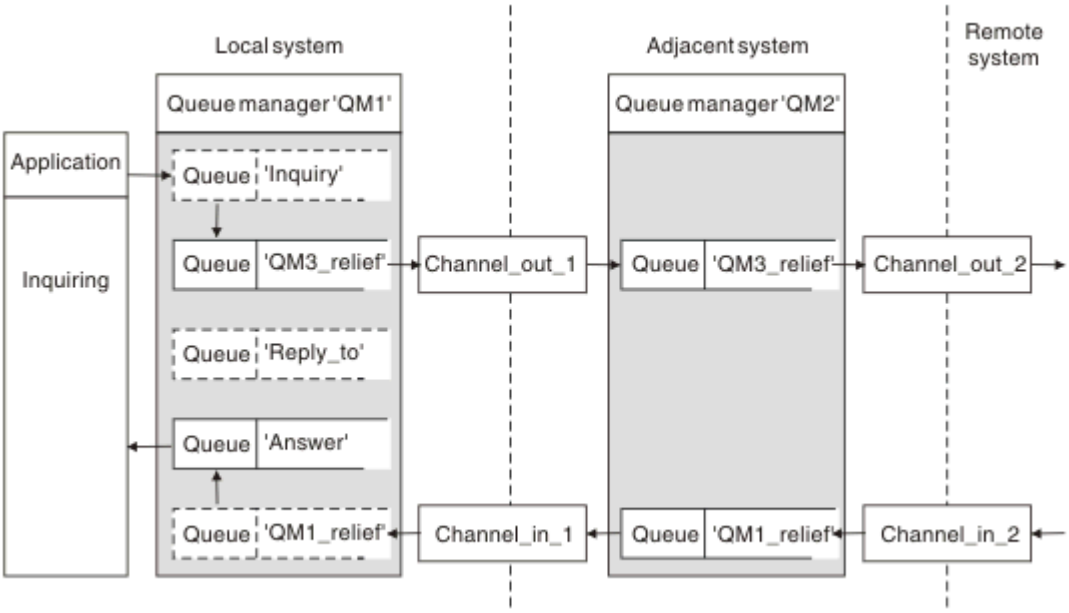


Figura 16. Alias della coda di risposta utilizzato per modificare l'ubicazione della risposta

Nell'esempio in Figura 16 a pagina 56:

1. L'applicazione inserisce un messaggio utilizzando la chiamata MQPUT e specificando le seguenti informazioni nel descrittore del messaggio:

   ```
   ReplyToQ='Reply_to'
   ReplyToQMgr=' '
   ```

   ReplyToIl gestore code deve essere vuoto per poter utilizzare l'alias della coda di risposta.
2. Si crea una definizione alias della coda di risposta denominata `Reply_to`, che contiene il nome `Answere` il nome gestore code `QM1_relief`.

   ```
   DEFINE QREMOTE ('Reply_to') RNAME ('Answer')
   RQMNAME ('QM1_relief')
   ```
3. I messaggi vengono inviati con un descrittore di messaggi che mostra `ReplyToQ='Answer'` e `ReplyToQMgr='QM1_relief'`.
4. La specificazione dell'applicazione deve includere le informazioni che le risposte devono trovare nella coda `Answer` piuttosto che `Reply_to`.

Per prepararsi per le risposte è necessario creare il canale di ritorno parallelo, definendo:

- Su QM2, la coda di trasmissione denominata QM1_relief

```
DEFINE QLOCAL ('QM1_relief') USAGE(XMITQ)
```

- In QM1, l'alias del gestore code QM1_relief

```
DEFINE QREMOTE ('QM1_relief') RNAME() RQMNAME(QM1)
```

  Questo alias del gestore code termina la catena di canali di ritorno paralleli e cattura i messaggi per QM1.

Se si pensa di voler eseguire questa operazione in futuro, assicurarsi che le applicazioni utilizzino il nome alias dall'inizio. Per ora si tratta di un alias di coda normale per la coda di risposta, ma successivamente può essere modificato in un alias del gestore code.

Nome delle repliche alla coda

È necessario prestare attenzione alla denominazione delle code di risposta. Il motivo per cui un'applicazione inserisce un nome coda di risposta nel messaggio è che può specificare la coda a cui vengono inviate le proprie risposte. Quando si crea una definizione dell'alias della coda di risposta con questo nome, non è possibile avere la coda di risposta effettiva (ovvero, una definizione della coda locale) con lo stesso nome. Pertanto, la definizione dell'alias della coda reply - to deve contenere un nuovo nome coda e il nome del gestore code e la specifica dell'applicazione deve includere le informazioni che le sue risposte vengono trovate in questa altra coda.

Le applicazioni ora devono richiamare i messaggi da una coda diversa da quella indicata come coda di risposta quando inseriscono il messaggio originale.

Componenti cluster

I cluster sono composti da gestore code, repository di cluster, canali cluster e code cluster.

Consultare i seguenti argomenti secondari per informazioni su ciascuno dei componenti cluster:

Concetti correlati

Confronto tra cluster e accodamento distribuito

Attività correlate

Configurazione di un cluster di gestore code

Configurazione di un nuovo cluster

Repository cluster

Un repository è una raccolta di informazioni sui gestori code che sono membri di un cluster.

Le informazioni sul repository includono i nomi dei gestori code, le loro ubicazioni, i loro canali, quali code ospitano e altre informazioni. Le informazioni vengono memorizzate sotto forma di messaggi su una coda denominata SYSTEM.CLUSTER.REPOSITORY.QUEUE. Questa coda è uno degli oggetti predefiniti. Multi Su Multiplatforms, viene definito quando viene creato un gestore code IBM MQ . z/OS Su IBM MQ for z/OS, viene definito come parte della personalizzazione del gestore code.

Repository completo e repository parziale

In genere, due gestori code in un cluster contengono un repository completo. Tutti i rimanenti gestori code contengono un repository parziale.

Un gestore code che ospita una serie completa di informazioni su ogni gestore code nel cluster ha un repository completo. Altri gestori code nel cluster dispongono di repository parziali contenenti un sottoinsieme di informazioni nei repository completi.

Un repository parziale contiene informazioni solo sui gestori code con cui il gestore code deve scambiare messaggi. I gestori code richiedono aggiornamenti alle informazioni di cui hanno bisogno, in modo che, in caso di modifiche, il gestore code del repository completo invii loro le nuove informazioni. Per gran parte del tempo, un repository parziale contiene tutte le informazioni che un gestore code deve eseguire all'interno del cluster. Quando un gestore code richiede ulteriori informazioni, interroga il repository completo e aggiorna quindi il repository parziale. I gestori code utilizzano la coda SYSTEM.CLUSTER.COMMAND.QUEUE per richiedere e ricevere aggiornamenti ai repository.

Quando si migra i gestori code che sono membri di un cluster, migrare i repository completi prima dei repository parziali. Ciò è dovuto al fatto che un repository meno recente non può memorizzare attributi più recenti introdotti in una versione più recente. Li tollera, ma non li conserva.

Gestore code del cluster

Un gestore code cluster è un gestore code membro di un cluster.

Un gestore code può essere un membro di più di un cluster. Ogni gestore code cluster deve avere un nome univoco in tutti i cluster di cui è membro.

Un gestore code del cluster può ospitare code, che vengono pubblicizzate agli altri gestori code del cluster. Tuttavia, non è necessario che lo faccia. Può invece alimentare i messaggi nelle code ospitate altrove nel cluster e ricevere solo le risposte che sono indirizzate esplicitamente ad essa.

z/OS In IBM MQ for z/OS, un gestore code del cluster può essere membro di un gruppo di condivisione code. In questo caso, condivide le definizioni di coda con altri gestori code nello stesso gruppo di condivisione code.

I gestori code del cluster sono autonomi. Hanno il controllo completo delle code e dei canali che definiscono. Le loro definizioni non possono essere modificate da altri gestori code (diversi dai gestori code nello stesso gruppo di condivisione code). I gestori code del repository non controllano le definizioni in altri gestori code nel cluster. Essi contengono una serie completa di tutte le definizioni, da utilizzare quando richiesto. Un cluster è una federazione di gestori code.

Dopo aver creato o modificato una definizione su un gestore code del cluster, le informazioni vengono inviate al gestore code del repository completo. Gli altri repository nel cluster vengono aggiornati successivamente.

Gestore code con repository completo

Un gestore code del repository completo è un gestore code del cluster che contiene una rappresentazione completa delle risorse del cluster. Per garantire la disponibilità, impostare due o più gestori code del repository completo in ciascun cluster. I gestori code del repository completo ricevono le informazioni inviate dagli altri gestori code nel cluster e aggiornano i relativi repository. Inviano messaggi l'uno all'altro per essere certi di essere entrambi aggiornati con nuove informazioni sul cluster.

Gestori code e repository

Ogni cluster ha almeno un gestore code (preferibilmente due) che contiene repository completi di informazioni sui gestori code, le code e i canali in un cluster. Questi repository contengono anche richieste dagli altri gestori code nel cluster per aggiornamenti alle informazioni.

Gli altri gestori code contengono un repository parziale, contenente informazioni sul sottoinsieme di code e gestori code con cui devono comunicare. I gestori code creano i propri repository parziali effettuando richieste quando devono accedere per la prima volta a un'altra coda o a un altro gestore code. Richiedono la notifica di eventuali nuove informazioni relative a tale coda o gestore code.

Ciascun gestore code memorizza le proprie informazioni sul repository in messaggi su una coda denominata SYSTEM.CLUSTER.REPOSITORY.QUEUE. I gestori code scambiano le informazioni del repository nei messaggi su una coda denominata SYSTEM.CLUSTER.COMMAND.QUEUE.

Ogni gestore code che unisce un cluster definisce un canale mittente del cluster, CLUSSDR, in uno dei repository. Apprende immediatamente quali altri gestori code nel cluster contengono repository completi.

Da quel momento in poi, il gestore code può richiedere informazioni da uno qualsiasi dei repository. Quando il gestore code invia le informazioni al repository scelto, invia anche le informazioni a un altro repository (se presente).

Un repository completo viene aggiornato quando il gestore code che lo ospita riceve nuove informazioni da uno dei gestori code ad esso collegati. Le nuove informazioni vengono inviate anche a un altro repository, per ridurre il rischio che vengano ritardate se un gestore code del repository è fuori servizio. Poiché tutte le informazioni vengono inviate due volte, i repository devono eliminare i duplicati. Ogni elemento di informazioni contiene un numero di sequenza, che i repository utilizzano per identificare i duplicati. Tutti i repository vengono mantenuti in linea tra loro scambiando messaggi.

Code cluster

Una coda cluster è una coda ospitata da un gestore code cluster e resa disponibile ad altri gestori code del cluster.

Una definizione di coda cluster viene pubblicizzata in altri gestori code nel cluster. Gli altri gestori code nel cluster possono inserire i messaggi in una coda cluster senza la necessità di una definizione di coda remota corrispondente. Una coda cluster può essere pubblicizzata in più di un cluster utilizzando un elenco dei nomi di cluster.

Quando una coda viene pubblicizzata, qualsiasi gestore code del cluster può inserire dei messaggi al suo interno. Per inserire un messaggio, il gestore code deve scoprire, dai repository completi, la posizione in cui è ospitata la coda. Aggiunge quindi alcune informazioni di instradamento al messaggio e inserisce tale messaggio su una coda di trasmissione del cluster.

z/OS Una coda del cluster può essere una coda condivisa dai membri di un gruppo di condivisione di code in IBM MQ for z/OS.

Attività correlate

Definizione di code cluster

Confronto tra code condivise e code cluster

Queste informazioni sono progettate per confrontare le code condivise e le code cluster e decidere quali potrebbero essere più adatte al proprio sistema.

Costi iniziatore canale

Nelle code cluster, i messaggi vengono inviati dai canali, in modo da consentire i costi dell'iniziatore del canale oltre ai costi dell'applicazione. Ci sono dei costi nella rete perché i canali ricevono e inseriscono messaggi. Questi costi non sono presenti con le code condivise, che pertanto utilizzano una minore potenza di elaborazione rispetto alle code cluster quando si spostano i messaggi tra gestori code in un gruppo di condivisione code.

Disponibilità dei messaggi

Durante l'inserimento in una coda, le code cluster inviano il messaggio a un gestore code con canali attivi connessi al gestore code. Sul gestore code remoto, se le applicazioni utilizzate per elaborare i messaggi non funzionano, i messaggi non vengono elaborati e attendono l'avvio delle applicazioni. Allo stesso modo, se un gestore code viene arrestato, i messaggi sul gestore code non vengono resi disponibili fino al riavvio del gestore code. Queste istanze mostrano una disponibilità di messaggi inferiore rispetto all'utilizzo di code condivise.

Quando si utilizzano le code condivise, qualsiasi applicazione nel gruppo di condivisione code può ricevere i messaggi inviati. Se si arresta un gestore code nel gruppo di condivisione code, i messaggi sono disponibili per gli altri gestori code, fornendo una maggiore disponibilità di messaggi rispetto a quando si utilizzano le code cluster.

Capacità

Una CF (Coupling Facility) è più costosa di un disco; di conseguenza, il costo di memorizzazione di 1.000.000 di messaggi in una coda locale è inferiore a quello di una CF con capacità sufficiente per memorizzare lo stesso numero di messaggi.

Invio ad altri gestori code

I messaggi della coda condivisa sono disponibili solo all'interno di un gruppo di condivisione code. Se si desidera utilizzare un gestore code esterno al gruppo di condivisione code, è necessario utilizzare i canali. È possibile utilizzare il clustering per bilanciare il carico di lavoro tra più gestori code distribuiti remoti.

Bilanciamento dei carichi di lavoro

È possibile utilizzare il clustering per dare peso a quali canali e gestori code ricevono una parte dei messaggi inviati. Ad esempio, è possibile inviare il 60% dei messaggi a un gestore code e il 40% dei messaggi a un altro gestore code. Questa istanza non dipende dalla capacità del gestore code remoto di elaborare il lavoro. Il sistema con il primo gestore code potrebbe essere sovraccarico e il sistema con il secondo gestore code potrebbe essere inattivo, ma la maggior parte dei messaggi viene ancora indirizzata al primo gestore code.

Con le code condivise, due sistemi CICS possono ricevere messaggi. Se un sistema è sovraccarico, l'altro sistema assume la maggior parte del carico di lavoro.

Canali cluster

Su ogni repository completo, si definisce manualmente un canale ricevente del cluster e una serie di canali mittenti del cluster per connettersi a ogni altro repository completo nel cluster. Quando si aggiunge un repository parziale, definire manualmente un canale ricevente del cluster e un singolo canale mittente del cluster che si connette a uno dei repository completi. Ulteriori canali mittente del cluster vengono definiti automaticamente dal cluster quando necessario. I canali mittenti del cluster definiti automaticamente prendono i loro attributi dalla corrispondente definizione di canale ricevente del cluster sul gestore code di ricezione.

Canale ricevente del cluster: CLUSRCVR

Una definizione di canale CLUSRCVR definisce la fine di un canale su cui un gestore code cluster può ricevere messaggi da altri gestori code nel cluster.

È necessario definire almeno un canale CLUSRCVR per ogni gestore code cluster. Definendo il canale CLUSRCVR , il gestore code mostra agli altri gestori code del cluster che è disponibile per ricevere messaggi.

Una definizione di canale CLUSRCVR consente inoltre ad altri gestori code di definire automaticamente le corrispondenti definizioni di canale mittente del cluster. Consultare la sezione “Canali mittenti del cluster definiti automaticamente” a pagina 61 di questo articolo.

Canale mittente del cluster: CLUSSDR

Definire manualmente il canale CLUSSDR da ogni gestore code del repository completo a ogni altro gestore code del repository completo nel cluster. Tutti gli aggiornamenti scambiati dai repository completi fluiscono esclusivamente su questi canali. Definendo manualmente questi canali, si controlla esplicitamente la rete di repository completi.

Quando si aggiunge un gestore code del repository parziale a un cluster, si definisce manualmente un singolo canale CLUSSDR per connettersi ad uno dei repository completi. Fa poca differenza quale repository completo si sceglie, perché una volta stabilito il contatto iniziale, ulteriori oggetti del gestore code del cluster per il gestore code, inclusi i canali CLUSSDR , vengono definiti automaticamente come necessario. Ciò consente al gestore code di inviare informazioni sul cluster a qualsiasi repository completo e di inviare messaggi a qualsiasi gestore code nel cluster.

Come spiegato nella sezione di questo articolo, i canali mittenti definiti automaticamente si basano sulla configurazione del canale ricevente del cluster. Pertanto, tutte le proprietà del canale impostate sui canali cluster devono essere impostate in modo identico per i canali CLUSSDR e i canali riceventi del cluster oppure solo per i canali riceventi del cluster.

È necessario definire manualmente solo i canali CLUSSDR per i motivi precedentemente descritti. Vale a dire, per collegare inizialmente un repository parziale a un repository completo o per collegare due repository completi insieme. La configurazione manuale di un canale CLUSSDR che si connette a un repository parziale o a un gestore code non incluso nel cluster, causa l'emissione di messaggi di errore come AMQ9427 e AMQ9428 . Anche se a volte ciò potrebbe essere inevitabile come situazione temporanea, ad esempio quando si modifica l'ubicazione di un repository completo, la definizione manuale deve essere eliminata il più presto possibile.

Canali mittenti del cluster definiti automaticamente

Di solito, quando si aggiunge un gestore code del repository parziale a un cluster, si definiscono manualmente solo due canali cluster sul gestore code:

- Un mittente del cluster (CLUSSDR) da un canale ad un gestore code del repository completo per il cluster.
- Un ricevitore cluster (CLUSRCVR) canale.

Il canale CLUSSDR definito consente al gestore code di stabilire un contatto iniziale con il cluster. Una volta stabilito il contatto iniziale, ulteriori canali CLUSSDR vengono definiti automaticamente dal cluster quando necessario.

Un canale CLUSSDR definito automaticamente prende i propri attributi dalla corrispondente definizione di canale CLUSRCVR sul gestore code di ricezione. Anche se è presente un canale CLUSSDR definito manualmente, vengono utilizzati gli attributi del canale CLUSSDR definito automaticamente. Si supponga, ad esempio, di definire un canale CLUSRCVR senza specificare un numero di porta nel parametro **CONNAME** e definire manualmente un canale CLUSSDR che specifichi un numero di porta. Quando il canale CLUSSDR definito automaticamente sostituisce quello definito manualmente, il numero di porta (preso dal canale CLUSRCVR) diventa vuoto. Viene utilizzato il numero di porta predefinito e il canale ha esito negativo.

Laddove vi sono differenze di configurazione tra un canale CLUSSDR definito manualmente e la corrispondente definizione di canale CLUSRCVR , alcune differenze diventano effettive immediatamente (ad esempio, i parametri di bilanciamento del carico di lavoro) e alcune diventano effettive solo al riavvio del canale (ad esempio, la configurazione TLS).

Per evitare confusione, osservare per quanto possibile le seguenti linee guida:

- Definire solo manualmente i canali CLUSSDR per puntare a repository completi.
- Se si dispone di canali CLUSSDR definiti manualmente, configurarli in modo che corrispondano in modo identico alla definizione di canale CLUSRCVR corrispondente sul gestore code di ricezione.

Consultare anche Gestione dei canali definiti automaticamente.

Concetti correlati

Utilizzo dei canali definiti automaticamente

Utilizzo delle code di trasmissione del cluster e dei canali mittente del cluster

Attività correlate

Configurazione di un nuovo cluster

Aggiunta di un gestore code a un cluster

Argomenti cluster

sono argomenti di amministrazione con l'attributo **cluster** definito. Le informazioni sugli argomenti cluster vengono inoltrate a tutti i membri di un cluster e combinate con argomenti locali per creare parti di uno spazio argomento che si estende su più gestori code. Ciò fa sì che i messaggi pubblicati su un argomento in un gestore code vengano consegnati alle sottoscrizioni di altri gestori code nel cluster.

Quando si definisce un argomento cluster su un gestore code, la definizione dell'argomento cluster viene inviata ai gestori code del repository completo. I repository completi propagano quindi la definizione dell'argomento cluster a tutti i gestori code all'interno del cluster, rendendo disponibile lo stesso argomento cluster ai publisher e sottoscrittori di qualsiasi gestore code del cluster. Il gestore code su cui si crea un argomento cluster è noto come host dell'argomento cluster. L'argomento cluster può essere utilizzato da qualsiasi gestore code nel cluster, ma le modifiche a un cluster devono essere effettuate sul gestore code dove è definito tale argomento (l'host) nel punto in cui la modifica viene propagata a tutti i membri del cluster attraverso i repository completi.

Per informazioni sulla configurazione degli argomenti del cluster per l'utilizzo dell' *instradamento diretto* o dell' *instradamento host dell'argomento*e sull'eredità dell'argomento in cluster e le sottoscrizioni con caratteri jolly, consultare Definizione degli argomenti del cluster.

Per informazioni sui comandi da utilizzare per visualizzare gli argomenti cluster, consultare le informazioni correlate.

Concetti correlati

Utilizzo degli argomenti di gestione

Utilizzo delle sottoscrizioni

Riferimenti correlati

VISUALIZZAZIONEARGOMENTO

STATOVISUALIZZAZIONEP

VISUALIZZAZIONESUB

Oggetti cluster predefiniti

Multi Su Multiplatforms, gli oggetti cluster predefiniti sono inclusi nella serie di oggetti predefiniti creati automaticamente quando si definisce un gestore code. z/OS Su z/OS, le definizioni di oggetti cluster predefinite sono disponibili negli esempi di personalizzazione.

Nota: È possibile modificare le definizioni di canale predefinite come qualsiasi altra definizione di canale, eseguendo i comandi MQSC o PCF. Non modificare le definizioni di coda predefinite, tranne per SYSTEM.CLUSTER.HISTORY.QUEUE.

SYSTEM.CLUSTER.COMMAND.QUEUE
Ogni gestore code in un cluster ha una coda locale denominata SYSTEM.CLUSTER.COMMAND.QUEUE che viene utilizzata per trasferire i messaggi al repository completo. Il messaggio contiene qualsiasi informazione nuova o modificata sul gestore code o qualsiasi richiesta di informazioni su altri gestori code. SYSTEM.CLUSTER.COMMAND.QUEUE è normalmente vuoto.

SYSTEM.CLUSTER.HISTORY.QUEUE
Ogni gestore code in un cluster ha una coda locale denominata SYSTEM.CLUSTER.HISTORY.QUEUE. SYSTEM.CLUSTER.HISTORY.QUEUE viene utilizzato per memorizzare la cronologia delle informazioni sullo stato del cluster per scopi di servizio.

Nelle impostazioni dell'oggetto predefinito, SYSTEM.CLUSTER.HISTORY.QUEUE è impostata su PUT (ENABLED). Per eliminare la raccolta cronologica, modificare l'impostazione in PUT (DISABLED).

SYSTEM.CLUSTER.REPOSITORY.QUEUE
Ogni gestore code in un cluster ha una coda locale denominata SYSTEM.CLUSTER.REPOSITORY.QUEUE. Questa coda viene utilizzata per memorizzare tutte le informazioni complete del repository. Questa coda non è normalmente vuota.

SYSTEM.CLUSTER.TRANSMIT.QUEUE
Ogni Gestore code ha una definizione per una coda locale denominata SYSTEM.CLUSTER.TRANSMIT.QUEUE. SYSTEM.CLUSTER.TRANSMIT.QUEUE è la coda di trasmissione predefinita per tutti i messaggi a tutte le code e i gestori code all'interno dei cluster. È possibile modificare la coda di trasmissione predefinita per ogni canale mittente del cluster in SYSTEM.CLUSTER.TRANSMIT. *ChannelName*, modificando l'attributo gestore code DEFCLXQ . Non è possibile eliminare SYSTEM.CLUSTER.TRANSMIT.QUEUE. Viene utilizzato anche

per definire i controlli di autorizzazione se la coda di trasmissione predefinita utilizzata è SYSTEM.CLUSTER.TRANSMIT.QUEUE o SYSTEM.CLUSTER.TRANSMIT. *ChannelName*.

SYSTEM.DEF.CLUSRCVR
Ogni cluster ha una definizione di canale CLUSRCVR predefinita denominata SYSTEM.DEF.CLUSRCVR. SYSTEM.DEF.CLUSRCVR viene utilizzato per fornire i valori predefiniti per gli attributi che non vengono specificati quando si crea un canale ricevente del cluster su un gestore code nel cluster.

SYSTEM.DEF.CLUSSDR
Ogni cluster ha una definizione di canale CLUSSDR predefinita denominata SYSTEM.DEF.CLUSSDR. SYSTEM.DEF.CLUSSDR viene utilizzato per fornire valori predefiniti per gli attributi che non vengono specificati quando si crea un canale mittente del cluster su un gestore code nel cluster.

Concetti correlati
Utilizzo degli oggetti cluster predefiniti

Pubblicazione/sottoscrizione della messaggistica

La messaggistica di pubblicazione / sottoscrizione consente di disaccoppiare il fornitore di informazioni dai consumatori di tali informazioni. L'applicazione mittente e l'applicazione ricevente non hanno bisogno di sapere nulla l'uno sull'altro per le informazioni da inviare e ricevere.

Prima che un'applicazione IBM MQ point-to-point possa inviare un messaggio ad un'altra applicazione, è necessario che conosca qualcosa su tale applicazione. Ad esempio, deve conoscere il nome della coda a cui inviare le informazioni e potrebbe anche specificare un nome gestore code.

La pubblicazione / sottoscrizione IBM MQ elimina la necessità che la tua applicazione conosca qualcosa sull'applicazione di destinazione. Tutto ciò che l'applicazione di invio deve fare è questo:

- *Inserire* un messaggio IBM MQ che contiene le informazioni richieste dall'applicazione.
- Assegnare il messaggio a un argomento che denota l'oggetto delle informazioni.
- Lasciare che IBM MQ gestisca la distribuzione di tali informazioni.

Allo stesso modo, l'applicazione di destinazione non deve conoscere l'origine delle informazioni che riceve.

La seguente figura mostra il più semplice sistema di pubblicazione / sottoscrizione. Sono presenti un publisher, un gestore code e un sottoscrittore. Una sottoscrizione viene effettuata dal sottoscrittore su un gestore code, una pubblicazione viene inviata dal publisher al gestore code e la pubblicazione viene quindi inoltrata dal gestore code al sottoscrittore.

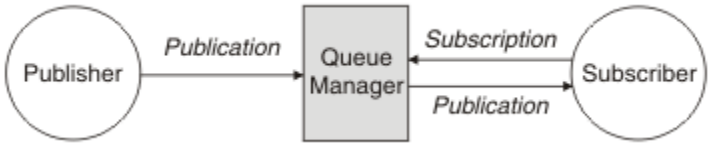


Figura 17. Configurazione di pubblicazione / sottoscrizione semplice

Un tipico sistema di pubblicazione / sottoscrizione ha più di un publisher e più di un sottoscrittore su diversi argomenti e spesso ha più di un gestore code. Un'applicazione può essere sia un publisher che un sottoscrittore.

Un'altra differenza significativa tra la messaggistica di pubblicazione / sottoscrizione e quella point - to - point è che un messaggio inviato a una coda point - to - point viene elaborato solo da una singola applicazione. Un messaggio pubblicato in un argomento di pubblicazione / sottoscrizione, in cui più di un sottoscrittore ha registrato un interesse, viene elaborato da ogni sottoscrittore interessato.

Componenti di pubblicazione / sottoscrizione

La pubblicazione / sottoscrizione è il meccanismo mediante il quale i sottoscrittori possono ricevere informazioni, sotto forma di messaggi, dai publisher. Le interazioni tra i publisher e i sottoscrittori sono controllate dai gestori code, utilizzando le funzioni IBM MQ standard.

Un tipico sistema di pubblicazione / sottoscrizione ha più di un publisher e più di un sottoscrittore su diversi argomenti e spesso ha più di un gestore code. Un'applicazione può essere sia un publisher che un sottoscrittore.

Il fornitore di informazioni è denominato *publisher*. Gli editori forniscono informazioni su un argomento, senza bisogno di sapere nulla sulle applicazioni che sono interessate a tali informazioni. I publisher generano queste informazioni sotto forma di messaggi, denominati *pubblicazioni* che desiderano pubblicare e definire l'argomento di questi messaggi.

Il destinatario di un'informazione è chiamato *sottoscrittore*. I sottoscrittori creano *sottoscrizioni* che descrivono l'argomento a cui il sottoscrittore è interessato. Pertanto, la sottoscrizione determina quali pubblicazioni vengono inoltrate al sottoscrittore. I sottoscrittori possono effettuare più sottoscrizioni e possono ricevere informazioni da diversi publisher.

Le informazioni pubblicate vengono inviate in un messaggio IBM MQ e l'oggetto delle informazioni viene identificato dal relativo argomento . Il publisher specifica l'argomento quando pubblica le informazioni e il sottoscrittore specifica gli argomenti su cui desidera ricevere le pubblicazioni. Al sottoscrittore vengono inviate informazioni solo sugli argomenti sottoscritti.

È l'esistenza di argomenti che consentono ai fornitori e ai consumatori di informazioni di essere disaccoppiati nella messaggistica di pubblicazione / sottoscrizione, eliminando la necessità di includere una specifica destinazione in ogni messaggio come richiesto nella messaggistica point - to - point.

Le interazioni tra i publisher e i sottoscrittori sono tutte controllate da un gestore code. Il gestore code riceve i messaggi dai publisher e le sottoscrizioni dai sottoscrittori (a una gamma di argomenti). Il lavoro del gestore code consiste nell'instradare i messaggi pubblicati ai sottoscrittori che hanno registrato un interesse nell'argomento dei messaggi.

Le funzioni IBM MQ standard vengono utilizzate per distribuire i messaggi, in modo che le proprie applicazioni possano utilizzare tutte le funzioni disponibili per le applicazioni IBM MQ esistenti. Ciò significa che è possibile utilizzare i messaggi persistenti per ottenere una consegna garantita una sola volta e che i messaggi possono far parte di un'unità di lavoro transazionale per garantire che i messaggi vengano consegnati al sottoscrittore solo se ne è stato eseguito il commit da parte del publisher.

Editori e pubblicazioni

In IBM MQ la pubblicazione / sottoscrizione di un publisher è un'applicazione che rende disponibili le informazioni su un argomento specificato per un gestore code sotto forma di un messaggio IBM MQ standard denominato pubblicazione. Un publisher può pubblicare informazioni su più di un argomento.

I publisher utilizzano il verbo MQPUT per inserire un messaggio in un argomento precedentemente aperto, questo messaggio è una pubblicazione. Il gestore code locale instrada quindi la pubblicazione a tutti i sottoscrittori che hanno sottoscrizioni all'argomento della pubblicazione. Un messaggio pubblicato può essere utilizzato da più di un sottoscrittore.

Oltre a distribuire le pubblicazioni a tutti i sottoscrittori locali che dispongono di sottoscrizioni appropriate, un gestore code può anche distribuire la pubblicazione a qualsiasi altro gestore code ad esso connesso, direttamente o tramite una rete di gestori code che hanno sottoscrittori all'argomento.

In una rete di pubblicazione / sottoscrizione IBM MQ , un'applicazione di pubblicazione può essere anche un sottoscrittore.

Pubblicazioni nel punto di sincronizzazione

Gli autori (publisher) possono emettere chiamate MQPUT o MQPUT1 nel punto di sincronizzazione per includere tutti i messaggi consegnati ai sottoscrittori in un'unità di lavoro. Se l'opzione MQPMO_RETAIN o le opzioni di consegna dell'argomento NPMSGDLV o PMSGDLV con i valori ALL o ALLDUR sono specificate,

il gestore code utilizza le chiamate interne MQPUT o MQPUT1 nel punto di sincronizzazione, nell'ambito della chiamata MQPUT o MQPUT1 del publisher.

Informazioni di stato ed evento

Le pubblicazioni possono essere categorizzate come pubblicazioni di stato, come il prezzo corrente di un'azione, o pubblicazioni di eventi, come una compravendita di tale azione.

Pubblicazioni di stato

Le *pubblicazioni di stato* contengono informazioni sullo stato corrente di qualcosa, come il prezzo delle azioni o il punteggio corrente in una partita di calcio. Quando succede qualcosa (ad esempio, si verifica una modifica nel listino di borsa o nel risultato della partita di calcio), le informazioni relative allo stato precedente non vengono più richieste, poiché vengono sostituite dalle nuove informazioni.

Un sottoscrittore desidera ricevere la versione corrente delle informazioni di stato all'avvio e ricevere nuove informazioni ogni volta che lo stato cambia.

Se una pubblicazione contiene informazioni sullo stato, viene spesso pubblicata come una pubblicazione conservata. Un nuovo sottoscrittore in genere desidera che le informazioni sullo stato corrente vengano immediatamente; il sottoscrittore non desidera attendere un evento che causa la ripubblicazione delle informazioni. I sottoscrittori riceveranno automaticamente la pubblicazione conservata di un argomento quando lo sottoscriveranno, a meno che il sottoscrittore non utilizzi le opzioni MQSO_PUBLICATIONS_ON_REQUEST o MQSO_NEW_PUBLICATIONS_ONLY.

Pubblicazioni di eventi

Le *pubblicazioni di eventi* contengono informazioni sui singoli eventi che si verificano, ad esempio una compravendita in qualche borsa o il calcolo del punteggio di un determinato gol. Ciascun evento è indipendente dagli altri.

Un sottoscrittore desidera ricevere informazioni sugli eventi man mano che si verificano.

Pubblicazioni conservate

Per impostazione predefinita, dopo che una pubblicazione viene inviata a tutti i sottoscrittori interessati, viene eliminata. Tuttavia, un editore può specificare che venga conservata una copia di una pubblicazione in modo che possa essere inviata ai futuri sottoscrittori che registrano un interesse nell'argomento.

L'eliminazione delle pubblicazioni dopo che sono state inviate a tutti i sottoscrittori interessati è adatta per le informazioni sugli eventi, ma non è sempre adatta per le informazioni sullo stato. Conservando un messaggio, i nuovi sottoscrittori non devono attendere che le informazioni vengano pubblicate di nuovo prima di ricevere le informazioni di stato iniziali. Ad esempio, un sottoscrittore con una sottoscrizione a un prezzo azionario riceverà immediatamente il prezzo corrente, senza attendere che il prezzo del titolo cambi (e quindi venga ripubblicato).

Il gestore code può conservare solo una pubblicazione per ciascun argomento, quindi la pubblicazione conservata esistente di un argomento viene eliminata quando una nuova pubblicazione conservata arriva al gestore code. Tuttavia, l'eliminazione della pubblicazione esistente potrebbe non verificarsi in modo sincrono con l'arrivo della nuova pubblicazione conservata. Pertanto, ove possibile, non più di un editore che invia pubblicazioni conservate su qualsiasi argomento.

I sottoscrittori possono specificare che non desiderano ricevere le pubblicazioni conservate utilizzando l'opzione di sottoscrizione MQSO_NEW_PUBLICATIONS_ONLY. I sottoscrittori esistenti possono richiedere l'invio di copie duplicate delle pubblicazioni conservate.

In alcuni casi, è possibile che non si desideri conservare le pubblicazioni, anche per le informazioni sullo stato:

- Se tutte le sottoscrizioni ad un argomento vengono effettuate prima di qualsiasi pubblicazione su tale argomento e non si prevedono o non si consentono nuove sottoscrizioni, non è necessario conservare

le pubblicazioni perché vengono consegnate alla serie completa di sottoscrittori la prima volta che vengono pubblicate.

- Se le pubblicazioni si verificano frequentemente, ad esempio ogni secondo, un nuovo sottoscrittore (o un sottoscrittore in fase di ripristino da un errore) riceve lo stato corrente quasi immediatamente dopo la sottoscrizione iniziale, quindi non è necessario conservare tali pubblicazioni.
- Se le pubblicazioni sono di grandi dimensioni, potrebbe essere necessaria una notevole quantità di spazio di archiviazione per memorizzare la pubblicazione conservata per ogni argomento. In un ambiente con più gestori code, le pubblicazioni conservate sono memorizzate da tutti i gestori code della rete che hanno una sottoscrizione corrispondente.

Quando si decide se utilizzare le pubblicazioni conservate, considerare il modo in cui le applicazioni di sottoscrizione si ripristinano da un errore. Se l'autore (publisher) non utilizza pubblicazioni conservate, l'applicazione del sottoscrittore (subscriber) potrebbe aver necessità di memorizzare il relativo stato corrente in locale.

Per assicurarsi che una pubblicazione venga conservata, utilizzare l'opzione put - message MQPMO_RETAIN. Se viene utilizzata questa opzione e la pubblicazione non può essere conservata, il messaggio non viene pubblicato e la chiamata ha esito negativo con MQRC_PUT_NOT_USED.

Se un messaggio è una pubblicazione conservata, ciò viene indicato dalla proprietà del messaggio MQIsRetained . La persistenza di un messaggio è quella di quando è stato originariamente pubblicato.

Concetti correlati

Considerazioni di progettazione per le pubblicazioni conservate nei cluster di pubblicazione / sottoscrizione

Pubblicazioni nel punto di sincronizzazione

Nella pubblicazione / sottoscrizione IBM MQ , il punto di sincronizzazione può essere utilizzato dai publisher o internamente dal gestore code.

I publisher utilizzano il punto di sincronizzazione quando emettono chiamate MQPUT/MQPUT1 con opzione MQPMO_SYNCPOINT. Tutti i messaggi inviati ai sottoscrittori vengono conteggiati per il numero massimo di messaggi di cui non è stato eseguito il commit in un'unità di lavoro work.The MAXUMSGS specifica questo limite. Se il limite viene raggiunto, il publisher riceve il codice di errore 2024 (07E8) (RC2024): MQRC_SYNCPOINT_LIMIT_RAGGIUNTO .

Quando un publisher emette chiamate MQPUT/MQPUT1 utilizzando MQPMO_NO_SYNCPOINT con l'opzione MQPMO_RETAIN o le opzioni di consegna argomenti NPMSGDLV/PMSGDLV con valori ALL o ALLDUR, il gestore code utilizza punti di sincronizzazione interni per garantire che i messaggi vengano consegnati come richiesto. Il publisher può ricevere il codice motivo 2024 (07E8) (RC2024): MQRC_SYNCPOINT_LIMIT_RAGGIUNTI se il limite viene raggiunto nell'ambito della chiamata MQPUT/MQPUT1 del publisher.

Sottoscrittori e sottoscrizioni

Nella pubblicazione / sottoscrizione IBM MQ , un sottoscrittore è un'applicazione che richiede informazioni su un argomento specifico da un gestore code in una rete di pubblicazione / sottoscrizione. Un sottoscrittore può ricevere messaggi, sullo stesso argomento o su argomenti diversi, da più di un publisher.

Le sottoscrizioni possono essere create manualmente utilizzando un comando MQSC o dalle applicazioni. Queste sottoscrizioni vengono emesse per il gestore code locale e contengono informazioni sulle pubblicazioni che il sottoscrittore desidera ricevere:

- L'argomento a cui il sottoscrittore è interessato; questo può risolversi in più argomenti se vengono utilizzati i caratteri jolly.
- Una stringa di selezione facoltativa da applicare ai messaggi pubblicati.
- Un handle per una coda (nota come *coda del sottoscrittore*), su cui devono essere collocate le pubblicazioni selezionate e il CorrelIdfacoltativo.

Il gestore code locale memorizza le informazioni sulla sottoscrizione e, quando riceve una pubblicazione, esegue la scansione delle informazioni per stabilire se esiste una sottoscrizione che corrisponde all'argomento della pubblicazione e alla stringa di selezione. Per ogni sottoscrizione corrispondente, il gestore code indirizza la pubblicazione alla coda del sottoscrittore del sottoscrittore. Le informazioni che un gestore code memorizza sulle sottoscrizioni possono essere visualizzate utilizzando i comandi DIS SUB e DIS SBSTATUS.

Una sottoscrizione viene eliminata solo quando si verifica uno dei seguenti eventi:

- Il sottoscrittore (subscriber) annulla la sottoscrizione utilizzando la chiamata MQCLOSE (se la sottoscrizione è stata effettuata in modo non duraturo).
- La sottoscrizione scade.
- La sottoscrizione viene cancellata dal responsabile di sistema utilizzando il comando DELETE SUB.
- L'applicazione del sottoscrittore viene terminata (se la sottoscrizione è stata effettuata in modo non duraturo).
- Il gestore code viene arrestato o riavviato (se la sottoscrizione è stata effettuata in maniera non durevole).

Quando si ricevono i messaggi, utilizzare le opzioni appropriate sulla chiamata MQGET. Se l'applicazione elabora solo i messaggi per una sottoscrizione, è necessario utilizzare almeno `get-by-correlid`, come dimostrato nel programma di esempio C `amqssbxa.c` e all'indirizzo sottoscrittore MQ non gestito. Il **`CorrelId`** da utilizzare viene restituito da MQSUB in MQSD.Campo **`SubCorrelId`** .

Concetti correlati

Sottoscrizioni clonate e condivise

Riferimenti correlati

Esempi di come definire la proprietà sharedSubscription

Code gestite e pubblicazione / sottoscrizione

Quando si crea una sottoscrizione è possibile scegliere di utilizzare l'accodamento gestito. Se si utilizza l'accodamento gestito, una coda di sottoscrizione viene creata automaticamente quando si crea una sottoscrizione. Le code gestite vengono ridefinite automaticamente in base alla durata della sottoscrizione. L'utilizzo delle code gestite significa che non è necessario preoccuparsi della creazione di code per ricevere pubblicazioni e le pubblicazioni non utilizzate vengono rimosse automaticamente dalle code del sottoscrittore se viene chiusa una connessione di sottoscrizione non durevole.

Se un'applicazione non ha bisogno di utilizzare una particolare coda come coda del sottoscrittore, la destinazione per le pubblicazioni che riceve, può utilizzare le *sottoscrizioni gestite* utilizzando l'opzione di sottoscrizione MQSO_MANAGED. Se si crea una sottoscrizione gestita, il gestore code restituisce un handle di oggetto al sottoscrittore per una coda sottoscrittore creata dal gestore code in cui verranno ricevute le pubblicazioni. Ciò è dovuto al fatto che una *sottoscrizione gestita* è quella in cui IBM MQ gestisce la sottoscrizione. L'handle dell'oggetto della coda verrà restituito consentendo di sfogliare, richiamare o analizzare la coda (non è possibile inserire o impostare attributi di una coda gestita a meno che non sia stato esplicitamente concesso l'accesso alle code dinamiche temporanee).

La durata della sottoscrizione determina se la coda gestita rimane quando la connessione dell'applicazione di sottoscrizione al gestore code è interrotta.

Le sottoscrizioni gestite sono particolarmente utili quando vengono utilizzate con sottoscrizioni non durevoli perché quando la connessione dell'applicazione viene terminata, i messaggi non consumati rimarrebbero sulla coda del sottoscrittore occupando spazio indefinito nel gestore code. Se si utilizza una sottoscrizione gestita, la coda gestita sarà una coda dinamica temporanea e come tale verrà eliminata insieme ai messaggi non utilizzati quando la connessione viene interrotta per uno dei seguenti motivi:

- MQCLOSE con MQCO_REMOVE_SUB viene utilizzato e l'Hobj gestito viene chiuso.
- una connessione viene persa per un'applicazione che utilizza una sottoscrizione non durevole (MQSO_NON_DURABLE).
- una sottoscrizione viene rimossa perché è scaduta e l'Hobj gestito è chiuso.

Le sottoscrizioni gestite possono essere utilizzate anche con sottoscrizioni durevoli, ma è possibile che si desideri lasciare messaggi non utilizzati nella coda del sottoscrittore in modo che possano essere richiamati quando la connessione viene riaperta. Per questo motivo, le code gestite per le sottoscrizioni durevoli assumono la forma di una coda dinamica permanente e rimarranno quando la connessione dell'applicazione di sottoscrizione al gestore code viene interrotta.

È possibile impostare una scadenza sulla sottoscrizione se si desidera utilizzare una coda gestita dinamica permanente in modo che, anche se la coda esisterà ancora dopo che la connessione è stata interrotta, non continuerà a esistere indefinitamente.

Se si elimina la coda gestita, si riceverà un messaggio di errore.

Le code gestite che vengono create sono denominate con numeri alla fine (timestamp) in modo che ciascuno sia univoco.

Durata sottoscrizione

Le sottoscrizioni possono essere configurate come durevoli o non durevoli. La durata della sottoscrizione determina cosa succede alle sottoscrizioni quando le applicazioni di sottoscrizione si disconnettono da un gestore code.

Sottoscrizioni durevoli

Le sottoscrizioni durevoli continuano a esistere quando una connessione dell'applicazione di sottoscrizione al gestore code viene chiusa. Se una sottoscrizione è durevole, quando l'applicazione di sottoscrizione si disconnette, la sottoscrizione rimane attiva e può essere utilizzata dall'applicazione di sottoscrizione quando si riconnette richiedendo nuovamente la sottoscrizione utilizzando il **SubName** restituito quando è stata creata la sottoscrizione.

Quando si esegue la sottoscrizione in modo duraturo, un nome sottoscrizione (**SubName**) è obbligatorio. I nomi delle sottoscrizioni devono essere univoci all'interno di un gestore code in modo che possano essere utilizzati per identificare una sottoscrizione. Questo metodo di identificazione è necessario quando si specifica una sottoscrizione che si desidera riprendere, se la connessione alla sottoscrizione è stata deliberatamente chiusa (utilizzando l'opzione MQCO_KEEP_SUB) o è stata disconnessa dal gestore code. È possibile riprendere una sottoscrizione esistente utilizzando la chiamata MQSUB con opzione MQSO_RESUME. I nomi delle sottoscrizioni vengono visualizzati anche se si utilizza il comando DISPLAY SBSTATUS con SUBTYPE ALL o ADMIN.

Quando un'applicazione non richiede più una sottoscrizione duratura, è possibile rimuoverla utilizzando la chiamata alla funzione MQCLOSE con l'opzione MQCO_REMOVE_SUB oppure è possibile eliminarla manualmente utilizzando il comando MQSC DELETE SUB.

È possibile utilizzare l'attributo dell'argomento **DURSUB** per specificare se è possibile effettuare o meno sottoscrizioni durevoli a un argomento.

Al ritorno da una chiamata MQSUB che utilizza l'opzione MQSO_RESUME, la scadenza della sottoscrizione viene impostata sulla scadenza originale della sottoscrizione e non sulla scadenza rimanente.

Un gestore code continua a inviare pubblicazioni per soddisfare una sottoscrizione durevole anche se l'applicazione del sottoscrittore non è connessa. Ciò porta ad un accumulo di messaggi nella coda del sottoscrittore. Il modo più semplice per evitare questo problema è quello di utilizzare un abbonamento non durevole dove appropriato. Tuttavia, quando è necessario utilizzare sottoscrizioni durevoli, è possibile evitare una creazione di messaggi se il sottoscrittore (subscriber) effettua la sottoscrizione utilizzando l'opzione Pubblicazioni conservate . Un sottoscrittore può quindi controllare quando riceve le pubblicazioni utilizzando la chiamata MQSUBRQ.

Sottoscrizioni non durevoli

Le sottoscrizioni non durevoli esistono solo finché la connessione dell'applicazione di sottoscrizione al gestore code rimane aperta. La sottoscrizione viene rimossa quando l'applicazione di sottoscrizione si disconnette dal gestore code deliberatamente o a causa di un'interruzione. Quando la connessione viene chiusa, le informazioni relative alla sottoscrizione vengono rimosse dal gestore code e non vengono più

visualizzate se si visualizzano le sottoscrizioni utilizzando il comando DISPLAY SBSTATUS. Nessun altro messaggio viene inserito nella coda del sottoscrittore.

Ciò che accade alle pubblicazioni non utilizzate sulla coda del sottoscrittore per le sottoscrizioni non durevoli è determinato come segue.

- Se un'applicazione di sottoscrizione utilizza una destinazione gestita, tutte le pubblicazioni che non sono state utilizzate vengono rimosse automaticamente.
- Se l'applicazione di sottoscrizione fornisce un handle alla propria coda del sottoscrittore quando effettua la sottoscrizione, i messaggi non utilizzati non verranno rimossi automaticamente. È responsabilità dell'applicazione cancellare la coda, se appropriato. Se la coda è condivisa da più di un sottoscrittore o da altre applicazioni point - to - point, potrebbe non essere appropriato cancellare completamente la coda.

Sebbene non sia richiesto per sottoscrizioni non durevoli, il gestore code utilizza un nome di sottoscrizione, se fornito. I nomi delle sottoscrizioni devono essere univoci nel gestore code in modo che possano essere utilizzati per identificare una sottoscrizione.

Concetti correlati

Sottoscrizioni clonate e condivise

Attività correlate

Utilizzo delle sottoscrizioni condivise JMS 2.0

Riferimenti correlati

Esempi di come definire la proprietà sharedSubscription

Stringhe di selezione

Una *stringa di selezione* è un'espressione che viene applicata a una pubblicazione per determinare se corrisponde a una sottoscrizione. Le stringhe di selezione possono includere caratteri jolly.

Quando si esegue la sottoscrizione, oltre a specificare un argomento, è possibile specificare una stringa di selezione per selezionare le pubblicazioni in base alle relative proprietà del messaggio.

La stringa di selezione viene valutata rispetto al messaggio inserito dal publisher prima che venga modificato per la consegna a ogni sottoscrittore. Prestare attenzione quando si utilizzano i campi nella stringa di selezione che potrebbero essere modificati come parte dell'operazione di pubblicazione. Ad esempio, i campi MQMD `UserIdentifier`, `MsgIde` `CorrelId`.

Le stringhe di selezione non devono fare riferimento ai campi delle proprietà del messaggio aggiunti dal gestore code come parte dell'operazione di pubblicazione (vedere Proprietà del messaggio di pubblicazione / sottoscrizione), ad eccezione della proprietà del messaggio `MQTopicString`, che contiene la stringa dell'argomento per la pubblicazione.

Concetti correlati

Regole e limitazioni della stringa di selezione

Argomenti

Un argomento è l'oggetto delle informazioni pubblicate in un messaggio di pubblicazione / sottoscrizione.

I messaggi nei sistemi point-to-point vengono inviati a un indirizzo di destinazione specifico. I messaggi nei sistemi di pubblicazione / sottoscrizione basati sull'oggetto vengono inviati ai sottoscrittori in base all'oggetto che descrive il contenuto del messaggio. Nei sistemi basati sul contenuto, i messaggi vengono inviati agli utenti in base al contenuto del messaggio stesso.

Il sistema di pubblicazione / sottoscrizione IBM MQ è un sistema di pubblicazione / sottoscrizione basato sull'oggetto. Un publisher crea un messaggio e lo pubblica con una stringa di argomenti che meglio si adatta all'oggetto della pubblicazione. Per ricevere pubblicazioni, un sottoscrittore crea una sottoscrizione con una stringa di argomenti corrispondente al modello per selezionare gli argomenti di pubblicazione. Il gestore code consegna le pubblicazioni ai sottoscrittori che hanno sottoscrizioni che corrispondono all'argomento della pubblicazione e sono autorizzati a ricevere le pubblicazioni. L'articolo, “Stringhe argomento” a pagina 70, descrive la sintassi delle stringhe argomento che identificano l'argomento

di una pubblicazione. I sottoscrittori creano anche stringhe di argomenti per selezionare quali argomenti ricevere. Le stringhe di argomenti create dai sottoscrittori possono contenere uno dei due schemi di caratteri jolly alternativi per la corrispondenza del modello rispetto alle stringhe di argomenti nelle pubblicazioni. La corrispondenza del modello è descritta in “Schemi dei caratteri jolly” a pagina 71.

Nella pubblicazione / sottoscrizione basata sull'oggetto, i publisher o gli amministratori sono responsabili della classificazione dei soggetti in argomenti. In genere, i soggetti sono organizzati gerarchicamente, in strutture ad alberi di argomenti, utilizzando il carattere ' / ' per creare sottoargomenti nella stringa di argomento. Consultare “Strutture ad albero degli argomenti” a pagina 77 per esempi di strutture ad albero degli argomenti. Gli argomenti sono nodi nella struttura ad albero degli argomenti. Gli argomenti possono essere nodi foglia senza ulteriori argomenti secondari o nodi intermedi con argomenti secondari.

In parallelo con l'organizzazione dei soggetti in una struttura gerarchica di argomenti, è possibile associare gli argomenti agli oggetti argomento di gestione. Gli attributi vengono assegnati a un argomento, ad esempio se l'argomento viene distribuito in un cluster, associandolo a un oggetto argomento di gestione. L'associazione viene effettuata denominando l'argomento utilizzando l'attributo `TOPICSTR` dell'oggetto argomento di gestione. Se non si associa esplicitamente un oggetto argomento di gestione a un argomento, l'argomento eredita gli attributi del suo predecessore più vicino nella struttura ad albero degli argomenti *associata* a un oggetto argomento di gestione. Se non è stato definito alcun argomento principale, viene ereditato da `SYSTEM.BASE.TOPIC`. Gli oggetti argomento di gestione sono descritti in “Oggetti argomento di gestione” a pagina 78.

Nota: Anche se si ereditano tutti gli attributi di un argomento da `SYSTEM.BASE.TOPIC`, definire un argomento root per i propri argomenti che eredita direttamente da `SYSTEM.BASE.TOPIC`. Ad esempio, nello spazio argomento degli stati degli Stati Uniti, `USA/Alabama` `USA/Alaskae` così via, `USA` è l'argomento root. Lo scopo principale dell'argomento root è creare spazi argomento discreti e non sovrapposti per evitare che le pubblicazioni corrispondano alle sottoscrizioni errate. Ciò significa anche che è possibile modificare gli attributi dell'argomento root per influire sull'intero spazio argomento. Ad esempio, è possibile impostare il nome per l'attributo **CLUSTER** .

Quando si fa riferimento a un argomento come publisher o sottoscrittore, è possibile scegliere di fornire una stringa di argomenti o fare riferimento a un oggetto argomento. Oppure è possibile eseguire entrambe le operazioni, nel qual caso la stringa di argomenti fornita definisce un sottoargomento dell'oggetto argomento. Il gestore code identifica l'argomento accodando la stringa dell'argomento al prefisso della stringa dell'argomento indicato nell'oggetto argomento, inserendo un ulteriore ' / ' tra le due stringhe dell'argomento, ad esempio *stringa dell'argomento/stringa dell'oggetto*. “Combinazione di stringhe di argomenti” a pagina 75 lo descrive ulteriormente. La stringa di argomenti risultante viene utilizzata per identificare l'argomento e associarlo a un oggetto argomento di gestione. L'oggetto argomento di gestione non è necessariamente lo stesso oggetto argomento dell'oggetto argomento corrispondente all'argomento principale.

Nella pubblicazione / sottoscrizione basata sul contenuto, si definiscono i messaggi che si desidera ricevere fornendo stringhe di selezione che ricercano il contenuto di ogni messaggio. IBM MQ fornisce una forma intermedia di pubblicazione / sottoscrizione basata su contenuto utilizzando i selettori di messaggi che eseguono la scansione delle proprietà del messaggio piuttosto che del contenuto completo del messaggio, consultare Selettori. L'utilizzo archetipico dei selettori di messaggi consiste nel sottoscrivere un argomento e quindi qualificare la selezione con una proprietà numerica. Il selettore consente di specificare che si è interessati ai valori solo in un determinato intervallo; operazione che non è possibile eseguire utilizzando caratteri o caratteri jolly basati sull'argomento. Se è necessario filtrare in base al contenuto completo del messaggio, è necessario utilizzare IBM Integration Bus.

Stringhe argomento

Etichettare le informazioni pubblicate come argomento utilizzando una stringa di argomenti. Eseguire la sottoscrizione a gruppi di argomenti utilizzando stringhe di argomenti jolly basate su caratteri o argomenti.

Argomenti

Una *stringa argomento* è una stringa di caratteri che identifica l'argomento di un messaggio di pubblicazione / sottoscrizione. È possibile utilizzare qualsiasi carattere che si desidera quando si crea una stringa di argomenti.

Stringa argomento

Tre caratteri hanno un significato particolare nella pubblicazione / sottoscrizione IBM WebSphere MQ 7 . Sono consentiti in qualsiasi punto di una stringa di argomenti, ma sono utilizzati con cautela. L'uso dei caratteri speciali è spiegato in “Schema dei caratteri jolly basati su argomenti” a pagina 72.

Una barra (/)

Il separatore di livello argomento. Utilizzare il carattere `'/'` per strutturare l'argomento in una albero degli argomenti.

Evitare i livelli di argomento vuoti, `'//'`, se possibile. Questi corrispondono ai nodi nella gerarchia degli argomenti senza alcuna stringa di argomenti. Un `'/'` iniziale o finale in una stringa di argomento corrisponde a un nodo vuoto iniziale o finale e deve essere evitato.

Il segno hash (#)

Utilizzato in combinazione con `'/'` per creare un carattere jolly a più livelli nelle sottoscrizioni. Prestare attenzione all'utilizzo di `'#'` accanto a `'/'` nelle stringhe di argomento utilizzate per denominare gli argomenti pubblicati. “Esempi di stringhe di argomento” a pagina 71 mostra un uso ragionevole di `'#'`.

Le stringhe `'.../#/...'`, `'#/...'` e `'.../#'` hanno un significato speciale nelle stringhe degli argomenti di sottoscrizione. Le stringhe corrispondono a tutti gli argomenti a uno o più livelli nella gerarchia degli argomenti. Pertanto, se è stato creato un argomento con una di queste sequenze, non è possibile sottoscriverlo, senza sottoscrivere anche tutti gli argomenti a più livelli nella gerarchia degli argomenti.

Il segno più (+)

Utilizzato in combinazione con `'/'` per creare un carattere jolly a livello singolo nelle sottoscrizioni. Prestare attenzione all'utilizzo di `'+'` accanto a `'/'` nelle stringhe di argomento utilizzate per denominare gli argomenti pubblicati.

Le stringhe `'.../+/...'`, `'+/...'` e `'.../+'` hanno un significato speciale nelle stringhe degli argomenti di sottoscrizione. Le stringhe corrispondono a tutti gli argomenti ad un livello nella gerarchia degli argomenti. Pertanto, se è stato creato un argomento con una di queste sequenze, non è possibile sottoscriverlo, senza sottoscrivere anche tutti gli argomenti ad un unico livello nella gerarchia degli argomenti.

Esempi di stringhe di argomento

```
IBM/Business Area#/Results
IBM/Diversity/%African American
```

Riferimenti correlati

TOPIC

Schemi dei caratteri jolly

Esistono due schemi di caratteri jolly utilizzati per sottoscrivere più argomenti. La scelta dello schema è un'opzione di sottoscrizione.

MQSO_WILDCARD_TOPIC

Selezionare gli argomenti da sottoscrivere utilizzando lo schema di caratteri jolly basato sugli argomenti.

Questo è il valore predefinito se non viene selezionato esplicitamente alcuno schema di caratteri jolly.

MQSO_WILDCARD_CHAR
Selezionare gli argomenti da sottoscrivere utilizzando lo schema di caratteri jolly basato sui caratteri.

Impostare uno schema specificando il parametro **wschema** nel comando DEFINE SUB. Per ulteriori informazioni, vedere DEFINE SUB.

Nota: Le sottoscrizioni create prima di IBM WebSphere MQ 7.0 utilizzano sempre lo schema di caratteri jolly basato sui caratteri.

Esempi

```
IBM/+/Results
#/Results
IBM/Software/Results
IBM/*ware/Results
```

Schema dei caratteri jolly basati su argomenti
I caratteri jolly basati sull'argomento consentono ai sottoscrittori di sottoscrivere più di un argomento alla volta.

I caratteri jolly basati su argomenti sono una potente funzione del sistema di argomenti nella pubblicazione / sottoscrizione IBM MQ . I caratteri jolly multilivello e di livello singolo possono essere usati per le sottoscrizioni, ma non possono essere usati all'interno di un argomento dal publisher del messaggio.

Lo schema di caratteri jolly basati su argomenti consente di selezionare le pubblicazioni raggruppate per livello di argomento. È possibile scegliere, per ogni livello nella gerarchia degli argomenti, se la stringa nella sottoscrizione per quel livello di argomento deve corrispondere esattamente alla stringa nella pubblicazione o meno. Ad esempio, la sottoscrizione `IBM/+/Results` seleziona tutti gli argomenti,

```
IBM/Software/Results
IBM/Services/Results
IBM/Hardware/Results
```

Esistono due tipi di caratteri jolly.

wildcard a più livelli

- Il carattere jolly multilivello viene utilizzato nelle sottoscrizioni. Quando viene utilizzato in una pubblicazione, viene considerato come un valore letterale.
- Il carattere jolly multilivello `'#'` viene utilizzato per corrispondere a qualsiasi numero di livelli all'interno di un argomento. Ad esempio, utilizzando la struttura ad albero degli argomenti di esempio, se si esegue la sottoscrizione a `'USA/Alaska/#'`, si ricevono messaggi sugli argomenti `'USA/Alaska'` e `'USA/Alaska/Juneau'`.
- Il carattere jolly multilivello può rappresentare zero o più livelli. Pertanto, `'USA/#'` può anche corrispondere al singolare `'USA'`, dove `'#'` rappresenta zero livelli. Il separatore di livello argomento non ha senso in questo contesto, perché non ci sono livelli da separare.
- Il carattere jolly multilivello è valido solo quando viene specificato da solo o accanto al carattere separatore del livello di argomento. Pertanto, `'#'` e `'USA/#'` sono argomenti validi in cui il carattere `'#'` viene considerato come un carattere jolly. Tuttavia, sebbene `'USA#'` sia anche una stringa di argomenti valida, il carattere `'#'` non viene considerato come un carattere jolly e non ha alcun significato speciale. Per ulteriori informazioni, consultare “Quando i caratteri jolly basati sugli argomenti non sono jolly” a pagina 74 .

Carattere jolly di livello singolo

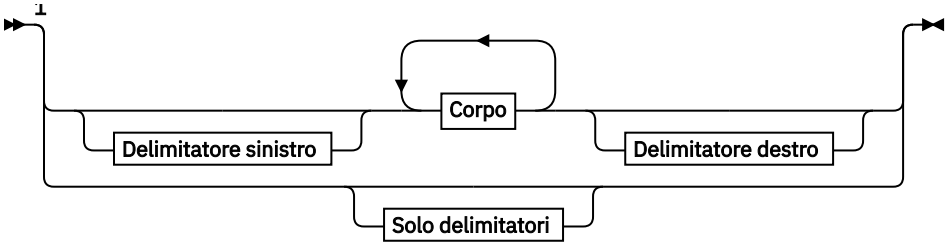
- Il singolo carattere jolly viene utilizzato nelle sottoscrizioni. Quando viene utilizzato in una pubblicazione, viene considerato come un valore letterale.
- Il carattere jolly a livello singolo `'+'` corrisponde ad un solo livello di argomento. Ad esempio, `'USA/+'` corrisponde a `'USA/Alabama'`, ma non `'USA/Alabama/Auburn'`. Poiché il carattere jolly a livello singolo corrisponde solo ad un singolo livello, `'USA/+'` non corrisponde a `'USA'`.

- Il carattere jolly a livello singolo può essere utilizzato a qualsiasi livello nella struttura ad albero degli argomenti e insieme al carattere jolly multilivello. Il carattere jolly a livello singolo deve essere specificato accanto al separatore di livello argomento, tranne quando è specificato da solo. Pertanto, `'+'` e `'USA/+'` sono argomenti validi in cui il carattere `'+'` viene considerato come un carattere jolly. Tuttavia, sebbene `'USA+'` sia anche una stringa di argomenti valida, il carattere `'+'` non viene considerato come un carattere jolly e non ha alcun significato speciale. Per ulteriori informazioni, consultare “Quando i caratteri jolly basati sugli argomenti non sono jolly” a pagina 74 .

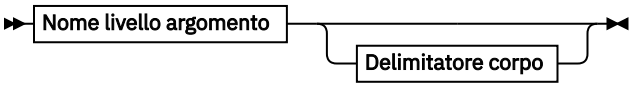
La sintassi per lo schema del carattere jolly basato sull'argomento non ha caratteri escape. Il fatto che `'#'` e `'+'` vengano trattati o meno come caratteri jolly dipende dal loro contesto. Per ulteriori informazioni, consultare “Quando i caratteri jolly basati sugli argomenti non sono jolly” a pagina 74 .

Nota: L'inizio e la fine di una stringa di argomenti vengono trattati in modo speciale. Utilizzando `'$'` per indicare la fine della stringa, `'$#/...'` è un carattere jolly multilivello e `'$/#/..'`. è un nodo vuoto nella root, seguito da un carattere jolly multilivello.

Stringa di caratteri jolly basata sull'argomento

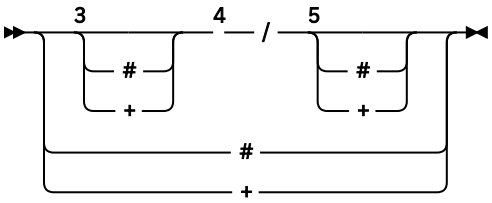


Corpo



Nome livello argomento

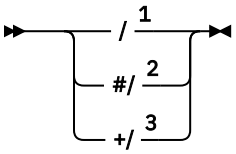
Solo delimitatori



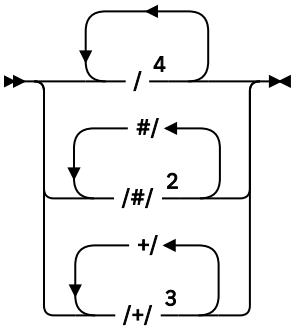
Note:

[1] Una stringa di argomento di lunghezza null o zero non è valida
[2] Si consiglia di non utilizzare `*`, `?`, `%` nelle stringhe dei nomi di livello per la compatibilità tra schemi di caratteri jolly basati su caratteri e su argomenti.
[3] Questi casi sono equivalenti al pattern *delimitatore di sinistra* .
[4] / senza caratteri jolly corrisponde a un singolo argomento vuoto.
[5] Questi casi sono equivalenti al pattern *delimitatore di destra* .
[6] Associa ogni argomento.
[7] Corrisponde a ogni argomento in cui è presente un solo livello.

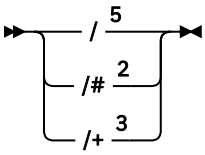
Delimitatore di sinistra



Delimitatore corpo



Delimitatore di destra



Note:

[1] La stringa dell'argomento inizia con un argomento vuoto
[2] Corrisponde a zero o più livelli. Più stringhe di corrispondenza multilivello hanno lo stesso effetto di una stringa di corrispondenza multilivello.
[3] Corrisponde esattamente a un livello.
[4] // è un argomento vuoto - un oggetto argomento senza stringa di argomenti.
[5] La stringa di argomenti termina con un argomento vuoto

Quando i caratteri jolly basati sugli argomenti non sono jolly

I caratteri jolly '+' e '#' non hanno alcun significato speciale quando sono combinati con altri caratteri (inclusi se stessi) in un livello di argomento.

Ciò significa che gli argomenti che contengono '+' o '#' insieme ad altri caratteri in un livello di argomento possono essere pubblicati.

Per esempio, considerare i due argomenti seguenti:

1. level0/level1/+/level4/#
2. level0/level1/#+/level4/level#

Nel primo esempio, i caratteri '+' e '#' vengono trattati come caratteri jolly e quindi non sono validi in una stringa di argomenti che deve essere pubblicata ma che sono validi in una sottoscrizione.

Nel secondo esempio, i caratteri '+' e '#' non vengono considerati come caratteri jolly e quindi la stringa di argomenti può essere sia pubblicata che sottoscritta.

Esempi

```
IBM/+/Results
#/Results
IBM/Software/Results
```

Schema di caratteri jolly basato su caratteri
Lo schema di caratteri jolly basato sui caratteri consente di selezionare argomenti basati sulla corrispondenza dei caratteri tradizionali.

È possibile selezionare tutti gli argomenti a più livelli in una gerarchia di argomenti utilizzando la stringa `'*'`. L'utilizzo di `'*'` nello schema di caratteri jolly basato sui caratteri equivale all'utilizzo della stringa di caratteri jolly basata sull'argomento `'#'`

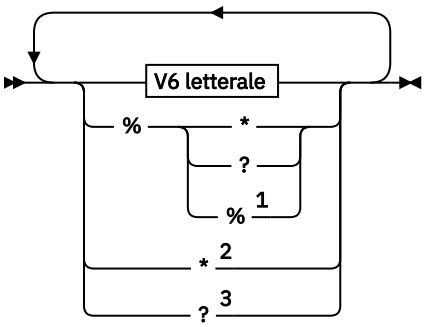
`' x/*/y '` è equivalente a `' x/#/y '` nello schema basato sull'argomento e seleziona tutti gli argomenti nella gerarchia degli argomenti tra i livelli `' x` e `y '`, dove `' x '` e `' y '` sono nomi di argomenti che non si trovano nella serie di livelli restituiti dal carattere jolly.

`'/+/'` nello schema basato sull'argomenti non dispone di un equivalente esatto nello schema basato sui caratteri. `'IBM/*/Results'` seleziona anche `'IBM/Patents/Software/Results'`. Solo se la serie di nomi argomento a ciascun livello della gerarchia è univoca, è sempre possibile costruire query con i due schemi che producono corrispondenze identiche.

Utilizzati in modo generale, `'*'` e `'?'` nello schema basato su caratteri non hanno equivalenti nello schema basato sugli argomenti. Lo schema basato sull'argomento non esegue una corrispondenza parziale utilizzando caratteri jolly. La sottoscrizione del carattere jolly basata sui caratteri `'IBM/*ware/Results'` non ha un equivalente basato sull'argomento.

Nota: Le corrispondenze che utilizzano sottoscrizioni con caratteri jolly sono più lente delle corrispondenze che utilizzano sottoscrizioni basate su argomenti.

Stringa di caratteri jolly basata su caratteri



V6 letterale

Note:

[1] Indica l'escape del seguente carattere, in modo che venga considerato come un letterale. `'%'` deve essere seguito da `'*'`, `'?'` o `'%'`. Consultare “Esempi di stringhe di argomento” a pagina 71.
[2] Indica la corrispondenza di zero o più caratteri in una sottoscrizione.
[3] Indica che corrisponde esattamente a un carattere in una sottoscrizione.

Esempi

```
IBM/*/Results
IBM/*ware/Results
```

Combinazione di stringhe di argomenti

Quando si creano sottoscrizioni o si aprono argomenti in modo da poter pubblicare messaggi, la stringa di argomenti può essere formata combinando due stringhe di argomenti secondari separate o "argomenti secondari". Un argomento secondario viene fornito dall'applicazione o dal comando di gestione come stringa di argomenti e l'altro è la stringa di argomenti associata a un oggetto argomento. È possibile

utilizzare un argomento secondario come stringa di argomenti da solo o combinarli per formare un nuovo nome di argomento.

Ad esempio, quando si definisce una sottoscrizione utilizzando il comando MQSC **DEFINE SUB**, il comando può utilizzare **TOPICSTR** (stringa argomento) o **TOPICOBJ** (oggetto argomento) come attributo o entrambi insieme. Se viene fornito solo **TOPICOBJ** , la stringa argomento associata a tale oggetto argomento viene utilizzata come stringa argomento. Se viene fornito solo **TOPICSTR** , viene utilizzato come stringa di argomenti. Se vengono forniti entrambi, vengono concatenati per formare una singola stringa di argomenti nel modulo **TOPICOBJ** / **TOPICSTR**, dove la stringa di argomenti configurata **TOPICOBJ** è sempre la prima e le due parti della stringa sono sempre separate da un carattere "/" .

Allo stesso modo, in un programma MQI il nome completo dell'argomento viene creato da MQOPEN. Si compone di due campi utilizzati nelle chiamate MQI di pubblicazione / sottoscrizione, nell'ordine elencato:

1. L'attributo **TOPICSTR** dell'oggetto argomento, denominato nel campo **ObjectName** .
2. Il parametro **ObjectString** che definisce l'argomento secondario fornito dall'applicazione.

La stringa di argomenti risultante viene restituita nel parametro **ResObjectString** .

Questi campi vengono considerati come presenti se il primo carattere di ogni campo non è uno spazio vuoto o un carattere null e la lunghezza del campo è maggiore di zero. Se è presente solo uno dei campi, viene utilizzato non modificato come nome dell'argomento. Se nessuno dei due campi ha un valore, la chiamata ha esito negativo con codice di errore MQRC_UNKNOWN_OBJECT_NAMEo MQRC_TOPIC_STRING_ERROR se il nome completo dell'argomento non è valido.

Se sono presenti entrambi i campi, viene inserito un carattere "/" tra i due elementi del nome argomento combinato risultante.

La seguente tabella mostra esempi di concatenazione di stringhe di argomenti:

Tabella 2. Esempi di concatenazione di stringhe di argomenti

TOPICSTR dell'oggetto argomento	**Stringa argomento fornita dall'applicazione o dal comando DEFINE SUB**	**Nome argomento completo**	**Commento**
Calcio / Punteggi	' '	Calcio / Punteggi	Il TOPICSTR dell'oggetto argomento viene utilizzato da solo.
' '	Calcio / Punteggi	Calcio / Punteggi	ObjectString/TOPICSTR viene utilizzato da solo.
football	Punteggi	Calcio / Punteggi	Un carattere "/" viene aggiunto al punto di concatenazione.
football	/Punteggi	Calcio / /Punteggi	Viene prodotto un 'nodo vuoto ' tra le due stringhe. È diverso da "Calcio / Punteggi".
/Calcio	Punteggi	/Calcio / Punteggi	L'argomento inizia con un 'nodo vuoto '. È diverso da "Calcio / Punteggi".

Il carattere "/" viene considerato come un carattere speciale, fornendo una struttura al nome completo dell'argomento in“Strutture ad albero degli argomenti” a pagina 77. Il carattere "/" non deve essere utilizzato per altri motivi, poiché la struttura della struttura ad albero degli argomenti è interessata. L'argomento "/Football" non è uguale all'argomento "Football".

Nota: Se si utilizza un oggetto argomento durante la creazione di una sottoscrizione, il valore della stringa di argomenti dell'oggetto argomento viene fissato nella sottoscrizione al momento della definizione. Qualsiasi modifica successiva all'oggetto argomento non influisce sulla stringa argomento per cui è definita la sottoscrizione.

Caratteri jolly nelle stringhe di argomenti

I seguenti caratteri jolly sono caratteri speciali:

- segno più (+)
- segno numerico (#)
- asterisco (*)
- punto interrogativo (?)

I caratteri jolly hanno un significato speciale solo se utilizzati da una sottoscrizione. Questi caratteri non sono considerati non validi quando vengono utilizzati altrove, tuttavia è necessario accertarsi di comprendere come vengono utilizzati e si potrebbe preferire di non utilizzare questi caratteri nelle stringhe argomento durante la pubblicazione o la definizione di oggetti argomento.

Se si esegue la pubblicazione su una stringa di argomenti con # o + combinati con altri caratteri (inclusi se stessi) all'interno di un livello di argomenti, la stringa di argomenti può essere sottoscritta con uno schema di caratteri jolly.

Se si pubblica su una stringa di argomenti con # o + come unico carattere compreso tra due / caratteri, la stringa di argomenti non può essere sottoscritta esplicitamente da un'applicazione utilizzando lo schema di caratteri jolly MQSO_WILDCARD_TOPIC. Questa situazione fa sì che l'applicazione ottena un numero di pubblicazioni superiore a quello previsto.

Non utilizzare un carattere jolly nella stringa dell'argomento di un oggetto argomento definito. In questo caso, il carattere viene considerato come un carattere letterale quando l'oggetto viene utilizzato da un publisher e come un carattere jolly quando viene utilizzato da una sottoscrizione. Ciò può portare a confusione.

Frammento di codice di esempio

Questo frammento di codice, estratto dal programma di esempio Esempio 2: Publisher su un argomento variabile, combina un oggetto argomento con una stringa argomento variabile:

```
MQOD   td = {MQOD_DEFAULT}; /* Object Descriptor      */
td.ObjectType = MQOT_TOPIC;  /* Object is a topic      */
td.Version = MQOD_VERSION_4;  /* Descriptor needs to be V4  */
strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
td.ObjectString.VSPtr = topicString;
td.ObjectString.VSLength = (MQLONG)strlen(topicString);
td.ResObjectString.VSPtr = resTopicStr;
td.ResObjectString.VSBufSize = sizeof(resTopicStr)-1;
MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF_QUIESCING, &Hobj, &CompCode, &Reason);
```

Strutture ad albero degli argomenti

Ciascun argomento che viene definito è un elemento, o nodo, della struttura ad albero degli argomenti. La struttura ad albero degli argomenti può essere vuota per iniziare o contenere argomenti che sono stati precedentemente definiti utilizzando i comandi MQSC o PCF. È possibile definire un nuovo argomento utilizzando i comandi di creazione degli argomenti o specificando l'argomento per la prima volta in una pubblicazione o in una sottoscrizione.

Sebbene sia possibile utilizzare qualsiasi stringa di caratteri per definire una stringa di argomenti, è consigliabile scegliere una stringa di argomenti che si adatti a una struttura ad albero gerarchica. Una progettazione accurata delle punte e delle strutture ad albero degli argomenti può essere utile per le seguenti operazioni:

- La sottoscrizione a più argomenti.

- Stabilire criteri di sicurezza.

Sebbene sia possibile costruire una struttura ad albero degli argomenti come una struttura lineare, è meglio crearne una in una struttura gerarchica con uno o più argomenti root. Per ulteriori informazioni sulla pianificazione della protezione e sugli argomenti, consultare Sicurezza di pubblicazione / sottoscrizione.

Figura 18 a pagina 78 mostra un esempio di una struttura ad albero di argomenti con un argomento root.

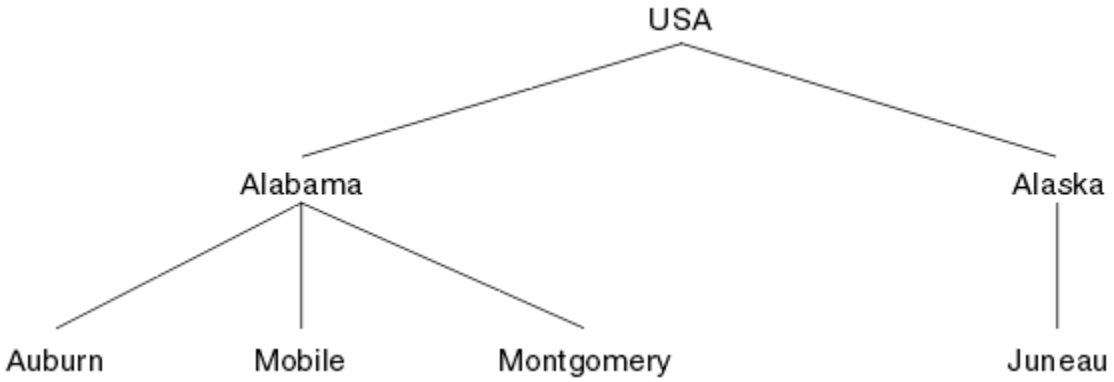


Figura 18. Esempio di struttura ad albero degli argomenti

Ciascuna stringa di caratteri nella figura rappresenta un nodo nella struttura ad albero degli argomenti. Una stringa di argomenti completa viene creata aggregando i nodi da uno o più livelli nella struttura di argomenti. I livelli sono separati dal carattere "/". Il formato di una stringa di argomenti completamente specificata è:root/level2/level3".

Gli argomenti validi nella struttura ad albero degli argomenti mostrata in Figura 18 a pagina 78 sono:

"USA"
"USA/Alabama"
"USA/Alaska"
"USA/Alabama/Auburn"
"USA/Alabama/Mobile"
"USA/Alabama/Montgomery"
"USA/Alaska/Juneau"

Quando si progettano le stringhe di argomenti e le strutture ad albero degli argomenti, tenere presente che il gestore code non interpreta o tenta di derivare il significato dalla stringa di argomenti stessa. Utilizza semplicemente la stringa di argomenti per inviare messaggi selezionati ai sottoscrittori di tale argomento.

Gli argomenti riportati di seguito sono validi per la costruzione e il contenuto di una struttura ad albero dell'argomento:

- Non esistono limiti relativi al numero di livelli di una struttura ad albero dell'argomento.
- Non esistono limiti relativi alla lunghezza del nome di un livello in una struttura ad albero dell'argomento.
- Può essere presente un numero qualsiasi di nodi "root"; ovvero, può essere presente un numero qualsiasi di strutture ad albero degli argomenti.

Attività correlate

Riduzione del numero di argomenti indesiderati nella struttura ad albero degli argomenti

Oggetti argomento di gestione

Utilizzando un oggetto argomento di amministrazione, è possibile assegnare specifici attributi non predefiniti agli argomenti.

Figura 19 a pagina 79 mostra in che modo un argomento di alto livello di `Sport` suddiviso in argomenti separati che coprono diversi sport può essere visualizzato come una struttura di argomenti:

Figura 19. Visualizzazione di una struttura ad albero degli argomenti

Figura 20 a pagina 79 mostra come la struttura ad albero degli argomenti può essere ulteriormente divisa, per separare diversi tipi di informazioni su ogni sport:

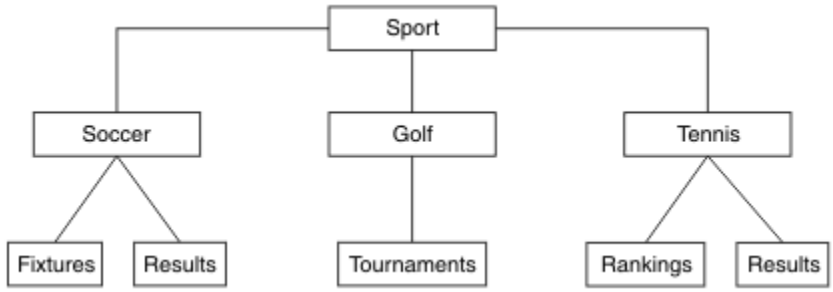


Figura 20. Struttura ad albero degli argomenti estesa

Per creare la struttura di argomenti illustrata, non è necessario definire alcun oggetto di argomento di gestione. Ciascuno dei nodi in questa struttura ad albero è definito da una stringa di argomenti creata in un'operazione di pubblicazione o sottoscrizione. Ogni argomento nella struttura ad albero eredita i propri attributi dal relativo parent. Gli attributi vengono ereditati dall'oggetto argomento principale, perché per impostazione predefinita tutti gli attributi sono impostati su ASPARENT. In questo esempio, ogni argomento ha gli stessi attributi dell'argomento `Sport` . L'argomento `Sport` non ha alcun oggetto argomento di gestione ed eredita i relativi attributi da SYSTEM.BASE.TOPIC.

Notare che non è consigliabile fornire autorizzazioni per gli utenti non mqm sul nodo root della struttura ad albero degli argomenti, che è SYSTEM.BASE.TOPIC, poiché le autorizzazioni sono ereditate ma non possono essere limitate. Pertanto, fornendo le autorizzazioni a questo livello, si forniscono le autorizzazioni all'intero albero. È necessario concedere l'autorità ad un livello di argomento inferiore nella gerarchia.

Gli oggetti argomento di gestione possono essere utilizzati per definire attributi specifici per particolari nodi nella struttura ad albero degli argomenti. Nel seguente esempio, l'oggetto argomento di gestione viene definito per impostare la proprietà delle sottoscrizioni durevoli DURSUB dell'argomento soccer sul valore NO:

```
DEFINE TOPIC(FOOTBALL.EUROPEAN)
TOPICSTR('Sport/Soccer')
DURSUB(NO)
DESCR('Administrative topic object to disallow durable subscriptions')
```

La struttura ad albero degli argomenti può ora essere visualizzata come:

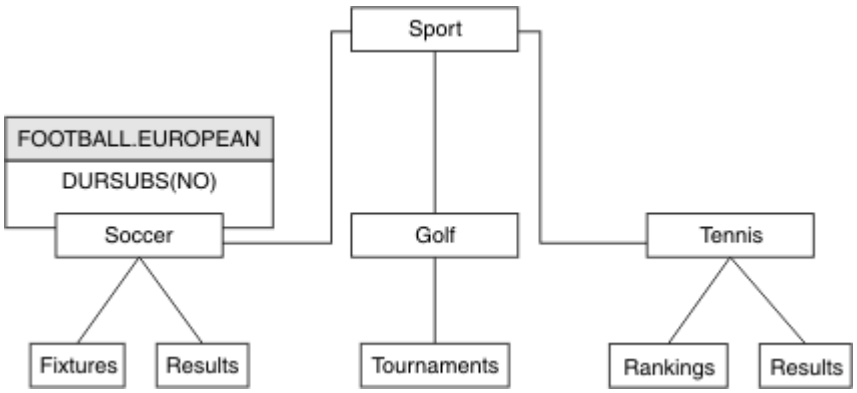


Figura 21. Visualizzazione di un oggetto di argomento amministrativo associato all'argomento Sport / Soccer

Le applicazioni che sottoscrivono gli argomenti sotto Soccer nella struttura ad albero possono ancora utilizzare le stringhe degli argomenti utilizzate prima dell'aggiunta dell'oggetto argomento di gestione. Tuttavia, è ora possibile scrivere un'applicazione per la sottoscrizione utilizzando il nome oggetto `FOOTBALL.EUROPEAN`, invece della stringa `/Sport/Soccer`. Ad esempio, per sottoscrivere `/Sport/Soccer/Results`, un'applicazione può specificare `MQSD.ObjectName` come `FOOTBALL.EUROPEAN` e `MQSD.ObjectString` come `Results`.

Con questa funzione, è possibile nascondere parte della struttura ad albero degli argomenti agli sviluppatori dell'applicazione. Definire un oggetto argomento di gestione in un nodo particolare nella struttura ad albero degli argomenti, quindi gli sviluppatori dell'applicazione possono definire i propri argomenti come elementi secondari del nodo. Gli sviluppatori devono conoscere l'argomento principale, ma non tutti gli altri nodi nella struttura ad albero principale.

Eredità di attributi

Se una struttura ad albero degli argomenti dispone di molti oggetti argomento di gestione, ciascun oggetto argomento di gestione, per impostazione predefinita, eredita i propri attributi dall'argomento di gestione principale più vicino. L'esempio precedente è stato esteso in Figura 22 a pagina 80:

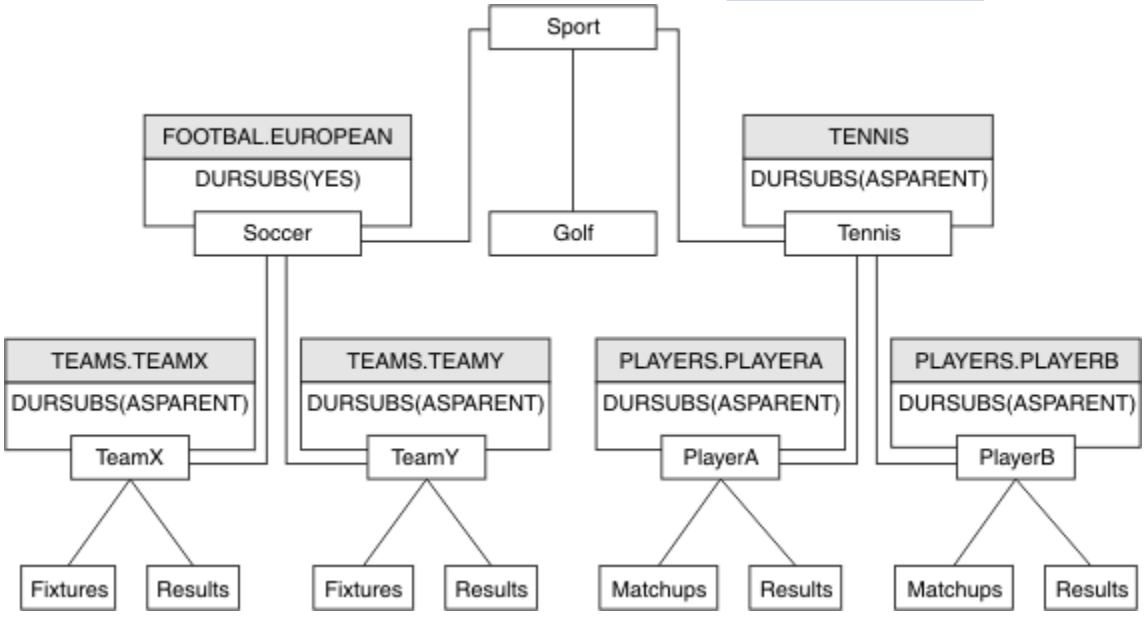


Figura 22. Struttura ad albero degli argomenti con diversi oggetti argomento di gestione

Ad esempio, utilizzare l'eredità per fornire a tutti gli argomenti secondari di `/Sport/Soccer` la proprietà che le sottoscrizioni non sono durevoli. Modificare l'attributo `DURSUB` di `FOOTBALL.EUROPEAN` in `NO`.

Questo attributo può essere impostato utilizzando il comando seguente:

```
ALTER TOPIC(FOOTBALL.EUROPEAN) DURSUB(NO)
```

Tutti gli oggetti argomento di gestione degli argomenti secondari di `Sport/Soccer` hanno la proprietà DURSUB impostata sul valore predefinito ASPARENT. Dopo aver modificato il valore della proprietà DURSUB di `FOOTBALL.EUROPEAN` in `NO`, gli argomenti secondari di `Sport/Soccer` ereditano il DURSUB valore della proprietà `NO`. Tutti gli argomenti child di `Sport/Tennis` ereditano il valore DURSUB dall'oggetto SYSTEM.BASE.TOPIC . SYSTEM.BASE.TOPIC ha il valore YES.

Il tentativo di effettuare una sottoscrizione durevole all'argomento `Sport/Soccer/TeamX/Results` non è riuscito; tuttavia, il tentativo di effettuare una sottoscrizione durevole a `Sport/Tennis/PlayerB/Results` avrà esito positivo.

Controllo dell'utilizzo dei caratteri jolly con la proprietà WILDCARD

Utilizzare la proprietà MQSC **Topic** WILDCARD o la proprietà Argomento `WildcardOperation` PCF equivalente per controllare la consegna di pubblicazioni alle applicazioni del sottoscrittore che utilizzano nomi stringa di argomenti con caratteri jolly. La proprietà WILDCARD può avere uno dei due valori possibili:

WILDCARD
Il funzionamento delle sottoscrizioni con caratteri jolly rispetto a questo argomento.

PASSTHRU
Le sottoscrizioni effettuate a un argomento con carattere jolly meno specifico della stringa argomento in questo oggetto argomento riceveranno le pubblicazioni relative a questo argomento e a stringhe argomento più specifiche di tale argomento.

BLOCK
Le sottoscrizioni effettuate a un argomento con carattere jolly meno specifico della stringa argomento in questo oggetto argomento non riceveranno le pubblicazioni relative a questo argomento o a stringhe argomento più specifiche di tale argomento.

Il valore di questo attributo è utilizzato quando vengono definite sottoscrizioni. Se si modifica questo attributo, la serie di argomenti trattati dalle sottoscrizioni esistenti non viene interessata dalla modifica. Questo scenario si applica anche se la topologia cambia quando si creano o eliminano oggetti argomento; la serie di argomenti che corrispondono alle sottoscrizioni create in seguito alla modifica dell'attributo WILDCARD viene creata utilizzando la topologia modificata. Se si desidera forzare una rivalutazione della serie corrispondente di argomenti per le sottoscrizioni esistenti, è necessario riavviare il gestore code.

Nell'esempio, “Esempio: creazione del cluster di pubblicazione / sottoscrizione Sport” a pagina 85, è possibile seguire la procedura per creare la struttura ad albero dell'argomento mostrata in Figura 23 a pagina 82.

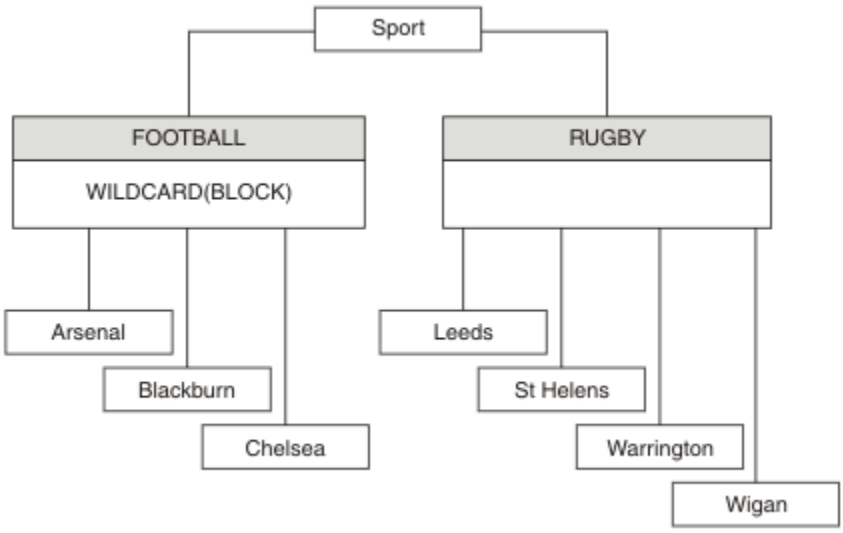


Figura 23. Una struttura di argomenti che utilizza la proprietà WILDCARD , BLOCK

Un sottoscrittore che utilizza la stringa di argomenti con caratteri jolly # riceve tutte le pubblicazioni nell'argomento `Sport` e nella struttura ad albero secondaria `Sport/Rugby` . Il sottoscrittore non riceve alcuna pubblicazione nella struttura secondaria `Sport/Football` , poiché il valore della proprietà `WILDCARD` dell'argomento `Sport/Football` è `BLOCK`.

`PASSTHRU` è l'impostazione predefinita. È possibile impostare il valore della proprietà `WILDCARD` `PASSTHRU` sui nodi nella struttura ad albero `Sport` . Se i nodi non hanno il valore della proprietà `WILDCARD` `BLOCK`, l'impostazione `PASSTHRU` non modifica il funzionamento osservato dai sottoscrittori ai nodi nella albero `Sports` .

Nell'esempio, creare sottoscrizioni per vedere in che modo l'impostazione del carattere jolly influisce sulle pubblicazioni fornite; consultare Figura 27 a pagina 87. Eseguire il comando publish in Figura 30 a pagina 88 per creare alcune pubblicazioni.

```
pub QMA
```

Figura 24. Pubblica in QMA

I risultati sono riportati in Tabella 3 a pagina 82. Si noti come l'impostazione del valore della proprietà `WILDCARD` `BLOCK`impedisca alle sottoscrizioni con caratteri jolly di ricevere pubblicazioni per argomenti nell'ambito del carattere jolly.

Tabella 3. Pubblicazioni ricevute su QMA

Sottoscrizione	**Stringa argomento**	**Pubblicazioni ricevute**	**Note**

SPORTS	`Sports/#`	`Sports`
		`Sports/Rugby`
		`Sports/Rugby/Leeds`
			Tutte le pubblicazioni della sottostruttura Football bloccate da `WILDCARD(BLOCK)` su `Sports/Football`

SARSENAL	`Sports/#/Arsenal`	-	`WILDCARD(BLOCK)` on `Sports/Football` impedisce la sottoscrizione di caratteri jolly su `Arsenal`

Tabella 3. Pubblicazioni ricevute su QMA (Continua)			
Sottoscrizione	**Stringa argomento**	**Pubblicazioni ricevute**	**Note**
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Il valore predefinito WILDCARD su Sports / Rugby non impedisce la sottoscrizione di caratteri jolly su Leeds.

Nota:

Si supponga che una sottoscrizione abbia un carattere jolly che corrisponde a un oggetto argomento con il valore della proprietà WILDCARD BLOCK. Se la sottoscrizione ha anche una stringa di argomenti a destra del carattere jolly corrispondente, la sottoscrizione non riceverà mai una pubblicazione. La serie di pubblicazioni che non sono bloccate sono pubblicazioni per argomenti che sono parent del carattere jolly bloccato. Le pubblicazioni per gli argomenti child dell'argomento con il valore della proprietà BLOCK sono bloccate dal carattere jolly. Pertanto, le stringhe di argomenti di sottoscrizione che includono un argomento alla destra del carattere jolly non ricevono mai alcuna pubblicazione corrispondente.

L'impostazione del valore della proprietà WILDCARD su BLOCK non significa che non è possibile effettuare la sottoscrizione utilizzando una stringa di argomenti che include caratteri jolly. Tale sottoscrizione è normale. La sottoscrizione dispone di un argomento esplicito che corrisponde all'argomento con un oggetto argomento con un valore della proprietà WILDCARD BLOCK. Utilizza i caratteri jolly per gli argomenti che sono parent o child dell'argomento con il valore della proprietà WILDCARD BLOCK. Nell'esempio in Figura 23 a pagina 82, una sottoscrizione come Sports/Football/# può ricevere pubblicazioni.

Caratteri jolly e argomenti cluster

Le definizioni degli argomenti del cluster vengono propagati a ogni gestore code in un cluster. Una sottoscrizione a un argomento cluster in un gestore code in un cluster determina la creazione di sottoscrizioni proxy da parte del gestore code. Una sottoscrizione proxy viene creata su ogni altro gestore code nel cluster. Le sottoscrizioni che utilizzano stringhe di argomenti contenenti caratteri jolly, combinate con argomenti cluster, possono fornire un comportamento difficile da prevedere. Il comportamento viene spiegato nel seguente esempio.

Nel cluster configurato per l'esempio, “Esempio: creazione del cluster di pubblicazione / sottoscrizione Sport” a pagina 85, QMB ha la stessa serie di sottoscrizioni di QMA, ma QMB non ha ricevuto alcuna pubblicazione dopo che il publisher è stato pubblicato in QMA, consultare Figura 24 a pagina 82. Anche se gli argomenti Sports/Football e Sports/Rugby sono argomenti del cluster, le sottoscrizioni definite in fullsubs.tst non fanno riferimento a un argomento del cluster. Nessuna sottoscrizione proxy viene propagata da QMB a QMA. Senza sottoscrizioni proxy, nessuna pubblicazione per QMA viene inoltrata a QMB.

Alcune sottoscrizioni, come Sports/#/Leeds, potrebbero far riferimento a un argomento cluster, in questo caso Sports/Rugby. La sottoscrizione Sports/#/Leeds viene effettivamente risolta nell'oggetto argomento SYSTEM.BASE.TOPIC.

La regola per risolvere l'oggetto argomento a cui fa riferimento una sottoscrizione come, Sports/#/Leeds è la seguente. Troncare la stringa argomento al primo carattere jolly. Eseguire la scansione a sinistra nella stringa dell'argomento cercando il primo argomento a cui è associato un oggetto argomento di gestione. L'oggetto argomento potrebbe specificare un nome cluster o definire un oggetto argomento locale. Nell'esempio, Sports/#/Leeds, la stringa di argomenti dopo il troncamento è Sports, che non ha alcun oggetto argomento e quindi Sports/#/Leeds eredita da SYSTEM.BASE.TOPIC, che è un oggetto argomento locale.

Per vedere come la sottoscrizione agli argomenti del cluster può modificare il funzionamento della propagazione dei caratteri jolly, eseguire lo script batch upsubs.bat. Lo script elimina le code di sottoscrizione e aggiunge le sottoscrizioni dell'argomento del cluster in fullsubs.tst. Eseguire nuovamente puba.bat per creare un batch di pubblicazioni; consultare Figura 24 a pagina 82.

Tabella 4 a pagina 84 mostra il risultato dell'aggiunta di due nuove sottoscrizioni allo stesso gestore code su cui sono state pubblicate le pubblicazioni. Il risultato è quello previsto, le nuove sottoscrizioni ricevono una pubblicazione ciascuna e il numero di pubblicazioni ricevute dalle altre sottoscrizioni è invariato. I risultati imprevisti si verificano sull'altro gestore code cluster; consultare Tabella 5 a pagina 84.

Tabella 4. Pubblicazioni ricevute su QMA

Sottoscrizione	**Stringa argomento**	**Pubblicazioni ricevute**	**Note**
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Tutte le pubblicazioni della sottostruttura Football bloccate da WILDCARD(BLOCK) su Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD(BLOCK) on Sports/Football impedisce la sottoscrizione di caratteri jolly su Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Il valore predefinito WILDCARD su Sports / Rugby non impedisce la sottoscrizione di caratteri jolly su Leeds.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal riceve una pubblicazione perché la sottoscrizione non ha un carattere jolly.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds riceverà una pubblicazione in ogni caso.

Tabella 5 a pagina 84 mostra i risultati dell'aggiunta delle due nuove sottoscrizioni su QMB e della pubblicazione su QMA. Si ricordi che QMB non ha ricevuto alcuna pubblicazione senza queste due nuove sottoscrizioni. Come previsto, le due nuove sottoscrizioni ricevono le pubblicazioni, poiché Sports/FootBall e Sports/Rugby sono entrambi argomenti cluster. QMB ha inoltrato le sottoscrizioni proxy per Sports/Football/Arsenal e Sports/Rugby/Leeds a QMA, che ha quindi inviato le pubblicazioni a QMB.

Il risultato imprevisto è che le due sottoscrizioni Sports/# e Sports/#/Leeds che in precedenza non avevano ricevuto alcuna pubblicazione, ora ricevono le pubblicazioni. Il motivo è che le pubblicazioni Sports/Football/Arsenal e Sports/Rugby/Leeds inoltrate a QMB per le altre sottoscrizioni sono ora disponibili per tutti i sottoscrittori collegati a QMB. Di conseguenza, le sottoscrizioni agli argomenti locali Sports/# e Sports/#/Leeds ricevono la pubblicazione Sports/Rugby/Leeds . Sports/#/Arsenal continua a non ricevere una pubblicazione, poiché Sport / Calcio ha il valore della proprietà WILDCARD impostato su BLOCK.

Tabella 5. Pubblicazioni ricevute su QMB

Sottoscrizione	**Stringa argomento**	**Pubblicazioni ricevute**	**Note**
SPORTS	*Sports/#*	*Sports/Rugby/Leeds*	*Tutte le pubblicazioni nell'albero secondario Calcio bloccate da WILDCARD(BLOCK) su Sports/Football*

Tabella 5. Pubblicazioni ricevute su QMB (Continua)			
Sottoscrizione	**Stringa argomento**	**Pubblicazioni ricevute**	**Note**
SARSENAL	Sports/#/Arsenal	-	WILDCARD(BLOCK) on Sports/Football impedisce la sottoscrizione di caratteri jolly su Arsenal
SLEEDS	*Sports/#/Leeds*	*Sports/Rugby/Leeds*	*Predefinito WILDCARD su Sports / Rugby non impedisce la sottoscrizione di caratteri jolly su Leeds.*
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal riceve una pubblicazione perché la sottoscrizione non ha un carattere jolly.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds riceverà una pubblicazione in ogni caso.

Nella maggior parte delle applicazioni, non è desiderabile che una sottoscrizione influenzi il comportamento di un'altra sottoscrizione. Un utilizzo importante della proprietà WILDCARD con valore BLOCK consiste nel rendere le sottoscrizioni alla stessa stringa di argomenti contenenti caratteri jolly uniformi. Se la sottoscrizione si trova sullo stesso gestore code del publisher o su un gestore code diverso, i risultati della sottoscrizione sono gli stessi.

Caratteri jolly e stream

Per una nuova applicazione scritta nell'API di pubblicazione / sottoscrizione, l'effetto è che una sottoscrizione a * non riceve alcuna pubblicazione. Per ricevere tutte le pubblicazioni Sport, è necessario sottoscrivere Sports/*o Sports/#e in modo simile per le pubblicazioni Business .

Il comportamento di un'applicazione di pubblicazione / sottoscrizione accodata esistente non cambia quando il Broker di pubblicazione / sottoscrizione viene migrato a una versione successiva di IBM MQ. La proprietà **StreamName** nei comandi **Publish**, **Register Publisher**o **Subscriber** è associata al nome dell'argomento in cui è stato migrato il flusso.

Caratteri jolly e punti di sottoscrizione

Per una nuova applicazione scritta nell'API di pubblicazione / sottoscrizione, l'effetto della migrazione è che una sottoscrizione a * non riceve alcuna pubblicazione. Per ricevere tutte le pubblicazioni Sport, è necessario sottoscrivere Sports/*o Sports/#e in modo simile per le pubblicazioni Business .

Il comportamento di un'applicazione di pubblicazione / sottoscrizione accodata esistente non cambia quando il Broker di pubblicazione / sottoscrizione viene migrato a una versione successiva di IBM MQ. La proprietà **SubPoint** nei comandi **Publish**, **Register Publisher**o **Subscriber** è associata al nome dell'argomento a cui è stata migrata la sottoscrizione.

Esempio: creazione del cluster di pubblicazione / sottoscrizione Sport

I passi che seguono creano un cluster, CL1, con quattro gestori code: due repository completi, CL1A e CL1B, e due repository parziali, QMA e QMB. I repository completi vengono utilizzati per contenere solo le definizioni cluster. QMA è designato come host argomento del cluster. Le sottoscrizioni durevoli sono definite sia su QMA che su QMB.

Nota: L'esempio è codificato per Windows. È necessario ricodificare Crea qmgrs.bat e creare pub.bat per configurare e verificare l'esempio su altre piattaforme.

1. Creare i file script.

a. Crea topics.tst
b. Crea wildsubs.tst
c. Crea fullsubs.tst
d. Crea qmgrs.bat
e. create pub.bat

2. Eseguire Create qmgrs.bat per creare la configurazione.

```
qmgrs
```

Creare gli argomenti in Figura 23 a pagina 82. Lo script nella figura 5 crea gli argomenti del cluster `Sports/Football` e `Sports/Rugby`.

Nota: L'opzione REPLACE non sostituisce le proprietà TOPICSTR di un argomento. TOPICSTR è una proprietà che viene utilmente variata nell'esempio per testare diverse strutture ad albero degli argomenti. Per modificare gli argomenti, eliminare prima l'argomento.

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports')   TOPICSTR('Sports')
DEFINE TOPIC ('Football')  TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal')  TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea')  TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby')   TOPICSTR('Sports/Rugby')  CLUSTER(CL1)
DEFINE TOPIC ('Leeds')   TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan')   TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

Figura 25. Eliminare e creare argomenti: topics.tst

Nota: Eliminare gli argomenti, poiché REPLACE non sostituisce le stringhe di argomenti.

Creare sottoscrizioni con caratteri jolly. I caratteri jolly corrispondenti agli argomenti con gli oggetti argomento in Figura 23 a pagina 82. Creare una coda per ogni sottoscrizione. Le code vengono cancellate e le sottoscrizioni vengono eliminate quando lo script viene eseguito o rieseguito.

Nota: L'opzione REPLACE non sostituisce le proprietà TOPICOBJ o TOPICSTR di una sottoscrizione. TOPICOBJ o TOPICSTR sono le proprietà che sono utilmente variate nell'esempio per testare sottoscrizioni differenti. Per modificarli, eliminare prima la sottoscrizione.

```
DEFINE QLOCAL(QSPORTS)  REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS)  REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB  (SPORTS)
DELETE SUB  (SARSENAL)
DELETE SUB  (SLEEDS)
DEFINE SUB  (SPORTS)  TOPICSTR('Sports/#')     DEST(QSPORTS)
DEFINE SUB  (SARSENAL) TOPICSTR('Sports/+/Arsenal') DEST(QSARSENAL)
DEFINE SUB  (SLEEDS)  TOPICSTR('Sports/+/Leeds')  DEST(QSLEEDS)
```

Figura 26. Creare sottoscrizioni jolly: wildsubs.tst

Creare sottoscrizioni che facciano riferimento agli oggetti argomento del cluster.

Nota:

Il delimitatore, /, viene automaticamente inserito tra la stringa argomento a cui fa riferimento TOPICOBJe la stringa argomento definita da TOPICSTR.

La definizione DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) crea la stessa sottoscrizione. TOPICOBJ viene utilizzato come un modo rapido per fare riferimento alla stringa di argomenti già definita. La sottoscrizione, una volta creata, non fa più riferimento all'oggetto argomento.

```
DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS)  REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB  (FARSENAL)
DELETE SUB  (RLEEDS)
DEFINE SUB  (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB  (RLEEDS)  TOPICOBJ('Rugby')  TOPICSTR('Leeds')  DEST(QRLEEDS)
```

Figura 27. Eliminare e creare sottoscrizioni: fullsubs.tst

Creare un cluster con due repository. Creare due repository parziali per la pubblicazione e la sottoscrizione. Eseguire nuovamente lo script per eliminare tutto e ricominciare. Lo script crea anche la gerarchia degli argomenti e le sottoscrizioni dei caratteri jolly iniziali.

Nota:

Su altre piattaforme, scrivere uno script simile oppure immettere tutti i comandi. L'uso di uno script rende veloce l'eliminazione di tutto e ricomincia con una configurazione identica.

```
@echo off
set port.CL1B=1421
set port.CLIA=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof
```

Figura 28. Creare gestori code: qmgrs.bat

Aggiorna la configurazione aggiungendo le sottoscrizioni agli argomenti del cluster.

```
@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst
```

Figura 29. Aggiorna sottoscrizioni: upsubs.bat

Eseguire pub.bat, con un gestore code come parametro, per pubblicare i messaggi contenenti la stringa di argomenti di pubblicazione. Pub.bat utilizza il programma di esempio **amqspub**.

```
@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1
```

Figura 30. Pubblica: pub.bat

Stream e argomenti

La pubblicazione / sottoscrizione accodata ha il concetto di un flusso di pubblicazione che non esiste nel modello di pubblicazione / sottoscrizione integrato. Nella pubblicazione / sottoscrizione accodata, i flussi forniscono un metodo per separare il flusso di informazioni per argomenti differenti. Un flusso viene implementato come un argomento di livello superiore che può essere associato a un diverso identificativo dell'argomento dal punto di vista amministrativo.

Il flusso predefinito SYSTEM.BROKER.DEFAULT.STREAM viene impostato automaticamente per tutti i broker e i gestori code su una rete e non è richiesta alcuna configurazione aggiuntiva per utilizzare il flusso predefinito. Considerare il flusso predefinito come uno spazio argomento predefinito senza nome. Gli argomenti pubblicati nel flusso predefinito sono disponibili immediatamente per tutti i gestori code connessi, con la pubblicazione / sottoscrizione in coda abilitata. I flussi denominati sono spazi argomento separati e denominati. Il flusso denominato deve essere definito su ciascun broker in cui viene utilizzato.

Se i publisher e i sottoscrittori si trovano su gestori code differenti, dopo che i broker sono connessi nella stessa gerarchia broker, non è richiesta alcuna ulteriore configurazione per le pubblicazioni e le sottoscrizioni per il flusso tra di loro. La stessa interoperabilità funziona anche al contrario.

Flussi denominati

Un solution designer, che utilizza il modello di programmazione di pubblicazione / sottoscrizione accodata, potrebbe decidere di inserire tutte le pubblicazioni sportive in un flusso denominato Sport. Affinché il flusso sia disponibile per un gestore code in esecuzione su IBM MQ con la pubblicazione / sottoscrizione in coda abilitata, il flusso deve essere aggiunto manualmente.

Le applicazioni di pubblicazione / sottoscrizione accodate che eseguono la sottoscrizione a Soccer/Results sul flusso Sport funzionano senza modifiche. Le applicazioni di pubblicazione / sottoscrizione integrate che sottoscrivono l'argomento Sport utilizzando MQSUBe fornendo la stringa di argomenti Soccer/Results ricevono anche le stesse pubblicazioni.

L'attività di aggiunta di un flusso è descritta nell'argomento Aggiunta di un flusso. Potrebbe essere necessario aggiungere gli stream manualmente per due motivi.

1. Si continua a sviluppare le applicazioni di pubblicazione / sottoscrizione accodate in esecuzione su gestori code di versione successiva, piuttosto che migrare le applicazioni all'interfaccia MQI di pubblicazione / sottoscrizione integrata.
2. L'associazione predefinita dei flussi agli argomenti porta a una "collisione" nello spazio argomento e le pubblicazioni su un flusso hanno la stessa stringa argomento delle pubblicazioni provenienti da altrove.

Autorizzazioni

Per impostazione predefinita, nella root della struttura ad albero degli argomenti sono presenti più oggetti argomento: SYSTEM.BASE.TOPIC, SYSTEM.BROKER.DEFAULT.STREAMe SYSTEM.BROKER.DEFAULT.SUBPOINT. Le autorità (ad esempio, per la pubblicazione o la sottoscrizione) sono determinate dalle autorità sul SYSTEM.BASE.TOPIC ; qualsiasi autorizzazione su SYSTEM.BROKER.DEFAULT.STREAM o SYSTEM.BROKER.DEFAULT.SUBPOINT viene ignorata. Se SYSTEM.BROKER.DEFAULT.STREAM o SYSTEM.BROKER.DEFAULT.SUBPOINT vengono eliminati e ricreati con una stringa di argomento non vuota, le autorizzazioni definite su tali oggetti vengono utilizzate allo stesso modo di un normale oggetto argomento.

Associazione tra stream e argomenti

Un flusso di pubblicazione / sottoscrizione accodato viene imitato in IBM MQ creando una coda e assegnandogli lo stesso nome del flusso. A volte la coda viene chiamata coda di flusso, perché è così che appare per le applicazioni di pubblicazione / sottoscrizione accodate. La coda viene identificata nel motore di pubblicazione / sottoscrizione aggiungendolo all'elenco nomi speciale denominato SYSTEM.QPUBSUB.QUEUE.NAMELIST. È possibile aggiungere tutti i flussi necessari, aggiungendo ulteriori code speciali all'elenco nomi. Infine, è necessario aggiungere argomenti, con gli stessi nomi dei flussi e le stesse stringhe di argomenti del nome del flusso, in modo da poter pubblicare e sottoscrivere gli argomenti.

Tuttavia, in circostanze eccezionali, è possibile fornire agli argomenti corrispondenti agli stream qualsiasi stringa di argomenti scelta quando si definiscono gli argomenti. Lo scopo della stringa di argomenti è fornire all'argomento un nome univoco nello spazio argomenti. In genere il nome del flusso serve perfettamente a questo scopo. A volte, un nome di flusso e un nome di argomento esistente sono in conflitto. Per risolvere il problema, scegliere un'altra stringa di argomenti per l'argomento associato al flusso. Scegliere una stringa di argomenti, assicurandosi che sia univoca.

La stringa dell'argomento definita nella definizione dell'argomento viene preceduta nel modo normale dalla stringa dell'argomento fornita dai publisher e dai sottoscrittori utilizzando le chiamate MQI MQOPEN o MQSUB . Le applicazioni che fanno riferimento ad argomenti che utilizzano oggetti argomento non sono influenzate dalla scelta della stringa argomento prefisso, motivo per cui è possibile scegliere qualsiasi stringa argomento che mantenga le pubblicazioni univoche nello spazio argomento.

La rimappatura di flussi differenti su argomenti differenti si basa sui prefissi utilizzati per le stringhe di argomenti che sono univoci, per separare completamente una serie di argomenti da un'altra. È necessario definire una convenzione di denominazione dell'argomento universale che sia rigidamente rispettata per il funzionamento dell'associazione.

In IBM MQ, si utilizza il meccanismo di prefissaggio per riassociare una stringa di argomenti ad un'altra posizione nello spazio argomenti.

Nota: Quando si elimina uno stream, eliminare prima tutte le sottoscrizioni nello stream. Questa azione è più importante se una delle sottoscrizioni proviene da altri broker nella gerarchia del broker.

Argomenti e punti di sottoscrizione

I punti di sottoscrizione denominati sono emulati da argomenti e oggetti argomento.

Per aggiungere i punti di sottoscrizione manualmente, consultare Aggiunta di un punto di sottoscrizione.

Punti di sottoscrizione in IBM MQ

IBM MQ associa i punti di sottoscrizione a diversi spazi argomento all'interno della struttura ad albero degli argomenti IBM MQ . Gli argomenti nei messaggi di comandi senza un punto di sottoscrizione vengono associati non modificati alla root della struttura ad albero degli argomenti IBM MQ ed ereditano le proprietà da SYSTEM.BASE.TOPIC.

I messaggi di comandi con un punto di sottoscrizione vengono elaborati utilizzando l'elenco di oggetti argomento in SYSTEM.QPUBSUB.SUBPOINT.NAMELIST. Il nome del punto di sottoscrizione nel messaggio di comando viene confrontato con la stringa di argomento per ciascuno degli oggetti

argomento nell'elenco. Se viene trovata una corrispondenza, il nome del punto di sottoscrizione viene anteposto, come un nodo di argomento, alla stringa di argomento. L'argomento eredita le proprietà dall'oggetto argomento associato trovato in SYSTEM.QPUBSUB.SUBPOINT.NAMELIST.

L'effetto dell'utilizzo dei punti di sottoscrizione è quello di creare uno spazio argomento separato per ciascun punto di sottoscrizione. Lo spazio argomento è instradato in un argomento che ha lo stesso nome del punto di sottoscrizione. Gli argomenti in ogni spazio argomento ereditano le proprietà dall'oggetto argomento con lo stesso nome del punto di sottoscrizione.

Tutte le proprietà non impostate nell'oggetto argomento corrispondente vengono ereditate, nel modo normale, da SYSTEM.BASE.TOPIC.

Le applicazioni di pubblicazione / sottoscrizione accodate esistenti, utilizzando le intestazioni del messaggio MQRFH2 , continuano a funzionare impostando la proprietà **SubPoint** nei messaggi di comando Publish o Register subscriber . Il punto di sottoscrizione viene combinato con la stringa di argomenti nel messaggio di comando e l'argomento risultante viene elaborato come qualsiasi altro.

Le applicazioni IBM MQ non sono interessate dai punti di sottoscrizione. Se un'applicazione utilizza un argomento che eredita le informazioni da uno degli oggetti argomento corrispondenti, tale applicazione interagisce con un'applicazione in coda utilizzando il punto di sottoscrizione corrispondente.

Esempio

Un WebSphere Message Broker esistente (ora noto come IBM Integration Bus) l'applicazione di pubblicazione / sottoscrizione migrata in IBM MQ ha creato due oggetti argomento, GBP e USD, con le corrispondenti stringhe argomento 'GBP' e 'USD'.

I publisher esistenti nell'argomento NYSE/IBM/SPOT, migrati per essere eseguiti su IBM MQ, che utilizzano il punto di sottoscrizione USD creano pubblicazioni sull'argomento USD/NYSE/IBM/SPOT. Analogamente, i sottoscrittori esistenti a NYSE/IBM/SPOT, utilizzando il punto di sottoscrizione USD creano sottoscrizioni a USD/NYSE/IBM/SPOT.

sottoscrivere il prezzo spot in dollari in un programma di pubblicazione / sottoscrizione IBM MQ chiamando MQSUB. Creare una sottoscrizione utilizzando l' USD oggetto argomento e la stringa argomento 'NYSE/IBM/SPOT', come illustrato nel frammento di codice ' C.

```
strncpy(sd.ObjectName, "USD", MQ_TOPIC_NAME_LENGTH);
sd.ObjectString.VSPtr = "NYSE/IBM/SPOT";
sd.ObjectString.VSLength = MQVS_NULL_TERMINATED;
MQSUB(Hconn, &sd, &Hobj, &Hsub, &CompCode, &Reason);
```

1. Impostare l'attributo CLUSTERDEGLI oggetti argomento USD e GBP sull'host argomento cluster.
2. Eliminare tutte le copie degli oggetti argomento USD e GBP su altri gestori code nel cluster.
3. Assicurarsi che USD e GBP siano definiti in SYSTEM.QPUBSUB.SUBPOINT.NAMELIST su ogni gestore code nel cluster.

Esempio di configurazione di pubblicazione / sottoscrizione di un singolo gestore code

Figura 31 a pagina 91 illustra una singola configurazione di pubblicazione / sottoscrizione del gestore code di base. L'esempio mostra la configurazione per un servizio di notizie, dove le informazioni sono disponibili dai publisher su diversi argomenti:

- L'editore 1 pubblica informazioni sui risultati sportivi utilizzando un argomento di Sport
- Publisher 2 sta pubblicando informazioni sui prezzi delle azioni utilizzando un argomento di Stock
- L'editore 3 sta pubblicando informazioni sulle recensioni dei film utilizzando un argomento di Films e sugli elenchi televisivi utilizzando un argomento di TV

Tre sottoscrittori hanno registrato un interesse per argomenti differenti, quindi il gestore code invia loro le informazioni a cui sono interessati:

- L'abbonato 1 riceve i risultati sportivi e le quotazioni azionarie
- L'abbonato 2 riceve le recensioni del film
- L'abbonato 3 riceve i risultati sportivi

Nessuno degli abbonati ha registrato un interesse per gli elenchi televisivi, quindi questi non sono distribuiti.

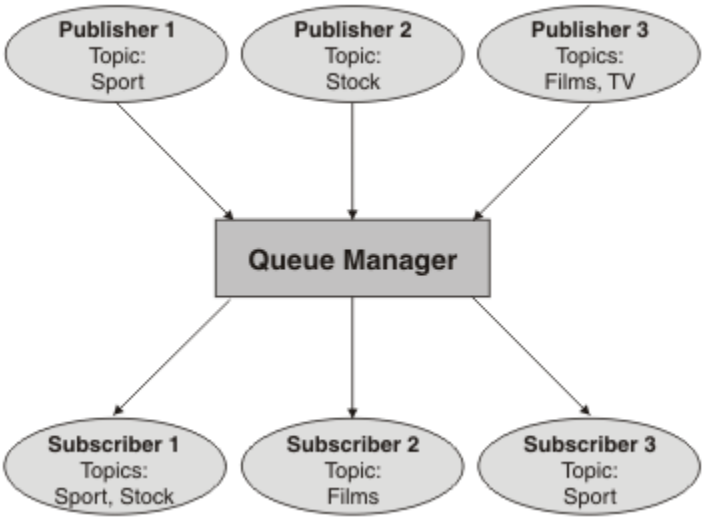


Figura 31. Esempio di pubblicazione / sottoscrizione di un singolo gestore code

Reti di pubblicazione / sottoscrizione distribuite

Ogni gestore code corrisponde ai messaggi pubblicati in un argomento con le sottoscrizioni create localmente che hanno sottoscritto tale argomento. È possibile configurare una rete di gestori code in modo che i messaggi pubblicati da un'applicazione connessa ad un gestore code vengano consegnati alle sottoscrizioni corrispondenti create su altri gestori code nella rete. Ciò richiede una configurazione aggiuntiva su canali semplici tra gestori code.

Una configurazione di pubblicazione / sottoscrizione distribuita è una serie di gestori code connessi tra loro. I gestori code possono trovarsi tutti sullo stesso sistema fisico oppure possono essere distribuiti su diversi sistemi fisici. Quando si collegano i gestori code, i sottoscrittori possono sottoscrivere un gestore code e ricevere i messaggi inizialmente pubblicati su un altro gestore code. Per illustrare questo aspetto, la seguente figura aggiunge un secondo gestore code alla configurazione descritta in “Esempio di configurazione di pubblicazione / sottoscrizione di un singolo gestore code” a pagina 90.

- Il gestore code 2 viene utilizzato dall'editore 4 per pubblicare le informazioni sulle previsioni meteo, utilizzando un argomento di Meteo, e le informazioni sulle condizioni del traffico sulle strade principali, utilizzando un argomento di Traffico.
- Anche il sottoscrittore (subscriber) 4 utilizza questo gestore code e sottoscrive le informazioni sulle condizioni del traffico utilizzando l'argomento Traffico.
- Il sottoscrittore 3 sottoscrive anche le informazioni sulle condizioni meteo, anche se utilizza un gestore code diverso dal publisher. Ciò è possibile perché i gestori code sono collegati tra loro.

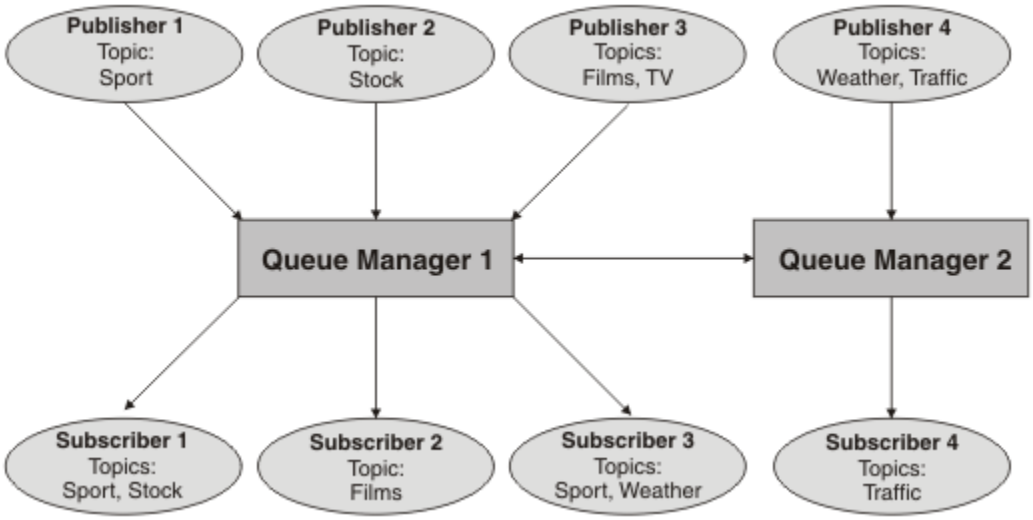


Figura 32. Esempio di pubblicazione / sottoscrizione con due gestori code

È possibile connettere manualmente i gestori code in una gerarchia parent e child oppure è possibile creare un cluster di pubblicazione / sottoscrizione e lasciare che IBM MQ definisca la maggior parte dei dettagli di connessione. È inoltre possibile utilizzare entrambe le topologie in combinazione, ad esempio unendo più cluster in una gerarchia.

Panoramica dei cluster di pubblicazione / sottoscrizione

Un cluster di pubblicazione / sottoscrizione è un cluster standard con uno o più oggetti argomento aggiunti al cluster. Quando si definisce un oggetto argomento di gestione su qualsiasi gestore code in un cluster e si crea un cluster di tale oggetto argomento specificando un nome cluster, i publisher e i sottoscrittori dell'argomento possono connettersi a qualsiasi gestore code nel cluster e i messaggi pubblicati vengono instradati ai sottoscrittori sui canali cluster tra i gestori code.

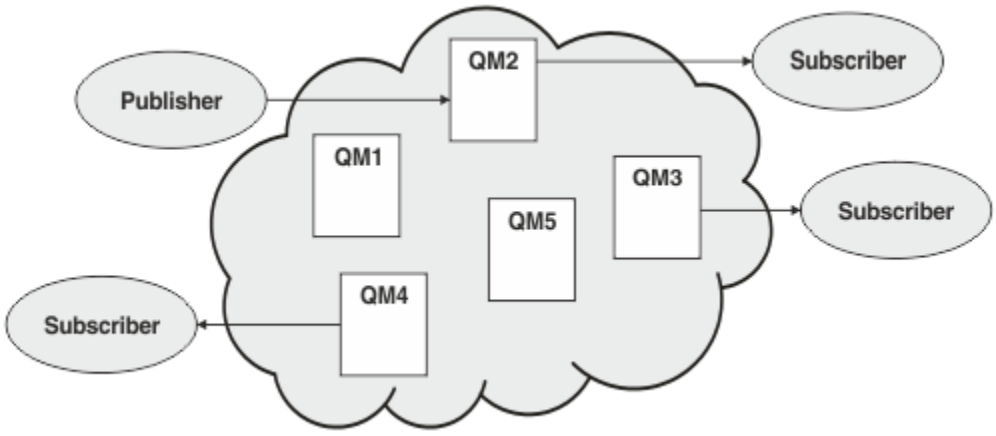


Figura 33. cluster di pubblicazione/sottoscrizione

Esistono due modi per configurare il modo in cui i messaggi di pubblicazione / sottoscrizione vengono instradati in un cluster:

- instradamento diretto
- instradamento host argomento

Quando si configura un argomento con cluster instradato direttamente, i messaggi pubblicati su un gestore code vengono inviati direttamente da tale gestore code a ogni sottoscrizione su qualsiasi altro gestore code nel cluster. Ciò può fornire il percorso più diretto per le pubblicazioni, ma fa sì che

tutti i gestori code in un cluster siano a conoscenza di tutti gli altri gestori code, ognuno dei quali potenzialmente dispone di canali cluster stabiliti tra loro.

Quando si utilizza l'instradamento host argomento, i messaggi pubblicati su un gestore code vengono inviati da lì a un gestore code su cui è presente una definizione dell'oggetto argomento gestito. Tale *gestore code dell'host argomento* instrada il messaggio a ogni sottoscrizione presente su qualsiasi altro gestore code nel cluster. Se i publisher o i sottoscrittori non si trovano sui gestori code dell'host argomento, ciò comporta un instradamento più lungo per le pubblicazioni. Tuttavia, il vantaggio consiste nel fatto che solo i gestori code dell'host argomento vengono a conoscenza di tutti gli altri gestori code nel cluster e potenzialmente dispongono di canali cluster stabiliti con essi.

Per ulteriori informazioni, consultare “Cluster di pubblicazione / sottoscrizione” a pagina 94.

Panoramica delle gerarchie di pubblicazione / sottoscrizione

Una gerarchia di pubblicazione / sottoscrizione è una serie di gestori code connessi da canali in una struttura gerarchica. Ogni gestore code identifica il proprio gestore code *principale* , come descritto in Connessione di un gestore code a una gerarchia di pubblicazione / sottoscrizione.

I publisher e i sottoscrittori di un argomento possono connettersi a qualsiasi gestore code nella gerarchia e il flusso di messaggi tra di essi utilizzando la connettività gerarchica del gestore code.

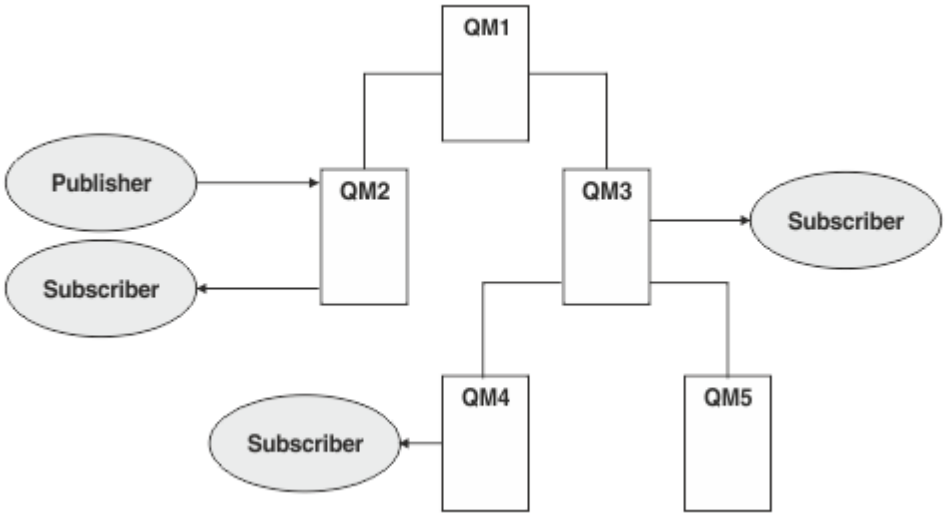


Figura 34. Gerarchia di pubblicazione / sottoscrizione

Nella figura precedente, le pubblicazioni fornite agli utenti su QM3 e QM4 sono state instradate da QM2 a QM1 e quindi su QM3 e infine su QM4.

Le gerarchie forniscono un controllo diretto sulle relazioni tra ogni gestore code nella gerarchia. Ciò consente un controllo dettagliato sull'instradamento dei messaggi dai publisher ai sottoscrittori ed è particolarmente utile quando si esegue l'instradamento tra reti di gestori code con connettività limitata. È necessario considerare attentamente la disponibilità e la funzionalità di ogni gestore code attraverso il quale un messaggio viene instradato dal publisher ai sottoscrittori.

Per ulteriori informazioni, consultare “Gerarchie di pubblicazione / sottoscrizione” a pagina 97.

Distribuzione della pubblicazione tra gestori code

Oltre alle scelte di instradamento, esistono due approcci per distribuire le pubblicazioni su una rete di gestori code:

- Inviare solo le pubblicazioni da un gestore code ai gestori code che attualmente ospitano una sottoscrizione per tale pubblicazione.
- Inviare ogni pubblicazione a tutti i gestori code e lasciare che corrispondano alle relative sottoscrizioni.

Il primo risultato è che i messaggi di pubblicazione vengono inviati solo dove necessario, ma richiede un livello di conoscenza della sottoscrizione da condividere tra i gestori code. Quest' ultima non richiede la condivisione della conoscenza della sottoscrizione, ma può comportare l'invio di messaggi di pubblicazione non necessari tra i gestori code.

Per impostazione predefinita, IBM MQ utilizza il metodo precedente, in cui le pubblicazioni sono inviate solo ai gestori code che dispongono di sottoscrizioni. La conoscenza della sottoscrizione viene propagata tra i gestori code sotto forma di *sottoscrizioni proxy*. Esso dipende dalla distribuzione e dalla durata delle sottoscrizioni e dalla frequenza delle pubblicazioni, per cui è il più efficiente da utilizzare in una topologia di pubblicazione / sottoscrizione distribuita. Vedere Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione.

Concetti correlati

“Strutture ad albero degli argomenti” a pagina 77
Ciascun argomento che viene definito è un elemento, o nodo, della struttura ad albero degli argomenti. La struttura ad albero degli argomenti può essere vuota per iniziare o contenere argomenti che sono stati precedentemente definiti utilizzando i comandi MQSC o PCF. È possibile definire un nuovo argomento utilizzando i comandi di creazione degli argomenti o specificando l'argomento per la prima volta in una pubblicazione o in una sottoscrizione.

Scenari di gerarchia di pubblicazione / sottoscrizione

Attività correlate

Progettazione di cluster di pubblicazione / sottoscrizione

Cluster di pubblicazione / sottoscrizione

Un cluster di pubblicazione / sottoscrizione è un cluster standard di gestori code interconnessi, su cui le pubblicazioni vengono spostate automaticamente dalle applicazioni di pubblicazione alle sottoscrizioni che esistono su uno qualsiasi dei gestori code nel cluster. Esistono due opzioni per l'instradamento delle pubblicazioni attraverso un cluster di pubblicazione/sottoscrizione: *instradamento diretto* e *instradamento all'host argomento*. L'instradamento scelto dipende dalla dimensione e dai modelli di attività previsti per il cluster.

Un cluster utilizzato per la messaggistica di pubblicazione / sottoscrizione non è diverso da un cluster IBM MQ standard. Pertanto, i gestori code all'interno del cluster di pubblicazione / sottoscrizione possono esistere su computer fisicamente separati e ogni coppia di gestori code viene connessa automaticamente dai canali cluster quando necessario. Per ulteriori informazioni, consultare Cluster.

Per configurare un cluster standard di gestori code per la messaggistica di pubblicazione/sottoscrizione, definire uno o più oggetti argomento amministrati su un gestore code nel cluster. Per rendere l'argomento un argomento cluster, configurare la proprietà **CLUSTER** con il nome del cluster. Quando si esegue questa operazione, qualsiasi argomento utilizzato da un publisher o sottoscrittore in quel punto o al di sotto della struttura ad albero degli argomenti viene condiviso tra tutti i gestori code nel cluster e i messaggi pubblicati in un ramo cluster della struttura ad albero degli argomenti vengono instradati automaticamente alle sottoscrizioni su altri gestori code nel cluster.

Viene inviata solo una copia di ciascun messaggio tra il gestore code del publisher e ognuno degli altri gestori code, indipendentemente dal numero di sottoscrittori per il messaggio sul gestore code di destinazione. All'arrivo in un gestore code con una o più sottoscrizioni, il messaggio viene duplicato in tutte le sottoscrizioni.

Qualsiasi gestore code che si unisce al cluster viene automaticamente a conoscenza degli argomenti del cluster e i publisher e i sottoscrittori su tale gestore code partecipano automaticamente al cluster.

L'attività di pubblicazione / sottoscrizione non in cluster può essere eseguita anche in un cluster di pubblicazione / sottoscrizione, utilizzando le stringhe di argomenti che non rientrano in un oggetto argomento in cluster.

Esistono due opzioni per l'instradamento delle pubblicazioni attraverso un cluster di pubblicazione/sottoscrizione: *instradamento diretto* e *instradamento all'host argomento*. Per scegliere il routing dei messaggi da utilizzare all'interno del cluster, impostare la proprietà **CLROUTE** sull'oggetto argomento amministrato su uno dei seguenti valori:

- **DIRECT**
- **TOPICHOST**

Per impostazione predefinita, l'instradamento argomento è **DIRECT**. Questa era l'unica opzione prima di IBM MQ 8.0. Quando si configura un argomento di cluster con instradamento diretto su un gestore code, tutti i gestori code presenti nel cluster sono a conoscenza di tutti gli altri gestori code del cluster. Quando si effettuano operazioni di pubblicazione e sottoscrizione, ogni gestore code può collegarsi direttamente ad ogni altro gestore code nel cluster.

Da IBM MQ 8.0, è invece possibile configurare l'instradamento degli argomenti come **TOPICHOST**. Quando si utilizza l'instradamento all'host argomento, tutti i gestori code presenti nel cluster sono a conoscenza dei gestori code del cluster che ospitano le definizioni dell'argomento instradato (ossia, i gestori code in cui è stato definito l'oggetto dell'argomento). Quando si effettuano operazioni di pubblicazione e sottoscrizione, i gestori code del cluster si connettono soltanto a questi gestori code dell'host argomento e non direttamente l'uno all'altro. I gestori code dell'host argomento sono responsabili dell'instradamento delle pubblicazioni dai gestori code su cui vengono pubblicate le pubblicazioni ai gestori code con le sottoscrizioni corrispondenti.

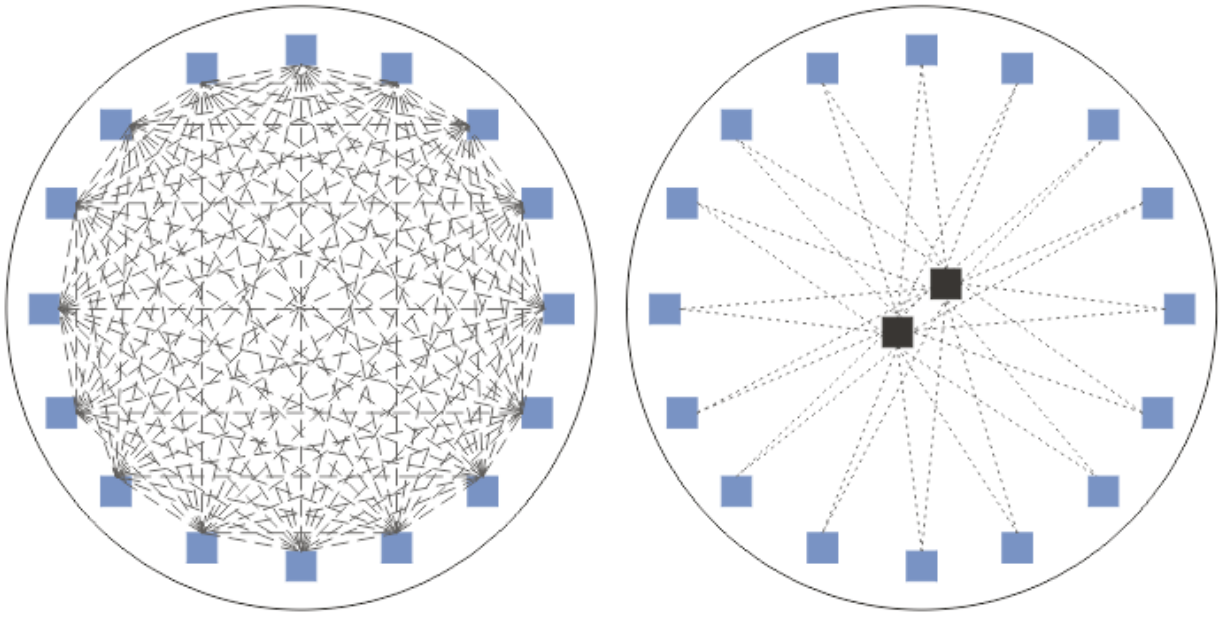


Figura 35. Instradamento diretto e instradamento host argomento

Una panoramica dell'instradamento diretto

Quando un oggetto argomento gestito è configurato per l'instradamento diretto, l'oggetto argomento deve essere definito solo su uno dei gestori code nel cluster per consentire a tutti i gestori code di conoscerlo. La scelta del gestore code su cui è definito il topic non influisce sul funzionamento della messaggistica di pubblicazione / sottoscrizione per il topic.

Ogni messaggio fluisce direttamente dal gestore code del publisher a ciascuna sottoscrizione sugli altri gestori code nel cluster, non passando attraverso alcun gestore code intermedio.

Per impostazione predefinita, i messaggi vengono inviati solo ad altri gestori code nel cluster che ospitano una o più sottoscrizioni.

- Ciò si basa su ciascun gestore code che informa direttamente tutti gli altri gestori code nel cluster di tutti gli argomenti che attualmente hanno una o più sottoscrizioni. Ciò comporta che tutti i gestori code nel cluster siano a conoscenza di tutti gli argomenti sottoscritti e di tutti i gestori code che ospitano una sottoscrizione che stabilisce un canale per ogni altro gestore code. Ciò è indipendente dal fatto che ciascun gestore code abbia o meno un publisher.

- La conoscenza di ogni singolo argomento sottoscritto su tutti i gestori code può essere rimossa passando a un modello di invio di tutte le pubblicazioni a tutti i gestori code nel cluster, indipendentemente dal fatto che dispongano o meno di sottoscrizioni. Ciò riduce il traffico di conoscenza delle sottoscrizioni, ma è probabile che aumenti il traffico di pubblicazione e il numero di canali stabilito da ogni gestore code. Vedere Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione.

I flussi di messaggi di pubblicazione / sottoscrizione che utilizzano argomenti con cluster instradati direttamente possono estendersi a più cluster di pubblicazione / sottoscrizione aggiungendo un gestore code da ciascun cluster in una gerarchia di pubblicazione / sottoscrizione. Consultare Combinazione di spazi argomento di più cluster.

Per un'esplorazione più dettagliata dell'instradamento diretto, consultare Instradamento diretto nei cluster di pubblicazione / sottoscrizione.

Una panoramica dell'instradamento dell'host argomento

Quando un oggetto argomento amministrato è configurato per l'instradamento dell'host argomento, le pubblicazioni da un gestore code nel cluster vengono instradate attraverso un gestore code in cui è configurato l'oggetto argomento (un "host argomento") e da lì in poi verso i gestori code in cui esistono le sottoscrizioni.

- Ciò si basa su ciascun gestore code che informa tutti gli host argomento di ogni argomento che attualmente dispone di una o più sottoscrizioni. Qualsiasi gestore code che ospita una sottoscrizione stabilisce un canale per ogni host argomento per l'argomento a cui è correlata la sottoscrizione.
- I gestori code che ospitano non argomenti non sono a conoscenza di altri gestori code che ospitano non argomenti nel cluster per scopi di pubblicazione / sottoscrizione e i canali non sono stabiliti tra loro per tale scopo.
- Se l'applicazione di pubblicazione è connessa a un gestore code che ospita l'argomento, i messaggi pubblicati vengono instradati direttamente ai gestori code in cui sono state create le sottoscrizioni corrispondenti, senza richiedere un ulteriore 'hop'. Allo stesso modo, se le sottoscrizioni corrispondenti vengono create sull'unico gestore code che ospita l'argomento, i messaggi pubblicati in tale argomento vengono instradati direttamente a tale gestore code, senza richiedere un ulteriore hop.
- Le sottoscrizioni sullo stesso gestore code del publisher vengono soddisfatte senza prima instradare le pubblicazioni agli host dell'oggetto argomento.

Come per le code cluster, più gestori code possono configurare lo stesso oggetto argomento di gestione. Ciò fornisce una maggiore disponibilità di instradamento dei messaggi e scalabilità orizzontale attraverso il bilanciamento del carico di lavoro. Per gli oggetti argomento instradati dell'host argomento, quando più gestori code configurano lo stesso argomento denominato per lo stesso ramo della struttura ad albero degli argomenti, ogni host argomento viene reso consapevole degli argomenti sottoscritti da ogni gestore code che ospita una sottoscrizione.

- Quando un messaggio viene pubblicato, viene inviato a un gestore code dell'host argomento per essere inoltrato alla sottoscrizione che ospita i gestori code. La scelta del gestore code dell'host argomento segue le stesse regole di bilanciamento del carico di lavoro predefinite delle code point-to-point con cluster.
- Se uno o più gestori code dell'host argomento non possono essere contattati da un gestore code di pubblicazione, i messaggi vengono instradati ai restanti gestori code dell'host argomento disponibili.

Ogni pubblicazione a un argomento in un ramo instradato della struttura ad albero degli argomenti viene inoltrata a uno degli host degli argomenti, anche se non vi sono sottoscrizioni a tale argomento in alcun punto del cluster. Per impostazione predefinita, i messaggi vengono inviati da qui solo ad altri gestori code nel cluster che ospitano una o più sottoscrizioni.

- Ciò si basa sul fatto che ciascun gestore code dell'host argomento venga informato di tutte le stringhe argomento sottoscritte su ciascun gestore code nel cluster.
- La conoscenza di ogni singolo argomento sottoscritto può essere rimossa passando a un modello di invio di tutte le pubblicazioni instradate a un host argomento su tutti i gestori code nel

cluster, indipendentemente dal fatto che dispongano o meno di sottoscrizioni. Ciò riduce il traffico di conoscenza della sottoscrizione, ma è probabile che aumenti il traffico di pubblicazione e potenzialmente il numero di canali stabiliti con ciascun gestore code che ospita l'argomento. Vedere Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione.

I flussi di messaggi di pubblicazione / sottoscrizione utilizzando gli argomenti del cluster instradati dell'host argomento **non possono** estendersi a più cluster di pubblicazione / sottoscrizione tramite l'uso di una gerarchia di pubblicazione / sottoscrizione.

Per un'esplorazione più dettagliata dell'instradamento dell'host argomento, consultare Instradamento dell'host argomento nei cluster di pubblicazione / sottoscrizione.

Gerarchie di pubblicazione / sottoscrizione

Si crea una gerarchia di pubblicazione / sottoscrizione collegando i gestori code utilizzando i canali, quindi definendo una relazione child - parent tra coppie di gestori code. Un messaggio fluisce da un publisher alle sottoscrizioni attraverso le relazioni dirette in una gerarchia. Nota che questo potrebbe significare più "hop" per arrivarci.

Solo una copia del messaggio viene inviata tra una coppia di gestori code, indipendentemente dal numero di sottoscrittori del messaggio sul gestore code di destinazione. All'arrivo in un gestore code con una o più sottoscrizioni, il messaggio viene duplicato in tutte le sottoscrizioni.

Per impostazione predefinita, i messaggi vengono inviati solo ad altri gestori code nella gerarchia che si trovano sull'instradamento a una sottoscrizione su un altro gestore code:

- Ciò si basa sul fatto che ogni gestore code informi ogni relazione diretta di tutti gli argomenti che attualmente hanno una o più sottoscrizioni, su questo gestore code o su una delle sue altre relazioni. Ciò fa sì che tutti i gestori code nella gerarchia siano a conoscenza di tutti gli argomenti sottoscritti.
- Questo comportamento può essere modificato per inviare sempre le pubblicazioni a tutti i gestori code della gerarchia, indipendentemente dalle sottoscrizioni esistenti. Ciò elimina la necessità di propagare le informazioni di sottoscrizione nella gerarchia, ma può aumentare il traffico di pubblicazione.

Quando si crea un cluster, è necessario prestare attenzione a non creare un loop che causi il ciclo dei messaggi all'interno della rete. Non è possibile creare tali loop in una gerarchia.

Ogni gestore code deve avere un nome gestore code univoco.

I flussi di messaggi di pubblicazione / sottoscrizione possono estendersi a più cluster di pubblicazione / sottoscrizione. A tale scopo, aggiungere un gestore code da ciascun cluster in una gerarchia di pubblicazione / sottoscrizione.

Per un'esplorazione più dettagliata, consultare Instradamento nelle gerarchie di pubblicazione / sottoscrizione.

Sottoscrizioni proxy in una rete di pubblicazione / sottoscrizione

Una sottoscrizione proxy è una sottoscrizione effettuata da un gestore code per gli argomenti pubblicati su un altro gestore code. Una sottoscrizione proxy transita tra i gestori code per ogni singola stringa argomento sottoscritta da una sottoscrizione. Non si creano le sottoscrizioni proxy esplicitamente; il gestore code lo fa per conto dell'utente.

È possibile connettere i gestori code in un cluster di pubblicazione / sottoscrizione o in una gerarchia di pubblicazione / sottoscrizione. Le sottoscrizioni proxy passano tra i gestori code connessi. Le sottoscrizioni proxy fanno in modo che le pubblicazioni di un argomento creato da un publisher connesso a un gestore code vengano ricevute dai sottoscrittori di tale argomento connessi ad altri gestori code. Consultare “Reti di pubblicazione / sottoscrizione distribuite” a pagina 91.

Nelle topologie di pubblicazione / sottoscrizione con molte migliaia di sottoscrizioni a singole stringhe di argomenti o dove l'esistenza di tali sottoscrizioni potrebbe essere in rapida modifica, è necessario considerare il sovraccarico della propagazione della sottoscrizione proxy. Oltre all'aggregazione automatica descritta nel resto di questo argomento, è possibile apportare modifiche di configurazione manuali che limitano ulteriormente il flusso delle sottoscrizioni proxy e delle pubblicazioni tra i gestori

code connessi e che riducono la latenza dell'attesa della propagazione di una sottoscrizione proxy a tutti i gestori code connessi. Vedere Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione.

Le sottoscrizioni proxy non contengono selettori utilizzati dalle sottoscrizioni locali e le stringhe di argomenti di sottoscrizione che contengono caratteri jolly potrebbero essere semplificate. Ciò può risultare in pubblicazioni che corrispondono alle sottoscrizioni proxy laddove non lo fanno le sottoscrizioni effettive, con conseguente ulteriore flusso di pubblicazione tra i gestori code. Il gestore code che ospita le sottoscrizioni filtra tali discrepanze in modo che le pubblicazioni aggiuntive non vengano restituite alle sottoscrizioni.

Aggregazione sottoscrizione proxy

Le sottoscrizioni proxy vengono aggregate utilizzando un sistema di eliminazione duplicato. Per una particolare stringa di argomenti risolta, una sottoscrizione proxy viene inviata alla prima sottoscrizione locale o alla sottoscrizione proxy ricevuta. Le sottoscrizioni successive alla stessa stringa di argomenti utilizzano questa sottoscrizione proxy esistente.

La sottoscrizione proxy viene annullata dopo l'annullamento dell'ultima sottoscrizione locale o della sottoscrizione proxy ricevuta.

Aggregazione pubblicazione

Quando è presente più di una sottoscrizione alla stessa stringa di argomenti su un gestore code, solo una singola copia di ogni pubblicazione corrispondente a tale stringa di argomenti viene inviata da altri gestori code nella topologia di pubblicazione / sottoscrizione. All'arrivo del messaggio, il gestore code locale consegna una copia del messaggio a ciascuna sottoscrizione corrispondente.

È possibile che più di una sottoscrizione proxy corrisponda alla stringa argomento di una singola pubblicazione quando le sottoscrizioni proxy contengono caratteri jolly. Se un messaggio viene pubblicato su un gestore code che corrisponde a due o più sottoscrizioni proxy create da un singolo gestore code connesso, solo una copia della pubblicazione viene inoltrata al gestore code remoto per soddisfare le sottoscrizioni proxy multiple.

Concetti correlati

Rilevamento loop in una rete di pubblicazione / sottoscrizione distribuita

Caratteri jolly nelle sottoscrizioni proxy

Le sottoscrizioni possono utilizzare caratteri jolly nelle stringhe di argomenti per la corrispondenza con più stringhe di argomenti nelle pubblicazioni.

Esistono due schemi di caratteri jolly che una sottoscrizione può utilizzare: *basato su argomenti* e *basato su caratteri*. Consultare“Schemi dei caratteri jolly” a pagina 71.

In IBM MQ tutte le sottoscrizioni proxy per le sottoscrizioni jolly vengono convertite per utilizzare i caratteri jolly basati sull'argomento. Se viene trovato un carattere jolly basato sui caratteri, viene sostituito con un carattere `#` , di nuovo al carattere `/` più vicino. Ad esempio, `/aaa/bbb/c*d` viene convertito in `/aaa/bbb/#`. La conversione risulta in gestori code remoti che inviano un numero di pubblicazioni leggermente superiore a quello esplicitamente sottoscritto. Le pubblicazioni aggiuntive vengono filtrate dal gestore code locale, quando consegna le pubblicazioni ai relativi sottoscrittori locali.

Controllo dell'utilizzo dei caratteri jolly con la proprietà WILDCARD

Utilizzare la proprietà MQSC **`Topic`** WILDCARD o la proprietà Argomento `WildcardOperation` PCF equivalente per controllare la consegna di pubblicazioni alle applicazioni del sottoscrittore che utilizzano nomi stringa di argomenti con caratteri jolly. La proprietà WILDCARD può avere uno dei due valori possibili:

WILDCARD
Il funzionamento delle sottoscrizioni con caratteri jolly rispetto a questo argomento.

PASSTHRU
Le sottoscrizioni effettuate a un argomento con carattere jolly meno specifico della stringa argomento in questo oggetto argomento riceveranno le pubblicazioni relative a questo argomento e a stringhe argomento più specifiche di tale argomento.

BLOCK
Le sottoscrizioni effettuate a un argomento con carattere jolly meno specifico della stringa argomento in questo oggetto argomento non riceveranno le pubblicazioni relative a questo argomento o a stringhe argomento più specifiche di tale argomento.

Il valore di questo attributo è utilizzato quando vengono definite sottoscrizioni. Se si modifica questo attributo, la serie di argomenti trattati dalle sottoscrizioni esistenti non viene interessata dalla modifica. Questo scenario si applica anche se la topologia cambia quando si creano o eliminano oggetti argomento; la serie di argomenti che corrispondono alle sottoscrizioni create in seguito alla modifica dell'attributo WILDCARD viene creata utilizzando la topologia modificata. Se si desidera forzare una rivalutazione della serie corrispondente di argomenti per le sottoscrizioni esistenti, è necessario riavviare il gestore code.

Nell'esempio, “Esempio: creazione del cluster di pubblicazione / sottoscrizione Sport” a pagina 85, è possibile seguire la procedura per creare la struttura ad albero dell'argomento mostrata in Figura 23 a pagina 82.

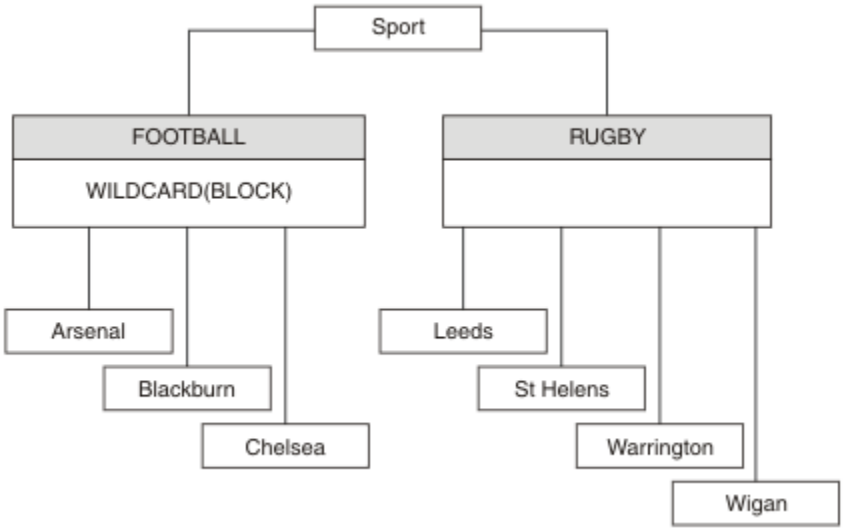


Figura 36. Una struttura di argomenti che utilizza la proprietà WILDCARD , BLOCK

Un sottoscrittore che utilizza la stringa di argomenti con caratteri jolly # riceve tutte le pubblicazioni nell'argomento `Sport` e nella struttura ad albero secondaria `Sport/Rugby` . Il sottoscrittore non riceve alcuna pubblicazione nella struttura secondaria `Sport/Football` , poiché il valore della proprietà `WILDCARD` dell'argomento `Sport/Football` è `BLOCK`.

`PASSTHRU` è l'impostazione predefinita. È possibile impostare il valore della proprietà `WILDCARD` `PASSTHRU` sui nodi nella struttura ad albero `Sport` . Se i nodi non hanno il valore della proprietà `WILDCARD` `BLOCK`, l'impostazione `PASSTHRU` non modifica il funzionamento osservato dai sottoscrittori ai nodi nella albero `Sports` .

Nell'esempio, creare sottoscrizioni per vedere in che modo l'impostazione del carattere jolly influisce sulle pubblicazioni fornite; consultare Figura 27 a pagina 87. Eseguire il comando publish in Figura 30 a pagina 88 per creare alcune pubblicazioni.

```
pub QMA
```

Figura 37. Pubblica in QMA

I risultati sono riportati in Tabella 3 a pagina 82. Si noti come l'impostazione del valore della proprietà WILDCARD BLOCKimpedisca alle sottoscrizioni con caratteri jolly di ricevere pubblicazioni per argomenti nell'ambito del carattere jolly.

Tabella 6. Pubblicazioni ricevute su QMA

Sottoscrizione	Stringa argomento	Pubblicazioni ricevute	Note
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Tutte le pubblicazioni della sottostruttura Football bloccate da WILDCARD(BLOCK) su Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD(BLOCK) on Sports/Football impedisce la sottoscrizione di caratteri jolly su Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Il valore predefinito WILDCARD su Sports / Rugby non impedisce la sottoscrizione di caratteri jolly su Leeds.

Nota:

Si supponga che una sottoscrizione abbia un carattere jolly che corrisponde a un oggetto argomento con il valore della proprietà WILDCARD BLOCK. Se la sottoscrizione ha anche una stringa di argomenti a destra del carattere jolly corrispondente, la sottoscrizione non riceverà mai una pubblicazione. La serie di pubblicazioni che non sono bloccate sono pubblicazioni per argomenti che sono parent del carattere jolly bloccato. Le pubblicazioni per gli argomenti child dell'argomento con il valore della proprietà BLOCK sono bloccate dal carattere jolly. Pertanto, le stringhe di argomenti di sottoscrizione che includono un argomento alla destra del carattere jolly non ricevono mai alcuna pubblicazione corrispondente.

L'impostazione del valore della proprietà WILDCARD su BLOCK non significa che non è possibile effettuare la sottoscrizione utilizzando una stringa di argomenti che include caratteri jolly. Tale sottoscrizione è normale. La sottoscrizione dispone di un argomento esplicito che corrisponde all'argomento con un oggetto argomento con un valore della proprietà WILDCARD BLOCK. Utilizza i caratteri jolly per gli argomenti che sono parent o child dell'argomento con il valore della proprietà WILDCARD BLOCK. Nell'esempio in Figura 23 a pagina 82, una sottoscrizione come Sports/Football/# può ricevere pubblicazioni.

Caratteri jolly e argomenti cluster

Le definizioni degli argomenti del cluster vengono propagati a ogni gestore code in un cluster. Una sottoscrizione a un argomento cluster in un gestore code in un cluster determina la creazione di sottoscrizioni proxy da parte del gestore code. Una sottoscrizione proxy viene creata su ogni altro gestore code nel cluster. Le sottoscrizioni che utilizzano stringhe di argomenti contenenti caratteri jolly, combinate con argomenti cluster, possono fornire un comportamento difficile da prevedere. Il comportamento viene spiegato nel seguente esempio.

Nel cluster configurato per l'esempio, “Esempio: creazione del cluster di pubblicazione / sottoscrizione Sport” a pagina 85, QMB ha la stessa serie di sottoscrizioni di QMA, ma QMB non ha ricevuto alcuna pubblicazione dopo che il publisher è stato pubblicato in QMA, consultare Figura 24 a pagina 82. Anche se gli argomenti Sports/Football e Sports/Rugby sono argomenti del cluster, le sottoscrizioni definite in fullsubs.tst non fanno riferimento a un argomento del cluster. Nessuna sottoscrizione proxy viene propagata da QMB a QMA. Senza sottoscrizioni proxy, nessuna pubblicazione per QMA viene inoltrata a QMB.

Alcune sottoscrizioni, come Sports/#/Leeds, potrebbero far riferimento a un argomento cluster, in questo caso Sports/Rugby. La sottoscrizione Sports/#/Leeds viene effettivamente risolta nell'oggetto argomento SYSTEM.BASE.TOPIC.

La regola per risolvere l'oggetto argomento a cui fa riferimento una sottoscrizione come, Sports/#/Leeds è la seguente. Troncare la stringa argomento al primo carattere jolly. Eseguire la scansione a sinistra nella stringa dell'argomento cercando il primo argomento a cui è associato un oggetto argomento di gestione. L'oggetto argomento potrebbe specificare un nome cluster o definire un oggetto argomento locale. Nell'esempio, Sports/#/Leeds, la stringa di argomenti dopo il troncamento è Sports, che non ha alcun oggetto argomento e quindi Sports/#/Leeds eredita da SYSTEM.BASE.TOPIC, che è un oggetto argomento locale.

Per vedere come la sottoscrizione agli argomenti del cluster può modificare il funzionamento della propagazione dei caratteri jolly, eseguire lo script batch upsubs.bat. Lo script elimina le code di sottoscrizione e aggiunge le sottoscrizioni dell'argomento del cluster in fullsubs.tst. Eseguire nuovamente puba.bat per creare un batch di pubblicazioni; consultare Figura 24 a pagina 82.

Tabella 4 a pagina 84 mostra il risultato dell'aggiunta di due nuove sottoscrizioni allo stesso gestore code su cui sono state pubblicate le pubblicazioni. Il risultato è quello previsto, le nuove sottoscrizioni ricevono una pubblicazione ciascuna e il numero di pubblicazioni ricevute dalle altre sottoscrizioni è invariato. I risultati imprevisti si verificano sull'altro gestore code cluster; consultare Tabella 5 a pagina 84.

Tabella 7. Pubblicazioni ricevute su QMA

Sottoscrizione	**Stringa argomento**	**Pubblicazioni ricevute**	**Note**
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Tutte le pubblicazioni della sottostruttura Football bloccate da WILDCARD(BLOCK) su Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD(BLOCK) on Sports/Football impedisce la sottoscrizione di caratteri jolly su Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Il valore predefinito WILDCARD su Sports / Rugby non impedisce la sottoscrizione di caratteri jolly su Leeds.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal riceve una pubblicazione perché la sottoscrizione non ha un carattere jolly.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds riceverà una pubblicazione in ogni caso.

Tabella 5 a pagina 84 mostra i risultati dell'aggiunta delle due nuove sottoscrizioni su QMB e della pubblicazione su QMA. Si ricordi che QMB non ha ricevuto alcuna pubblicazione senza queste due nuove sottoscrizioni. Come previsto, le due nuove sottoscrizioni ricevono le pubblicazioni, poiché Sports/FootBall e Sports/Rugby sono entrambi argomenti cluster. QMB ha inoltrato le sottoscrizioni proxy per Sports/Football/Arsenal e Sports/Rugby/Leeds a QMA, che ha quindi inviato le pubblicazioni a QMB.

Il risultato imprevisto è che le due sottoscrizioni Sports/# e Sports/#/Leeds che in precedenza non avevano ricevuto alcuna pubblicazione, ora ricevono le pubblicazioni. Il motivo è che le pubblicazioni Sports/Football/Arsenal e Sports/Rugby/Leeds inoltrate a QMB per le altre sottoscrizioni sono ora disponibili per tutti i sottoscrittori collegati a QMB. Di conseguenza, le sottoscrizioni agli argomenti locali Sports/# e Sports/#/Leeds ricevono la pubblicazione Sports/Rugby/Leeds . Sports/#/

Arsenal continua a non ricevere una pubblicazione, poiché Sport / Calcio ha il valore della proprietà WILDCARD impostato su BLOCK.

Tabella 8. Pubblicazioni ricevute su QMB

Sottoscrizione	Stringa argomento	Pubblicazioni ricevute	Note
SPORTS	*Sports/#*	*Sports/Rugby/Leeds*	*Tutte le pubblicazioni nell'albero secondario Calcio bloccate da WILDCARD(BLOCK) su Sports/Football*
SARSENAL	Sports/#/Arsenal	-	WILDCARD(BLOCK) on Sports/Football impedisce la sottoscrizione di caratteri jolly su Arsenal
SLEEDS	*Sports/#/Leeds*	*Sports/Rugby/Leeds*	*Predefinito WILDCARD su Sports / Rugby non impedisce la sottoscrizione di caratteri jolly su Leeds.*
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal riceve una pubblicazione perché la sottoscrizione non ha un carattere jolly.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds riceverà una pubblicazione in ogni caso.

Nella maggior parte delle applicazioni, non è desiderabile che una sottoscrizione influenzi il comportamento di un'altra sottoscrizione. Un utilizzo importante della proprietà WILDCARD con valore BLOCK consiste nel rendere le sottoscrizioni alla stessa stringa di argomenti contenenti caratteri jolly uniformi. Se la sottoscrizione si trova sullo stesso gestore code del publisher o su un gestore code diverso, i risultati della sottoscrizione sono gli stessi.

Caratteri jolly e stream

Per una nuova applicazione scritta nell'API di pubblicazione / sottoscrizione, l'effetto è che una sottoscrizione a * non riceve alcuna pubblicazione. Per ricevere tutte le pubblicazioni Sport, è necessario sottoscrivere Sports/*o Sports/#e in modo simile per le pubblicazioni Business .

Il comportamento di un'applicazione di pubblicazione / sottoscrizione accodata esistente non cambia quando il Broker di pubblicazione / sottoscrizione viene migrato a una versione successiva di IBM MQ. La proprietà **StreamName** nei comandi **Publish**, **Register Publisher**o **Subscriber** è associata al nome dell'argomento in cui è stato migrato il flusso.

Caratteri jolly e punti di sottoscrizione

Per una nuova applicazione scritta nell'API di pubblicazione / sottoscrizione, l'effetto della migrazione è che una sottoscrizione a * non riceve alcuna pubblicazione. Per ricevere tutte le pubblicazioni Sport, è necessario sottoscrivere Sports/*o Sports/#e in modo simile per le pubblicazioni Business .

Il comportamento di un'applicazione di pubblicazione / sottoscrizione accodata esistente non cambia quando il Broker di pubblicazione / sottoscrizione viene migrato a una versione successiva di IBM MQ. La proprietà **SubPoint** nei comandi **Publish**, **Register Publisher**o **Subscriber** è associata al nome dell'argomento a cui è stata migrata la sottoscrizione.

Esempio: creazione del cluster di pubblicazione / sottoscrizione Sport

I passi che seguono creano un cluster, CL1, con quattro gestori code: due repository completi, CL1A e CL1B, e due repository parziali, QMA e QMB. I repository completi vengono utilizzati per contenere solo

le definizioni cluster. QMA è designato come host argomento del cluster. Le sottoscrizioni durevoli sono definite sia su QMA che su QMB.

Nota: L'esempio è codificato per Windows. È necessario ricodificare Crea qmgrs.bat e creare pub.bat per configurare e verificare l'esempio su altre piattaforme.

1. Creare i file script.
   a. Crea topics.tst
   b. Crea wildsubs.tst
   c. Crea fullsubs.tst
   d. Crea qmgrs.bat
   e. create pub.bat
2. Eseguire Create qmgrs.bat per creare la configurazione.

```
qmgrs
```

Creare gli argomenti in Figura 23 a pagina 82. Lo script nella figura 5 crea gli argomenti del cluster Sports/Football e Sports/Rugby.

Nota: L'opzione REPLACE non sostituisce le proprietà TOPICSTR di un argomento. TOPICSTR è una proprietà che viene utilmente variata nell'esempio per testare diverse strutture ad albero degli argomenti. Per modificare gli argomenti, eliminare prima l'argomento.

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports')   TOPICSTR('Sports')
DEFINE TOPIC ('Football')  TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal')  TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea')  TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby')   TOPICSTR('Sports/Rugby')  CLUSTER(CL1)
DEFINE TOPIC ('Leeds')   TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan')   TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

Figura 38. Eliminare e creare argomenti: topics.tst

Nota: Eliminare gli argomenti, poiché REPLACE non sostituisce le stringhe di argomenti.

Creare sottoscrizioni con caratteri jolly. I caratteri jolly corrispondenti agli argomenti con gli oggetti argomento in Figura 23 a pagina 82. Creare una coda per ogni sottoscrizione. Le code vengono cancellate e le sottoscrizioni vengono eliminate quando lo script viene eseguito o rieseguito.

Nota: L'opzione REPLACE non sostituisce le proprietà TOPICOBJ o TOPICSTR di una sottoscrizione. TOPICOBJ o TOPICSTR sono le proprietà che sono utilmente variate nell'esempio per testare sottoscrizioni differenti. Per modificarli, eliminare prima la sottoscrizione.

```
DEFINE QLOCAL(QSPORTS)  REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS)  REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB  (SPORTS)
DELETE SUB  (SARSENAL)
DELETE SUB  (SLEEDS)
DEFINE SUB  (SPORTS)  TOPICSTR('Sports/#')     DEST(QSPORTS)
DEFINE SUB  (SARSENAL) TOPICSTR('Sports/+/Arsenal') DEST(QSARSENAL)
DEFINE SUB  (SLEEDS)  TOPICSTR('Sports/+/Leeds')  DEST(QSLEEDS)
```

Figura 39. Creare sottoscrizioni jolly: wildsubs.tst

Creare sottoscrizioni che facciano riferimento agli oggetti argomento del cluster.

Nota:

Il delimitatore, /, viene automaticamente inserito tra la stringa argomento a cui fa riferimento TOPICOBJe la stringa argomento definita da TOPICSTR.

La definizione DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) crea la stessa sottoscrizione. TOPICOBJ viene utilizzato come un modo rapido per fare riferimento alla stringa di argomenti già definita. La sottoscrizione, una volta creata, non fa più riferimento all'oggetto argomento.

```
DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS)  REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB  (FARSENAL)
DELETE SUB  (RLEEDS)
DEFINE SUB  (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB  (RLEEDS)  TOPICOBJ('Rugby')  TOPICSTR('Leeds')  DEST(QRLEEDS)
```

Figura 40. Eliminare e creare sottoscrizioni: fullsubs.tst

Creare un cluster con due repository. Creare due repository parziali per la pubblicazione e la sottoscrizione. Eseguire nuovamente lo script per eliminare tutto e ricominciare. Lo script crea anche la gerarchia degli argomenti e le sottoscrizioni dei caratteri jolly iniziali.

Nota:

Su altre piattaforme, scrivere uno script simile oppure immettere tutti i comandi. L'uso di uno script rende veloce l'eliminazione di tutto e ricomincia con una configurazione identica.

```
@echo off
set port.CL1B=1421
set port.CLIA=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof
```

Figura 41. Creare gestori code: qmgrs.bat

Aggiorna la configurazione aggiungendo le sottoscrizioni agli argomenti del cluster.

```
@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst
```

Figura 42. Aggiorna sottoscrizioni: upsubs.bat

Eseguire pub.bat, con un gestore code come parametro, per pubblicare i messaggi contenenti la stringa di argomenti di pubblicazione. Pub.bat utilizza il programma di esempio **amqspub**.

```
@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1
```

Figura 43. Pubblica: pub.bat

Concetti correlati

Sottoscrizioni jolly e pubblicazioni conservate

Ambito della pubblicazione

Quando si configura un cluster di pubblicazione / sottoscrizione o una gerarchia, l'ambito di una pubblicazione controlla ulteriormente se i gestori code inoltrano una pubblicazione ai gestori code remoti. Utilizzare l'attributo dell'argomento **PUBSCOPE** per gestire l'ambito delle pubblicazioni.

Se una pubblicazione non viene inoltrata ai gestori code remoti, solo i sottoscrittori locali ricevono la pubblicazione.

Quando si utilizza un cluster di pubblicazione / sottoscrizione, l'ambito delle pubblicazioni è principalmente controllato dalla definizione di oggetti argomento in cluster in determinati punti della

struttura ad albero dell'argomento. L'ambito di pubblicazione deve essere impostato per consentire il flusso di pubblicazioni ad altri gestori code nel cluster. È necessario limitare l'ambito di pubblicazione per un argomento con cluster solo quando è necessario un controllo più preciso di argomenti specifici su determinati gestori code.

Quando si utilizza una gerarchia di pubblicazione / sottoscrizione, l'ambito delle pubblicazioni è controllato principalmente da questo attributo in combinazione con l'attributo ambito sottoscrizione .

L'attributo **PUBSCOPE** viene utilizzato per stabilire l'ambito delle pubblicazioni effettuate per un argomento specifico. È possibile impostare l'attributo su uno dei seguenti valori:

QMGR
: La pubblicazione viene consegnata solo ai sottoscrittori locali. Queste pubblicazioni sono denominate *pubblicazioni locali*. Le pubblicazioni locali non vengono inoltrate ai gestori code remoti e quindi non vengono ricevute dai sottoscrittori connessi ai gestori code remoti.

TUTTO
: La pubblicazione viene consegnata ai sottoscrittori locali e ai sottoscrittori connessi ai gestori code remoti in una gerarchia o in un cluster di pubblicazione / sottoscrizione. Queste pubblicazioni sono denominate *pubblicazioni globali*.

Come parent
: Utilizzare l'impostazione **PUBSCOPE** dell'argomento principale nella struttura ad albero degli argomenti.

I publisher possono anche specificare se una pubblicazione è locale o globale utilizzando l'opzione di inserimento del messaggio MQPMO_SCOPE_QMGR . Se viene utilizzata questa opzione, sovrascrive qualsiasi comportamento impostato utilizzando l'attributo argomento **PUBSCOPE** .

Concetti correlati

“Oggetti argomento di gestione” a pagina 78
Utilizzando un oggetto argomento di amministrazione, è possibile assegnare specifici attributi non predefiniti agli argomenti.

Attività correlate

Configurazione delle reti di pubblicazione / sottoscrizione distribuite

Ambito della sottoscrizione

L'ambito di una sottoscrizione controlla se una sottoscrizione su un gestore code riceve le pubblicazioni pubblicate su un altro gestore code in un cluster o gerarchia di pubblicazione / sottoscrizione oppure solo le pubblicazioni dei publisher locali.

La limitazione dell'ambito di sottoscrizione a un gestore code impedisce l'inoltro delle sottoscrizioni proxy ad altri gestori code nella topologia di pubblicazione / sottoscrizione. Ciò riduce il traffico di messaggistica di pubblicazione / sottoscrizione tra gestori code.

Quando si utilizza un cluster di pubblicazione / sottoscrizione, l'ambito delle sottoscrizioni è controllato principalmente dalla definizione di oggetti argomento raggruppati in determinati punti della struttura ad albero degli argomenti. L'ambito della sottoscrizione deve essere impostato per consentire il flusso delle sottoscrizioni proxy ad altri gestori code nel cluster. Si consiglia di limitare l'ambito della sottoscrizione per un argomento cluster solo quando è necessario un controllo più preciso di argomenti specifici su determinati gestori code.

Quando si utilizza una gerarchia di pubblicazione / sottoscrizione, l'ambito delle sottoscrizioni è controllato principalmente da questo attributo in combinazione con l'attributo ambito di pubblicazione .

L'attributo dell'argomento **SUBSCOPE** viene utilizzato per determinare l'ambito delle sottoscrizioni effettuate a un argomento specifico. È possibile impostare l'attributo su uno dei seguenti valori:

QMGR
: Una sottoscrizione riceve solo pubblicazioni locali e le sottoscrizioni proxy non sono propagate ai gestori code remoti.

TUTTO
Una sottoscrizione proxy viene propagata ai gestori code remoti in un cluster o gerarchia di pubblicazione / sottoscrizione e il sottoscrittore riceve le pubblicazioni locali e remote.

Come parent
Utilizzare l'impostazione **SUBSCOPE** dell'argomento principale nella struttura ad albero degli argomenti.

Quando l'ambito della sottoscrizione per un argomento è impostato su TUTTO, direttamente o risolto tramite ASPARENT, le singole sottoscrizioni a tale argomento possono limitare il loro ambito a QMGR specificando MQSO_SCOPE_QMGR quando si crea la sottoscrizione. Una sottoscrizione a un argomento che ha un ambito QMGR non può ampliare l'ambito a ALL.

Concetti correlati

"Oggetti argomento di gestione" a pagina 78
Utilizzando un oggetto argomento di amministrazione, è possibile assegnare specifici attributi non predefiniti agli argomenti.

Attività correlate

Configurazione delle reti di pubblicazione / sottoscrizione distribuite

Spazi argomento

Uno spazio argomenti è una serie di argomenti su cui è possibile effettuare la sottoscrizione e la pubblicazione. Un gestore code in una topologia di pubblicazione / sottoscrizione distribuita ha uno spazio argomenti che potenzialmente include argomenti sottoscritti e pubblicati su gestori code connessi in tale topologia.

Nota: Per una panoramica degli argomenti all'interno di un gestore code, come gli oggetti argomento di gestione, le stringhe argomento e le strutture ad albero argomento, consultare"Argomenti" a pagina 69. Ulteriori riferimenti agli *argomenti* nell'articolo corrente fanno riferimento a *stringhe di argomenti* , se non diversamente specificato.

Gli argomenti vengono creati inizialmente in uno dei seguenti modi:

- amministrativamente, quando si definisce un oggetto argomento o una sottoscrizione durevole.
- dinamicamente, quando un'applicazione crea una pubblicazione o una sottoscrizione dinamicamente a un nuovo argomento.

Gli argomenti vengono propagati ad altri gestori code tramite sottoscrizioni proxy e creando oggetti argomento del cluster di gestione. Le sottoscrizioni proxy determinano l'inoltro delle pubblicazioni dal gestore code a cui è connesso un publisher ai gestori code dei sottoscrittori.

Le sottoscrizioni proxy vengono propagate tra tutti i gestori code connessi tra loro da relazioni padre - figlio in una gerarchia di gestori code. Il risultato è che è possibile sottoscrivere un gestore code a un argomento definito su qualsiasi altro gestore code nella gerarchia. Finché esiste un percorso connesso tra i gestori code, non importa come sono connessi i gestori code.

Le sottoscrizioni proxy vengono propagate anche per le sottoscrizioni agli argomenti cluster in un cluster di pubblicazione / sottoscrizione. Un argomento cluster è un argomento collegato a un oggetto argomento che ha l'attributo **CLUSTER** o eredita l'attributo dal relativo parent. Gli argomenti che non sono argomenti cluster sono noti come argomenti locali e non vengono replicati nel cluster. Nessuna sottoscrizione proxy viene propagata al cluster dalle sottoscrizioni agli argomenti locali.

Per riepilogare, le sottoscrizioni proxy vengono create per i sottoscrittori in due circostanze.

1. Un gestore code è un membro di una gerarchia e una sottoscrizione proxy viene inoltrata all'elemento principale e agli elementi secondari del gestore code.
2. Un gestore code è un membro di un cluster e la stringa dell'argomento della sottoscrizione si risolve in un argomento associato a un oggetto argomento del cluster. Quando l'argomento è un argomento cluster *diretto* , le sottoscrizioni proxy vengono inoltrate a tutti i membri del cluster. Quando l'argomento è un argomento cluster *host argomento instradato* , le sottoscrizioni proxy vengono

inoltrate solo ai gestori code nel cluster che hanno definito l'oggetto argomento cluster. Per ulteriori informazioni, consultare“Cluster di pubblicazione / sottoscrizione” a pagina 94.

Se un gestore code è un membro di un cluster e di una gerarchia, le sottoscrizioni proxy vengono propagate da entrambi i meccanismi senza consegnare le pubblicazioni duplicate al sottoscrittore.

Gli spazi argomenti di tre topologie di pubblicazione / sottoscrizione sono descritti nel seguente elenco:

- “Caso 1. Cluster di pubblicazione / sottoscrizione” a pagina 108.
- “Caso 2. Gerarchie di pubblicazione / sottoscrizione” a pagina 108.

In argomenti separati, le attività di configurazione riportate di seguito descrivono come combinare gli spazi argomento.

- Creazione di un singolo spazio argomento in un cluster di pubblicazione / sottoscrizione.
- Combinazione degli spazi argomento di più cluster.
- Combinazione e isolamento di spazi argomento in più cluster.
- Pubblicazione e sottoscrizione di spazi argomento in più cluster.

Caso 1. Cluster di pubblicazione / sottoscrizione

Nell'esempio, si supponga che il gestore code non sia connesso a una gerarchia di pubblicazione / sottoscrizione.

Se un gestore code è un membro di un cluster di pubblicazione / sottoscrizione, il relativo spazio argomenti è costituito da argomenti locali e argomenti cluster. Gli argomenti locali sono associati a oggetti argomento senza attributo **CLUSTER** . Se un gestore code dispone di definizioni di oggetti argomento locali, il relativo spazio argomento è diverso da un altro gestore code nel cluster che dispone anche di propri oggetti argomento definiti localmente.

In un cluster di pubblicazione / sottoscrizione, non è possibile sottoscrivere un argomento definito su un altro gestore code, a meno che l'argomento a cui si effettua la sottoscrizione non si risolva in un oggetto argomento cluster.

Quando le stesse definizioni di un oggetto argomento del cluster sono richieste su più gestori code, ad esempio quando si utilizza l' *instradamento host argomento*, è importante che tutte le definizioni corrispondano laddove necessario. Per ulteriori informazioni, vedi Creazione di un singolo spazio argomento in un cluster di pubblicazione / sottoscrizione.

Una definizione locale di un oggetto argomento, se la definizione è per un argomento cluster o per un argomento locale, ha la precedenza sullo stesso oggetto argomento definito altrove nel cluster. Viene utilizzato l'argomento definito localmente, anche se l'oggetto definito altrove è più recente.

È importante che un oggetto argomento cluster sia associato alla stessa stringa argomento ovunque nel cluster. Non è possibile modificare la stringa di argomenti a cui è associato un oggetto argomento. Per associare lo stesso oggetto argomento ad una stringa di argomenti differente, è necessario eliminare l'oggetto argomento e ricrearlo con la nuova stringa di argomenti. Se l'argomento è raggruppato in cluster, l'effetto è quello di eliminare le copie dell'oggetto argomento memorizzate sugli altri membri del cluster e quindi di creare copie del nuovo oggetto argomento ovunque nel cluster. Le copie dell'oggetto argomento si riferiscono tutte alla stessa stringa argomento.

È possibile creare accidentalmente due definizioni dello stesso oggetto argomento denominato su gestori code differenti nel cluster, con stringhe di argomenti differenti. Ciò può causare un comportamento confuso, poiché più definizioni dello stesso oggetto argomento con stringhe di argomento differenti possono produrre risultati differenti a seconda di come e dove si fa riferimento all'argomento. Per ulteriori informazioni su questo punto importante, consultare Definizioni di più argomenti cluster con lo stesso nome .

Caso 2. Gerarchie di pubblicazione / sottoscrizione

Nell'esempio, si presupponga che il gestore code non sia un membro di un cluster di pubblicazione / sottoscrizione.

In IBM MQ, se un gestore code è un membro di una gerarchia di pubblicazione / sottoscrizione, il relativo spazio argomenti è costituito da tutti gli argomenti definiti localmente e su gestori code connessi. Lo spazio argomento di tutti i gestori code in una gerarchia è lo stesso. Non vi è alcuna divisione degli argomenti in argomenti locali e globali.

Impostare una delle opzioni **PUBSCOPE** e **SUBSCOPE** su `QMGR`, per impedire che una pubblicazione su un argomento fluisce da un publisher a un sottoscrittore connesso a diversi gestori code nella gerarchia.

Si supponga di definire un oggetto argomento `Alabama` con la stringa argomento `USA/Alabama` sul gestore code `QMA`. Il risultato è il seguente:

1. Lo spazio argomenti in `QMA` ora include l'oggetto argomento `Alabama` e la stringa argomento `USA/Alabama`.
2. Un'applicazione o un amministratore possono creare una sottoscrizione in `QMA` utilizzando il nome oggetto argomento `Alabama`.
3. Un'applicazione può creare una sottoscrizione a qualsiasi argomento, incluso `USA/Alabama`, in qualsiasi gestore code nella gerarchia. Se `QMA` non è stato definito localmente, l'argomento `USA/Alabama` si risolve nell'oggetto argomento `SYSTEM.BASE.TOPIC`.

Concetti correlati

“Ambito della pubblicazione” a pagina 105
Quando si configura un cluster di pubblicazione / sottoscrizione o una gerarchia, l'ambito di una pubblicazione controlla ulteriormente se i gestori code inoltrano una pubblicazione ai gestori code remoti. Utilizzare l'attributo dell'argomento **PUBSCOPE** per gestire l'ambito delle pubblicazioni.

“Ambito della sottoscrizione” a pagina 106
L'ambito di una sottoscrizione controlla se una sottoscrizione su un gestore code riceve le pubblicazioni pubblicate su un altro gestore code in un cluster o gerarchia di pubblicazione / sottoscrizione oppure solo le pubblicazioni dei publisher locali.

Attività correlate

Configurazione delle reti di pubblicazione / sottoscrizione distribuite

IBM MQ Multicast

Il multicast IBM MQ offre un'affidabile messaggistica multicast a bassa latenza e ad elevato fanout.

Multicast è una forma efficiente di messaggistica di pubblicazione / sottoscrizione in quanto può essere scalata a un numero elevato di sottoscrittori senza effetti negativi sulle prestazioni. IBM MQ abilita la messaggistica Multicast affidabile utilizzando riconoscimenti, riconoscimenti negativi e numeri di sequenza per ottenere una messaggistica con latenza ridotta ed elevato fanout.

La consegna equa del multicast IBM MQ abilita una consegna quasi simultanea, garantendo così che nessun destinatario ottenga un vantaggio. Poiché il multicast IBM MQ si serve della rete per consegnare i messaggi, non è necessario un motore di pubblicazione/sottoscrizione per eseguire il fanout di dati. Dopo che un argomento è associato a un indirizzo di gruppo, non è necessario un gestore code perché i publisher e i sottoscrittori possono operare in modalità peer - to - peer. Ciò consente di ridurre il carico sui server del gestore code, pertanto tali server smettono di rappresentare un potenziale punto di errore.

Concetti multicast iniziali

IBM MQ Multicast può essere facilmente integrato in sistemi e applicazioni esistenti utilizzando l'oggetto COMMINFO (Communication Information). Due campi oggetto TOPIC consentono la configurazione rapida degli oggetti TOPIC esistenti per supportare o ignorare il traffico multicast.

Oggetti necessari per multicast

Le seguenti informazioni sono una breve panoramica dei due oggetti necessari per IBM MQ Multicast:

oggetto COMMINFO
L'oggetto COMMINFO contiene gli attributi associati alla trasmissione multicast. Per ulteriori informazioni sui parametri oggetto COMMINFO, consultare DEFINE COMMINFO.

L'unico campo COMMINFO che DEVE essere impostato è il nome dell'oggetto COMMINFO. Questo nome viene quindi utilizzato per identificare l'oggetto COMMINFO in un argomento. È necessario controllare il campo **GRPADDR** dell'oggetto COMMINFO per assicurarsi che il valore sia un indirizzo gruppo multicast valido.

Oggetto sezione
Un argomento è l'oggetto delle informazioni pubblicate in un messaggio di pubblicazione / sottoscrizione e un argomento viene definito creando un oggetto TOPIC. Per ulteriori informazioni sui parametri dell'oggetto TOPIC, consultare DEFINE TOPIC.

Gli argomenti esistenti possono essere utilizzati con il multicast modificando i valori dei seguenti parametri oggetto TOPIC: **COMMINFO** e **MCAST**.

- **COMMINFO** Questo parametro specifica il nome dell'oggetto delle informazioni di comunicazione multicast.
- **MCAST** Questo parametro specifica se il multicast è consentito in questa posizione nella struttura ad albero degli argomenti. Per impostazione predefinita, **MCAST** è impostato su ASPARENT , il che significa che l'attributo multicast dell'argomento viene ereditato dal parent. L'impostazione di **MCAST** su ENABLED consente il traffico multicast su questo nodo.

Argomenti e reti multicast

Le seguenti informazioni sono una panoramica di ciò che accade alle sottoscrizioni con diversi tipi di sottoscrizione e definizione argomento. Tutti questi esempi presuppongono che il parametro **COMMINFO** dell'oggetto TOPIC sia impostato sul nome di un oggetto COMMINFO valido:

Argomento impostato su multicast abilitato
Se il parametro della stringa di argomenti **MCAST** è impostato su ENABLED, le sottoscrizioni dai client con capacità multicast sono consentite e viene effettuata una sottoscrizione multicast a meno che:

- Si tratta di una sottoscrizione durevole da un client multicast.
- Si tratta di una sottoscrizione non gestita da un client multicast.
- Si tratta di una sottoscrizione da un client non multicast.

In questi casi viene effettuata una sottoscrizione non multicast e le sottoscrizioni vengono declassate alla normale pubblicazione / sottoscrizione.

Argomento impostato su multicast disabilitato
Se il parametro della stringa di argomenti **MCAST** è impostato su DISABLED, viene sempre effettuata una sottoscrizione non multicast e le sottoscrizioni vengono declassate alla normale pubblicazione / sottoscrizione.

Argomento impostato solo su multicast
Se il parametro **MCAST** della stringa dell'argomento è impostato su SOLO, le sottoscrizioni dai client con capacità multicast sono consentite e viene effettuata una sottoscrizione multicast a meno che:

- Si tratta di una sottoscrizione duratura: le sottoscrizioni durevoli vengono rifiutate con codice di errore 2436 (0984) (RC2436): MQRC_DURABILITY_NOT_ALLOWED
- È una sottoscrizione non gestita: le sottoscrizioni non gestite vengono rifiutate con codice motivo 2046 (07FE) (RC2046): MQRC_OPTIONS_ERROR
- Si tratta di una sottoscrizione da un client che non supporta il multicast: queste sottoscrizioni vengono rifiutate con il codice motivo 2560 (0A00) (RC2560): MQRC_MULTICAST_ONLY
- È una sottoscrizione da un'applicazione collegata localmente: queste sottoscrizioni vengono rifiutate con codice motivo 2560 (0A00) (RC2560): MQRC_MULTICAST_ONLY

Linux Windows AIX MQ TelemetryPanoramica

MQ Telemetry comprende un servizio di telemetria (MQXR) che fa parte di un gestore code, i client di telemetria che è possibile scrivere o scaricare gratuitamente e le interfacce di gestione della riga comandi ed explorer. La telemetria si riferisce alla raccolta di dati e alla gestione di una vasta gamma di dispositivi remoti. Con MQ Telemetry è possibile integrare la raccolta di dati e il controllo dei dispositivi con le applicazioni web.

MQ Telemetry è un componente di IBM MQ. L'aggiornamento per queste versioni consiste essenzialmente nell'installazione di una versione successiva di IBM MQ.

Le applicazioni di esempio continuano ad essere liberamente disponibili da Eclipse Paho e MQTT.org. Vedere IBM MQ Telemetry Transport programmi di esempio.

Poiché MQ Telemetry è un componente di IBM MQ, MQ Telemetry può essere installato con il prodotto principale o dopo che il prodotto principale è stato installato. Per informazioni sulla migrazione, consultare Migrazione MQ Telemetry su Windows e Migrazione MQ Telemetry su Linux.

Inclusi in MQ Telemetry sono i seguenti componenti:

Canali di telemetria
Utilizzare i canali di telemetria per gestire la connessione dei client MQTT a IBM MQ . I canali di telemetria utilizzano nuovi oggetti IBM MQ , come `SYSTEM.MQTT.TRANSMIT.QUEUE`, per interagire con IBM MQ.

Servizio di telemetria (MQXR)
I client MQTT utilizzano il servizio di telemetria `SYSTEM.MQXR.SERVICE` per connettersi ai canali di telemetria.

IBM MQ Explorer supporto per MQ Telemetry
MQ Telemetry può essere amministrato utilizzando IBM MQ Explorer.

Documentazione
La documentazione MQ Telemetry è inclusa nella documentazione del prodotto IBM MQ standard. La documentazione SDK per client Java e C viene fornita nella documentazione del prodotto e come Javadoc e HTML.

Concetti di telemetria

Si raccolgono informazioni dall'ambiente circostante per decidere cosa fare. Come consumatore, si controlla ciò che si ha in negozio, prima di decidere quale cibo acquistare. Vuoi sapere quanto tempo ci vorrà per un viaggio se te ne vai ora, prima di prenotare una connessione. Controlli i sintomi, prima di decidere se visitare il medico. Si controlla quando un autobus sta per arrivare, prima di decidere se attendere. Le informazioni per tali decisioni provengono direttamente dai contatori e dai dispositivi, dalla parola scritta su carta o da uno schermo e da te. Ovunque tu sia, e quando hai bisogno di farlo, raccogli informazioni, le metti insieme, le analizza e agisci su di esse.

Se le fonti di informazione sono ampiamente disperse o inaccessibili, diventa difficile e costoso raccogliere le informazioni più accurate. Se ci sono molte modifiche che si desidera apportare, o è difficile apportare le modifiche, le modifiche non vengono apportate o vengono apportate quando sono meno efficaci.

E se i costi di raccolta delle informazioni e di controllo dei dispositivi ampiamente dispersi fossero notevolmente ridotti collegando i dispositivi con la tecnologia digitale a Internet? Le informazioni possono essere analizzate utilizzando le risorse di Internet e dell'azienda. Hai più opportunità di prendere decisioni informate e agire su di esse.

Le tendenze tecnologiche e le pressioni ambientali ed economiche stanno portando a questi cambiamenti:

1. Il costo della connessione e del controllo di sensori e attuatori è in diminuzione, grazie alla standardizzazione e alla connessione a processori digitali a basso costo.

2. Internet, e le tecnologie di internet, sono sempre più utilizzati per collegare i dispositivi. In alcuni paesi, i telefoni cellulari superano i personal computer nel numero di connessioni alle applicazioni Internet. Altri dispositivi stanno sicuramente seguendo.
3. Internet, e le tecnologie di internet, rendono molto più facile per un'applicazione ottenere dati. Un facile accesso ai dati sta guidando l'utilizzo dell'analisi dei dati per trasformare i dati dai sensori in informazioni utili in molte altre soluzioni.
4. L'uso intelligente delle risorse è spesso un modo più rapido e meno costoso di ridurre le emissioni di carbonio e i costi. Le alternative: trovare nuove risorse o sviluppare nuove tecnologie per utilizzare le risorse esistenti potrebbero essere la soluzione a lungo termine. A breve termine lo sviluppo di nuove tecnologie o la ricerca di nuove risorse è spesso più rischioso, più lento e più costoso del miglioramento delle soluzioni esistenti.

Esempio

Un esempio mostra come queste tendenze creano nuove opportunità per interagire in modo intelligente con l'ambiente.

La Convenzione internazionale per la salvaguardia della vita umana in mare (SOLAS) prevede l'impiego di un sistema di identificazione automatica (AIS) su molte navi. È richiesto su navi mercantili di oltre 300 tonnellate e navi passeggeri. L'AIS è principalmente un sistema di prevenzione delle collisioni per la navigazione costiera. È utilizzato dalle autorità marittime per monitorare e controllare le acque costiere.

Gli appassionati di tutto il mondo stanno distribuendo stazioni di tracciamento AIS a basso costo e inserendo informazioni sulla navigazione costiera su Internet. Altri appassionati stanno scrivendo applicazioni che combinano informazioni da AIS con altre informazioni da Internet. I risultati vengono inseriti su siti Web e pubblicati tramite Twitter e SMS.

In un'applicazione, le informazioni provenienti dalle stazioni AIS vicino a Southampton sono combinate con la proprietà della nave e le informazioni geografiche. L'applicazione fornisce informazioni in tempo reale sugli arrivi e le partenze dei traghetti su Twitter. I pendolari regolari che utilizzano i traghetti tra Southampton e l'isola di Wight si abbonano al feed di notizie utilizzando Twitter o SMS. Se il feed mostra che il loro traghetto è in ritardo, i pendolari possono ritardare la loro partenza e prendere il traghetto quando attracca più tardi del suo orario di arrivo previsto.

Per ulteriori esempi, consultare “Casi di utilizzo della telemetria” a pagina 114.

Attività correlate

InstallazioneMQ Telemetry
AmministrazioneMQ Telemetry
Migrazione di MQ Telemetry su Windows
Migrazione di MQ Telemetry su Linux
Sviluppo di applicazioni per MQ Telemetry
Risoluzione dei problemi di MQ Telemetry

Riferimenti correlati

Riferimento di MQ Telemetry

Linux Windows AIX

Introduzione a MQ Telemetry

Le persone, le aziende e i governi vogliono sempre più utilizzare MQ Telemetry per interagire in modo più intelligente con l'ambiente in cui viviamo e lavoriamo. MQ Telemetry collega tutti i tipi di dispositivi a internet e all'azienda e riduce i costi di creazione di applicazioni per dispositivi intelligenti.

Che cos'è MQ Telemetry?

- È una funzione di IBM MQ che estende il backbone di messaggistica universale fornito da IBM MQ a una vasta gamma di sensori remoti, attuatori e dispositivi di telemetria. MQ Telemetry estende IBM MQ in modo da poter interconnettere le applicazioni aziendali intelligenti, i servizi e i responsabili delle decisioni con le reti di dispositivi strumentati.

- Le parti principali di MQ Telemetry sono:

  Il servizio MQ Telemetry (MQXR).
  Questo servizio viene eseguito nel server IBM MQ e utilizza il protocollo IBM MQ Telemetry Transport (MQTT) per comunicare con le periferiche di telemetria.

  Le applicazioni MQTT che si scrivono.
  Queste applicazioni controllano le informazioni trasmesse tra le periferiche di telemetria e il gestore code IBM MQ e tutte le azioni intraprese in risposta a tali informazioni. Per creare queste applicazioni, utilizzare le librerie client MQTT .

Cosa può fare per me?

- MQTT è un trasporto di messaggistica aperto che consente di creare implementazioni MQTT per un'ampia varietà di periferiche.
- I client MQTT possono essere eseguiti su dispositivi a ingombro ridotto con risorse limitate.
- MQTT funziona in modo efficiente su reti in cui la larghezza di banda è bassa, in cui i costi di invio dei dati sono costosi o che potrebbero essere fragili.
- La consegna dei messaggi è assicurata e disaccoppiata dall'applicazione.
- I programmatori di applicazioni non hanno bisogno di avere conoscenze di programmazione delle comunicazioni.
- I messaggi possono essere scambiati con altre applicazioni di messaggistica. Possono essere un'altra applicazione di telemetria o un'applicazione MQI, JMS o di messaggistica aziendale.

Come lo uso?

- Vengono forniti script di esempio che funzionano con un'applicazione client IBM MQ Telemetry Transport v3 di esempio (`mqttv3app.jar`). Vedere IBM MQ Telemetry Transport programmi di esempio.
- Utilizzare IBM MQ Explorer e gli strumenti associati per gestire la funzione di telemetria di IBM MQ.
- Utilizzare le librerie client per creare applicazioni MQTT che si connettono a un gestore code e che utilizzano la messaggistica di pubblicazione / sottoscrizione.
- Distribuire l'applicazione e la libreria client al dispositivo in cui deve essere eseguita l'applicazione.

Come funziona?

- MQTT è un protocollo di pubblicazione - sottoscrizione. Un'applicazione client MQTT può pubblicare messaggi su un server MQTT o sottoscrivere messaggi inviati dalle applicazioni che si collegano a un server MQTT .
- Le applicazioni client MQTT utilizzano librerie client che implementano il trasporto del messaggio MQTT .
- Un'applicazione client MQTT di base funziona come un client MQ standard ma può essere eseguita su una varietà molto più ampia di piattaforme e reti.
- Il servizio MQ Telemetry (MQXR) trasforma un gestore code IBM MQ in un server MQTT .
- Quando un gestore code IBM MQ agisce come server MQTT , altre applicazioni che si connettono al gestore code possono sottoscrivere e ricevere i messaggi dal client MQTT .
- Il gestore code agisce come router distribuendo i messaggi dalle applicazioni di pubblicazione alle applicazioni di sottoscrizione.
- I messaggi possono essere distribuiti tra diversi tipi di applicazioni client. Ad esempio, tra i client Telemetry e i client JMS .

Nota: MQ Telemetry sostituisce i nodi SCADA ritirati nella versione 7 di WebSphere Message Broker (ora noti come IBM Integration Bus) e viene eseguito su Windows, Linuxe AIX.

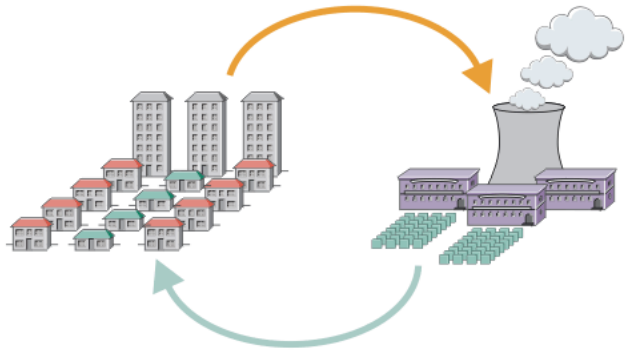
Linux Windows AIX Casi di utilizzo della telemetria

Telemetria è il rilevamento automatico, la misura dei dati e il controllo dei dispositivi remoti. L'accento è posto sulla trasmissione di dati da dispositivi a un punto di controllo centrale. La telemetria comprende anche l'invio di informazioni di controllo e di configurazione ai dispositivi.

MQ Telemetry connette piccoli dispositivi utilizzando MQTT protocole collega i dispositivi ad altre applicazioni utilizzando IBM MQ. MQ Telemetry colma un divario tra i dispositivi e Internet rendendo più semplice la creazione di "soluzioni intelligenti". Le soluzioni intelligenti sbloccano la ricchezza di informazioni disponibili su Internet e nelle applicazioni aziendali, per le applicazioni che monitorano e controllano i dispositivi.

I seguenti diagrammi illustrano alcuni utilizzi tipici di MQ Telemetry:

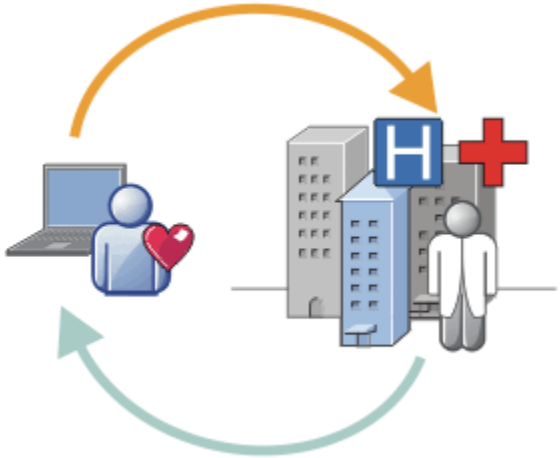
Telemetria: elettricità intelligente



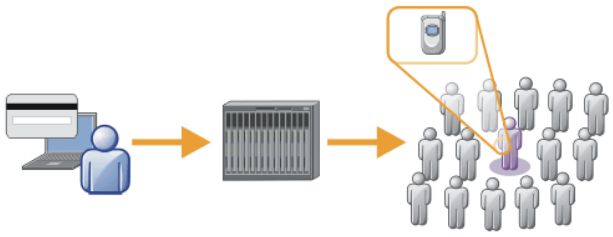
- Messaggio MQTT contenente i dati di utilizzo dell'energia inviati al service provider.
- MQ Telemetry invia i COMANDI DI CONTROLLO in base all'analisi dei dati di utilizzo dell'energia.
- Per ulteriori informazioni, consultare il seguente caso d'uso: “Caso d'uso della telemetria: monitoraggio e controllo dell'energia domestica” a pagina 116

Telemetria: servizi sanitari intelligenti

- MQ Telemetry invia i dati sanitari all'ospedale e al medico.
- Gli avvisi o il feedback dei messaggi MQTT possono essere inviati in base all'analisi dei dati di integrità.
- Per ulteriori informazioni, consultare il seguente caso d'uso: “Caso d'uso della telemetria: monitoraggio dei pazienti a domicilio” a pagina 115



Telemetria: Uno in mezzo alla folla



- Una semplice transazione con carta viene inviata al server della banca.
- MQ Telemetry identifica la persona tra le migliaia, avvisando il cliente che la sua carta è stata utilizzata.
- MQ Telemetry può utilizzare l'input più semplice di informazioni e individuare tale individuo.

I casi di utilizzo descritti nei sottoargomenti sono tratti da esempi reali. Essi illustrano alcuni modi di utilizzare la telemetria e alcuni dei problemi comuni che la tecnologia di telemetria deve risolvere.

Linux Windows AIX Caso d'uso della telemetria: monitoraggio dei pazienti a domicilio

In collaborazione tra IBM e un fornitore di assistenza sanitaria su un sistema di cura del paziente cardiaco, un defibrillatore cardioverter impiantato comunica con un ospedale. I dati sul paziente e sul dispositivo impiantato vengono trasferiti utilizzando la telemetria RF al dispositivo MQTT nella casa di un paziente.

Tipicamente il trasferimento avviene di notte a un trasmettitore situato sul lato del letto. Il trasmettitore trasferisce i dati in modo sicuro attraverso il sistema telefonico all'ospedale, dove i dati vengono analizzati.

Il sistema riduce il numero di visite che un paziente deve effettuare al medico. Rileva quando il paziente o il dispositivo ha bisogno di attenzione e, in caso di emergenza, avvisa il medico di guardia.

La collaborazione tra IBM e il fornitore di assistenza sanitaria ha caratteristiche comuni a diversi casi di utilizzo della telemetria:

Invisibilità

L'unità non richiede alcun intervento dell'utente se non la fornitura di alimentazione, una linea telefonica e la vicinanza all'unità per una parte della giornata. Il suo funzionamento è affidabile e semplice da usare.

Per rimuovere la necessità per il paziente di configurare il dispositivo, il fornitore del dispositivo preconfigura il dispositivo. Il paziente deve solo collegarlo. L'eliminazione della configurazione da parte del paziente semplifica il funzionamento del dispositivo e riduce la possibilità che il dispositivo venga configurato in modo errato.

Il client MQTT è integrato come parte della periferica. Lo sviluppatore del dispositivo integra l'implementazione del client MQTT nel dispositivo e lo sviluppatore o il fornitore configura il client MQTT come parte della pre - configurazione.

Il client MQTT viene fornito come un file JAR Java SE , che lo sviluppo include nella propria applicazione Java . Per ambienti nonJava , come questo, lo sviluppatore del dispositivo può implementare un client in un linguaggio diverso utilizzando il protocollo e formati MQTT pubblicati. In alternativa, lo sviluppatore può utilizzare uno dei client C forniti come librerie condivise per le piattaforme Windows, Linux e ARM.

Connettività irregolare

La comunicazione tra il defibrillatore e l'ospedale ha caratteristiche di rete irregolari. Due diverse reti vengono utilizzate per risolvere i diversi problemi di raccolta dei dati dal paziente e di invio dei dati all'ospedale. Tra il brevetto e il dispositivo MQTT , viene utilizzata una rete RF a bassa potenza a corto raggio. Il trasmettitore si connette all'ospedale utilizzando una connessione TCP/IP VPN su una linea telefonica a bassa larghezza di banda.

Spesso non è pratico trovare un modo per collegare ogni dispositivo direttamente a una rete Internet Protocol . Utilizzare due reti, collegate da un hub, è una soluzione comune. Il dispositivo MQTT è un hub semplice, che memorizza le informazioni dal paziente e le inoltra all'ospedale.

Sicurezza

Il medico deve essere in grado di fidarsi dell'autenticità dei dati del paziente e il paziente vuole che venga rispettata la privacy dei propri dati.

In alcune situazioni è sufficiente crittografare la connessione, utilizzando VPN o TLS. In altre situazioni, è auspicabile mantenere i dati al sicuro anche dopo che sono stati archiviati.

A volte il dispositivo di telemetria non è sicuro. Potrebbe essere in un'abitazione condivisa, ad esempio. L'utente del dispositivo deve essere autenticato per assicurarsi che i dati provengano dal

paziente corretto. Il dispositivo stesso può essere autenticato sul server utilizzando TLS e il server autenticato sul dispositivo.

Il canale di telemetria tra il dispositivo e il gestore code supporta JAAS per l'autenticazione utente e TLS per la crittografia delle comunicazioni e l'autenticazione del dispositivo. L'accesso a una pubblicazione è controllato dal gestore autorizzazione oggetto in IBM MQ.

L'identificativo utilizzato per autenticare l'utente può essere associato a un altro identificativo, ad esempio un'identità comune del paziente. Un identificativo comune semplifica la configurazione dell'autorizzazione per gli argomenti di pubblicazione in IBM MQ.

Connettività

La connessione tra la periferica MQTT e l'ospedale utilizza la connessione remota e funziona con una larghezza di banda inferiore a 300 baud.

Per operare efficacemente a 300 baud, MQTT protocol aggiunge solo pochi byte aggiuntivi a un messaggio oltre alle intestazioni TCP/IP.

MQTT protocol fornisce la messaggistica *fire and forget* a trasmissione singola, che mantiene basse le latenze. Può anche utilizzare più trasmissioni per garantire *almeno una volta* e *esattamente una volta* la consegna se la consegna garantita è più importante del tempo di risposta. Per garantire la consegna, i messaggi vengono memorizzati sul dispositivo fino a quando non vengono consegnati correttamente. Se un dispositivo è collegato in modalità wireless, la consegna garantita è particolarmente utile.

Scalabilità

I dispositivi di telemetria sono in genere distribuiti in grandi quantità, da decine di migliaia a milioni.

Il collegamento di molti dispositivi a un sistema richiede una soluzione di grandi dimensioni. Ci sono richieste di business come il costo dei dispositivi e il loro software, e le richieste di amministrazione di gestire licenze, dispositivi e utenti. I requisiti tecnici includono il carico sulla rete e sui server.

L'apertura delle connessioni utilizza più risorse del server rispetto alla gestione delle connessioni aperte. Ma in un caso d'uso come questo che utilizza le linee telefoniche, la spesa delle connessioni significa che le connessioni vengono lasciate aperte non più del necessario. I trasferimenti di dati sono in gran parte di natura batch. I collegamenti possono essere programmati per tutta la notte per evitare un picco improvviso di collegamenti al momento di coricarsi.

Sul client, la scalabilità dei client è agevolata dalla configurazione client minima richiesta. Il client MQTT è integrato nella periferica. Non c'è alcun requisito per una fase di configurazione o di accettazione della licenza client MQTT da incorporare nella distribuzione dei dispositivi ai pazienti.

Sul server, MQ Telemetry ha una destinazione iniziale di 50.000 connessioni aperte per gestore code.

Le connessioni sono gestite utilizzando IBM MQ Explorer. IBM MQ Explorer filtra le connessioni da visualizzare in un numero gestibile. Con uno schema scelto in modo appropriato di assegnare identificativi ai client, è possibile filtrare le connessioni in base all'area geografica o in ordine alfabetico in base al nome del paziente.

Linux Windows AIX Caso d'uso della telemetria: monitoraggio e controllo dell'energia domestica

I contatori intelligenti raccolgono maggiori dettagli sul consumo di energia rispetto ai contatori tradizionali.

I contatori intelligenti sono spesso accoppiati con una rete di telemetria locale per monitorare e controllare i singoli elettrodomestici in una casa. Alcuni sono anche collegati da remoto per il monitoraggio e il controllo a distanza.

La connessione remota può essere impostata da un individuo, da un programma di utilità o da un punto di controllo centrale. Il punto di controllo remoto può leggere l'utilizzo di energia e fornire i dati di utilizzo. Può fornire dati per influenzare l'utilizzo, come prezzi continui e informazioni meteorologiche. Può limitare il carico per migliorare l'efficienza generale della generazione di energia.

I contatori intelligenti stanno iniziando a essere ampiamente utilizzati. Il governo del Regno Unito, ad esempio, è in consultazione per l'introduzione di contatori intelligenti in ogni casa del Regno Unito entro il 2020.

I casi di utilizzo della misurazione a domicilio hanno una serie di caratteristiche comuni:

Invisibilità

A meno che l'utente non desideri essere coinvolto nel risparmio energetico utilizzando il contatore, il contatore non deve richiedere l'intervento dell'utente. Non deve ridurre l'affidabilità dell'approvvigionamento energetico dei singoli apparecchi.

Un client MQTT può essere incorporato nel software installato con il misuratore e non richiede un'installazione o una configurazione separate.

Connettività irregolare

La comunicazione tra le apparecchiature e il contatore intelligente richiede standard di connettività diversi rispetto a quelli tra il contatore e il punto di connessione remoto.

La connessione dal contatore intelligente alle apparecchiature deve essere altamente disponibile e conforme agli standard di rete per una rete domestica.

È probabile che la rete remota utilizzi varie connessioni fisiche. Alcuni di loro, come il cellulare, hanno un costo di trasmissione elevato e possono essere intermittenti. La specifica MQTT v3 è rivolta alle connessioni remote e alle connessioni tra gli adattatori locali e lo smart meter.

La connessione tra le prese di corrente e le applicazioni, e il contatore, utilizza una rete di aree domestiche, come Zigbee. MQTT per reti di sensori (MQTT-S), è progettato per funzionare con Zigbee e altri protocolli di rete a bassa larghezza di banda. MQ Telemetry non supporta MQTT-S direttamente. Richiede un gateway per collegare MQTT-S a MQTT v3.

Come il monitoraggio del paziente a casa, le soluzioni per il monitoraggio e il controllo dell'energia a casa richiedono reti multiple, connesse utilizzando lo smart meter come hub.

Sicurezza

Esistono diversi problemi di sicurezza associati ai contatori intelligenti. Questi problemi includono il non rifiuto delle transazioni, l'autorizzazione di eventuali azioni di controllo avviate e la riservatezza dei dati sul consumo di energia.

Per garantire la privacy, i dati trasferiti tra il contatore e il punto di controllo remoto da MQTT possono essere codificati utilizzando TLS. Per garantire l'autorizzazione delle azioni di controllo, la connessione MQTT tra il misuratore e il punto di controllo remoto può essere autenticata reciprocamente utilizzando TLS.

Connettività

La natura fisica della rete remota può variare notevolmente. Potrebbe utilizzare una connessione a banda larga esistente o utilizzare una rete mobile con costi di chiamata elevati e disponibilità intermittente. Per connessioni ad alto costo, intermittenti, MQTT è un protocollo efficiente e affidabile; consultare “Caso d'uso della telemetria: monitoraggio dei pazienti a domicilio” a pagina 115.

Scalabilità

Alla fine le aziende elettriche, o punti di controllo centrali, pianificano di implementare decine di milioni di contatori intelligenti. Inizialmente, il numero di metri per distribuzione è tra le decine e le centinaia di migliaia. Questo numero è paragonabile alla destinazione MQTT iniziale di 50.000 connessioni client aperte per gestore code.

Un aspetto critico dell'architettura per il monitoraggio e il controllo dell'energia domestica è quello di utilizzare il contatore intelligente come concentratore di rete. Ogni adattatore del dispositivo è un sensore separato. Collegandoli a un hub locale utilizzando MQTT, l'hub può concentrare i flussi di dati su una singola sessione TCP/IP con il punto di controllo centrale e memorizzare i messaggi per un breve periodo per superare le interruzioni di sessione.

Le connessioni remote devono essere lasciate aperte nei casi di utilizzo dell'energia domestica per due motivi. Innanzitutto, perché l'apertura delle connessioni richiede molto tempo rispetto all'invio delle richieste. Il tempo per aprire molte connessioni per inviare richieste di "limitazione del carico" in un breve intervallo è troppo lungo. In secondo luogo, per ricevere le richieste di limitazione del carico dalla società elettrica, la connessione deve essere prima aperta dal client. Con MQTT, le connessioni vengono sempre avviate dal client e, per ricevere le richieste di limitazione del carico dalla società elettrica, la connessione deve essere lasciata aperta.

Se la frequenza di apertura delle connessioni è critica, o se il server avvia richieste di tipo time - critical, la soluzione è in genere quella di mantenere molte connessioni aperte.

Linux Windows AIX Casi di utilizzo della telemetria: RFID (Radio Frequency Identification)

RFID è l'uso di un tag RFID incorporato per identificare e tenere traccia di un oggetto in modalità wireless. I tag RFID possono essere letti fino a una gamma di diversi metri e fuori dalla linea di vista del lettore RFID. I tag passivi sono attivati da un lettore RFID. I tag attivi vengono trasmessi senza attivazione esterna. I tag attivi devono avere una fonte di energia. I tag passivi possono includere una fonte di alimentazione per aumentare il loro intervallo.

L'RFID è utilizzato in molte applicazioni e i tipi di casi d'uso variano enormemente. I casi d'uso RFID e il monitoraggio dei pazienti a domicilio e il monitoraggio dell'energia a domicilio e i casi d'uso di controllo, hanno alcune somiglianze e differenze.

Invisibilità

In molti casi di utilizzo, il lettore RFID è distribuito in grandi quantità e deve funzionare senza l'intervento dell'utente. Il lettore include un client MQTT integrato per comunicare con un punto di controllo centrale.

Ad esempio, in un magazzino di distribuzione, un lettore utilizza un sensore di movimento per rilevare un pallet. Attiva i tag RFID degli elementi sul pallet e invia dati e richieste alle applicazioni centrali. I dati vengono utilizzati per aggiornare l'ubicazione dello stock. Le richieste controllano ciò che accade al pallet successivo, ad esempio spostarlo in una particolare baia. Le compagnie aeree e i sistemi di bagagli aeroportuali utilizzano la tecnologia RFID in questo modo.

In alcuni casi d'uso RFID, il lettore dispone di un ambiente di elaborazione standard, come Java Platform, Micro Edition (Java ME). In questi casi, il client MQTT potrebbe essere distribuito in un passo di configurazione distinto, dopo la produzione.

Connettività irregolare

I lettori RFID potrebbero essere separati dalla periferica di controllo locale che contiene un client MQTT oppure ogni lettore potrebbe incorporare un client MQTT . In genere, i fattori geografici o di comunicazione indicano la scelta della topologia.

Sicurezza

Privacy e autenticità sono problemi di sicurezza nell'allegato dei tag RFID. I tag RFID sono discreti e possono essere monitorati, falsificati o manomessi.

La soluzione dei problemi di sicurezza RFID aumenta l'opportunità di implementare nuove soluzioni RFID. Sebbene l'esposizione alla sicurezza sia nel tag RFID e nel lettore locale, l'utilizzo dell'elaborazione centrale delle informazioni suggerisce approcci per contrastare diverse minacce. Ad esempio, la manomissione di tag potrebbe essere rilevata correlando dinamicamente i livelli di scorte rispetto alle consegne e alle spedizioni.

Connettività

Le applicazioni RFID in genere coinvolgono sia l'archiviazione in batch che l'inoltro delle informazioni raccolte dai lettori RFID e dalle query immediate. Nel caso d'uso del magazzino di distribuzione, il lettore RFID è sempre collegato. Quando un tag viene letto, viene pubblicato insieme alle informazioni sul lettore. L'applicazione di archiviazione pubblica la risposta al lettore.

Nell'applicazione di archiviazione la rete è in genere affidabile e le richieste immediate potrebbero utilizzare i messaggi *fire and forget* (attiva e dimentica) per prestazioni a bassa latenza. I dati di archiviazione e inoltro in batch potrebbero utilizzare la messaggistica *esattamente una volta* per ridurre al minimo i costi di gestione associati alla perdita di dati.

Scalabilità

Se l'applicazione RFID richiede risposte immediate, nell'ordine di uno o due secondi, i lettori RFID devono rimanere connessi.

 Casi di utilizzo della telemetria: rilevamento dell'ambiente

Il rilevamento ambientale utilizza la telemetria per raccogliere informazioni sui livelli e sulla qualità delle acque fluviali, sugli inquinanti atmosferici e su altri dati ambientali.

I sensori si trovano spesso in luoghi remoti, senza accesso alla comunicazione cablata. La larghezza di banda wireless è costosa e l'affidabilità può essere bassa. In genere, una serie di sensori di ambiente in una piccola area geografica sono collegati a un dispositivo di monitoraggio locale in una posizione sicura. Le connessioni locali potrebbero essere cablate o wireless.

Invisibilità

È probabile che i dispositivi del sensore siano meno accessibili, meno alimentati e distribuiti in numero maggiore rispetto al dispositivo di monitoraggio centrale. I sensori sono a volte "stupidi" e il dispositivo di monitoraggio locale include adattatori per trasformare e memorizzare i dati del sensore. È probabile che il dispositivo di monitoraggio incorpori un computer di uso generale che supporta Java Platform, Standard Edition (Java SE) o Java Platform, Micro Edition (Java ME). È improbabile che l'invisibilità sia un requisito importante quando si configura il client MQTT .

Connettività irregolare

Le funzionalità dei sensori e il costo e la larghezza di banda della connessione remota, di solito si traduce in un hub di monitoraggio locale connesso a un server centrale.

Sicurezza

A meno che la soluzione non venga utilizzata in un caso di uso militare o difensivo, la sicurezza non è un requisito importante.

Connettività

Molti utilizzi non richiedono un monitoraggio continuo o una disponibilità immediata dei dati. I dati di eccezione, come un avviso di livello di inondazione, devono essere inoltrati immediatamente. I dati del sensore vengono aggregati sul monitor locale per ridurre i costi di connessione e comunicazione e quindi trasferiti utilizzando le connessioni pianificate. I dati di eccezione vengono inoltrati non appena vengono rilevati sul monitor.

Scalabilità

I sensori sono concentrati intorno agli hub locali e i dati del sensore vengono aggregati in pacchetti che vengono trasmessi in base a una pianificazione. Entrambi questi fattori riducono il carico sul server centrale che verrebbe imposto utilizzando sensori collegati direttamente.

 Casi di utilizzo della telemetria: applicazioni mobili

Le applicazioni mobili sono applicazioni che vengono eseguite su dispositivi wireless. I dispositivi sono piattaforme di applicazioni generiche o dispositivi personalizzati.

Le piattaforme generali includono dispositivi portatili come telefoni e assistenti di dati personali e dispositivi portatili come computer notebook. I dispositivi personalizzati utilizzano hardware per scopi speciali su misura per applicazioni specifiche. Un dispositivo per registrare la consegna dei pacchi "signed-for" è un esempio di dispositivo mobile personalizzato. Le applicazioni su dispositivi mobili personalizzati sono spesso costruite su una piattaforma software generica.

Invisibilità

La distribuzione delle applicazioni mobili personalizzate viene gestita e può includere la configurazione dell'applicazione client MQTT . È improbabile che l'invisibilità sia un requisito importante quando si configura il client MQTT .

Connettività irregolare

A differenza della topologia dell'hub locale dei precedenti casi di utilizzo, i client mobili si collegano in remoto. Il livello dell'applicazione client si connette direttamente a un'applicazione nell'hub centrale.

Sicurezza

Con poca sicurezza fisica, il dispositivo mobile e l'utente mobile devono essere autenticati. TLS viene utilizzato per confermare l'identità della periferica e JAAS per autenticare l'utente.

Connettività

Se l'applicazione mobile dipende dalla copertura wireless, deve essere in grado di operare offline e di gestire in maniera efficiente una connessione interrotta. In questo ambiente, l'obiettivo è rimanere connessi, ma l'applicazione deve essere in grado di memorizzare e inoltrare i messaggi. Spesso i messaggi sono ordini, o conferme di consegna, e hanno un valore di business importante. Devono essere archiviati e inoltrati in modo affidabile.

Scalabilità

La scalabilità non è un problema importante. È probabile che il numero di client applicativi non superi le migliaia, o le decine di migliaia, nei casi di utilizzo di applicazioni mobili personalizzate.

Linux Windows AIX Connessione di dispositivi di telemetria a un gestore code

I dispositivi di telemetria si collegano a un gestore code utilizzando un client MQTT v3 . Il client MQTT v3 utilizza TCP/IP per connettersi a un listener TCP/IP denominato servizio di telemetria (MQXR).

Quando si connette un dispositivo di telemetria a un gestore code, il client MQTT avvia una connessione TCP/IP utilizzando il metodo `MqttClient.connect` . Come i client IBM MQ , un client MQTT deve essere connesso al gestore code per inviare e ricevere messaggi. La connessione viene effettuata sul server utilizzando un listener TCP/IP, installato con MQ Telemetry, denominato servizio di telemetria (MQXR). Ogni gestore code esegue un massimo di un servizio di telemetria (MQXR).

Il servizio di telemetria (MQXR) utilizza l'indirizzo socket remoto impostato da ciascun client nel metodo `MqttClient.connect` per assegnare la connessione a un canale di telemetria. Un indirizzo socket è la combinazione di nome host TCP/IP e numero di porta. Più client che utilizzano lo stesso indirizzo socket remoto sono connessi allo stesso canale di telemetria dal servizio di telemetria (MQXR).

Se ci sono più gestori code su un server, suddividere i canali di telemetria tra i gestori code. Allocare gli indirizzi socket remoti tra i gestori code. Definire ogni canale di telemetria con un indirizzo socket remoto univoco. Due canali di telemetria non devono utilizzare lo stesso indirizzo socket.

Se lo stesso indirizzo del socket remoto è configurato per i canali di telemetria su più gestori code, vince il primo canale di telemetria a connettersi. I canali successivi che si collegano sullo stesso indirizzo non riescono.

Se sul server sono presenti più adattatori di rete, suddividere gli indirizzi socket remoti tra i canali di telemetria. L'assegnazione degli indirizzi socket è del tutto arbitraria, purché qualsiasi indirizzo socket specifico sia configurato su un solo canale di telemetria.

Configurare IBM MQ per connettere i client MQTT utilizzando le procedure guidate fornite nel supplemento MQ Telemetry per IBM MQ Explorer. In alternativa, seguire le istruzioni riportate in Configurazione di un gestore code per la telemetria su Linux e AIX e Configurazione di un gestore code per la telemetria su Windows per configurare la telemetria manualmente.

Riferimenti correlati

Proprietà MQXR

Linux Windows AIX Protocolli di connessione di telemetria

MQ Telemetry supporta TCP/IP IPv4 e IPv6e TLS.

Linux Windows AIX Servizio di telemetria (MQXR)

Il servizio di telemetria (MQXR) è un listener TCP/IP, gestito come servizio IBM MQ . Creare il servizio utilizzando una procedura guidata IBM MQ Explorer o con un comando **runmqsc** .

Il servizio MQ Telemetry (MQXR) è denominato SYSTEM.MQXR.SERVICE .

La procedura guidata di configurazione dell'esempio di telemetria, fornita in MQ Telemetry funzione per IBM MQ Explorer, crea il servizio di telemetria e un canale di telemetria di esempio; consultare Verifica dell'installazione di MQ Telemetry utilizzando IBM MQ Explorer .

Creare la configurazione di esempio dalla linea di comando; consultare Verifica dell'installazione di MQ Telemetry utilizzando la riga di comando.

Il servizio di telemetria (MQXR) viene avviato e arrestato automaticamente con il gestore code. Controllare il servizio utilizzando la cartella dei servizi in IBM MQ Explorer. Per vedere il servizio, devi fare clic sull'icona per arrestare IBM MQ Explorer il filtraggio degli oggetti SYSTEM dalla visualizzazione.

Per un esempio di come creare il servizio manualmente, consultare

- Linux AIX Creazione di SYSTEM.MQXR.SERVICE su Linux.
- Windows Creazione di SYSTEM.MQXR.SERVICE su Windows.

V 9.3.0 Da IBM MQ 9.3.0 in poi, Creazione di SYSTEM.MQXR.SERVICE su Linuxe Creazione di SYSTEM.MQXR.SERVICE su Windows vengono aggiornati per specificare la chiave predefinita per richiedere passphrase per i canali MQTT TLS da codificare. Per ulteriori informazioni, vedi Encryption of passphrase for MQTT TLS channels.

Linux Windows AIX Canali di telemetria

Creare canali di telemetria per creare connessioni con proprietà differenti, come JAAS (Java Authentication and Authorization Service) o autenticazione TLS oppure per gestire gruppi di client.

Creare canali di telemetria utilizzando la procedura guidata **New Telemetry Channel** , fornita nella funzione MQ Telemetry per IBM MQ Explorer. Configurare un canale, utilizzando il wizard, per accettare connessioni dai client MQTT su una particolare porta TCP/IP. Da IBM WebSphere MQ 7.1, è possibile configurare MQ Telemetry utilizzando il programma della riga comandi, **runmqsc**.

Creare più canali di telemetria, su porte diverse, per rendere più semplice la gestione di un gran numero di connessioni client, suddividendo i client in gruppi. Ogni canale di telemetria ha un nome differente.

È possibile configurare canali di telemetria con attributi di sicurezza differenti per creare diversi tipi di connessione. Creare più canali per accettare connessioni client su indirizzi TCP/IP differenti. Utilizzare TLS per crittografare i messaggi e autenticare il client e il canale di telemetria; consultare Configurazione TLS dei client MQTT e dei canali di telemetria. Specificare l'ID utente per semplificare l'autorizzazione dell'accesso agli oggetti IBM MQ . Specificare una configurazione JAAS per autenticare l'utente MQTT con JAAS; consultare MQTT client identification, authorization, and authentication.

Linux Windows AIX IBM MQ Telemetry Transport protocollo

Il protocollo IBM MQ Telemetry Transport (MQTT) v3 è progettato per lo scambio di messaggi tra periferiche di piccole dimensioni su una larghezza di banda bassa o su connessioni costose e per inviare messaggi in modo affidabile. Utilizza TCP/IP.

MQTT protocol è pubblicato; consultare IBM MQ Telemetry Transport formato e protocollo. La Versione 3 del protocollo utilizza la pubblicazione / sottoscrizione e supporta tre QOS (quality of service): *attiva e dimentica*, *almeno una volta*e *esattamente una volta*.

La dimensione ridotta delle intestazioni del protocollo e il payload del messaggio dell'array di byte, mantengono i messaggi piccoli. Le intestazioni comprendono un'intestazione fissa a 2 byte e fino a 12 byte di intestazioni variabili aggiuntive. Il protocollo utilizza intestazioni variabili a 12 byte per la sottoscrizione e la connessione e solo intestazioni variabili a 2 byte per la maggior parte delle pubblicazioni.

Con tre QoS (quality of service), puoi scambiare tra bassa latenza e affidabilità; vedi Qualità del servizio fornito da un client MQTT. *Fire and forget* non utilizza alcuna memoria del dispositivo persistente e solo una trasmissione per inviare o ricevere una pubblicazione. *Almeno una volta*e *esattamente una volta* richiedono l'archiviazione persistente sul dispositivo per mantenere lo stato del protocollo e salvare un messaggio fino a quando non viene riconosciuto.

Linux Windows AIX MQTT client

Un'app del client MQTT è responsabile della raccolta delle informazioni dal dispositivo di telemetria, della connessione al server e della pubblicazione delle informazioni sul server. Può anche sottoscrivere argomenti, ricevere pubblicazioni e controllare il dispositivo di telemetria.

A differenza delle applicazioni client IBM MQ , le MQTT applicazioni client non sono applicazioni IBM MQ . Non specificano un gestore code a cui connettersi. Non sono limitati all'utilizzo di interfacce di programmazione IBM MQ . Invece, i client MQTT implementano il protocollo MQTT 3 . È possibile scrivere la libreria client per interfacciarsi con MQTT protocol nel linguaggio di programmazione e sulla piattaforma di propria scelta. Vedere IBM MQ Telemetry Transport formato e protocollo.

Per semplificare la scrittura di applicazioni client MQTT , utilizzare le librerie client C, Javae JavaScript che incapsulano MQTT protocol per un numero di piattaforme. Se incorpori queste librerie nelle tue applicazioni MQTT , un client MQTT completamente funzionale può essere breve fino a 15 righe di codice. Le librerie client MQTT sono disponibili gratuitamente da Eclipse Paho e MQTT.org. Vedere IBM MQ Telemetry Transport programmi di esempio.

L'applicazione del client MQTT è sempre responsabile dell'avvio di un collegamento con un canale di telemetria. Dopo la connessione, l'app client MQTT o un'applicazione IBM MQ possono avviare uno scambio di messaggi.

Le applicazioni client MQTT e le applicazioni IBM MQ pubblicano e sottoscrivono la stessa serie di argomenti. Un'applicazione IBM MQ può anche inviare un messaggio direttamente a un'applicazione client MQTT senza che l'applicazione client crei prima una sottoscrizione. Consultare Configurazione dell'accodamento distribuito per inviare messaggi ai client MQTT.

Le applicazioni client MQTT vengono connesse a IBM MQ utilizzando un canale di telemetria. Il canale di telemetria funge da ponte tra i diversi tipi di messaggi utilizzati da MQTT e IBM MQ. Crea pubblicazioni e sottoscrizioni nel gestore code per conto dell'applicazione client MQTT . Il canale di telemetria invia pubblicazioni che corrispondono alle sottoscrizioni di un'app client MQTT dal gestore code all'applicazione client MQTT .

Linux Windows AIX Invio di un messaggio a un client MQTT

Le applicazioni IBM MQ possono inviare i messaggi dei client MQTT v3 mediante la pubblicazione su sottoscrizioni create dai client o inviando direttamente i messaggi. I client MQTT possono inviare messaggi tra loro pubblicando gli argomenti sottoscritti da altri client.

Un client MQTT sottoscrive una pubblicazione, che riceve da IBM MQ

Eseguire l'attività “Pubblicazione di un messaggio nel programma di utilità del client MQTT da IBM MQ Explorer” a pagina 124 per inviare una pubblicazione da IBM MQ a un client MQTT .

Il modo standard per un client MQTT v3 per ricevere i messaggi consiste nel creare una sottoscrizione a un argomento o a una serie di argomenti. Nel frammento di codice di esempio, Figura 44 a pagina 123, il client MQTT effettua la sottoscrizione utilizzando la stringa argomento `"MQTT Examples"`. Un'applicazione IBM MQ C, Figura 45 a pagina 124, pubblica l'argomento utilizzando la stringa di

argomenti `"MQTT Examples"`. Nel frammento di codice Figura 46 a pagina 124, il client MQTT riceve la pubblicazione nel metodo callback, `messageArrived`.

Per ulteriori informazioni su come configurare IBM MQ per inviare pubblicazioni in risposta alle sottoscrizioni dai client MQTT , consultare Pubblicazione di un messaggio in risposta a una sottoscrizione client MQTT.

Un'applicazione IBM MQ invia un messaggio direttamente ad un client MQTT

Eseguire l'attività, “Invio di un messaggio a un client MQTT utilizzando IBM MQ Explorer” a pagina 129 per inviare un messaggio direttamente da IBM MQ a un client MQTT .

Un messaggio inviato in tal modo a un client MQTT viene definito messaggio non richiesto. I client MQTT v3 ricevono messaggi non richiesti come pubblicazioni con un nome argomento impostato. Il servizio di telemetria (MQXR) imposta il nome dell'argomento sul nome della coda remota.

Per ulteriori informazioni su come configurare IBM MQ per inviare messaggi direttamente ai client MQTT , consultare Invio diretto di un messaggio a un cliente.

Un client MQTT pubblica un messaggio

Un MQTT v3 client può pubblicare un messaggio ricevuto da un altro client MQTT v3 , ma non può inviare un messaggio non richiesto. Il frammento di codici Figura 47 a pagina 124 mostra come un client MQTT v3 , scritto in Java, pubblica un messaggio.

Il modello tipico per l'invio di un messaggio a un client MQTT v3 , è che ciascun client crei una sottoscrizione al proprio `ClientIdentifier`. Eseguire l'attività “Pubblicazione di un messaggio in un client MQTT v3 specifico” a pagina 130 per pubblicare un messaggio da un client MQTT a un altro MQTT utilizzando `ClientIdentifier` come stringa di argomento.

Frammenti di codice di esempio

Il frammento di codice in Figura 44 a pagina 123 mostra come un client MQTT scritto in Java crea una sottoscrizione. È inoltre necessario un metodo di callback, `messageArrived` per ricevere le pubblicazioni per la sottoscrizione.

```
String    clientId = String.format("%-23.23s",
                     System.getProperty("user.name") + "_" +
                     (UUID.randomUUID().toString())).trim()).replace('-', '_');
MqttClient  client = new MqttClient("localhost", clientId);
String topicString = "MQTT Examples";
int            QoS = 1;
client.subscribe(topicString, QoS);
```

Figura 44. sottoscrittore client MQTT v3

Il frammento di codice in Figura 45 a pagina 124 mostra come un'applicazione IBM MQ scritta in C invia una pubblicazione. Il frammento di codice viene estratto dall'attività Crea un publisher in un argomento variabile

```
/* Define and set variables to.defaults */
/* Omitted lines declaring variables    */
char * topicName   = ""
char * topicString = "MQTT Examples"
char * publication = "Hello world!";
do {
  MQCONN(qMgrName, &Hconn, &CompCode, &Reason);
  if (CompCode != MQCC_OK) break;
  td.ObjectType = MQOT_TOPIC;    /* Object is a topic             */
  td.Version = MQOD_VERSION_4;   /* Descriptor needs to be V4     */
  strncpy(td.ObjectName, topicName,  MQ_TOPIC_NAME_LENGTH);
  td.ObjectString.VSPtr = topicString;
  td.ObjectString.VSLength = (MQLONG)strlen(topicString);
  MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF_QUIESCING, &Hobj, &CompCode, &Reason);
  if (CompCode != MQCC_OK) break;
  pmo.Options = MQPMO_FAIL_IF_QUIESCING | MQPMO_RETAIN;
  MQPUT(Hconn, Hobj, &md, &pmo, (MQLONG)strlen(publication)+1, publication, &CompCode, &Reason);
  if (CompCode != MQCC_OK) break;
  MQCLOSE(Hconn, &Hobj, MQCO_NONE, &CompCode, &Reason);
  if (CompCode != MQCC_OK) break;
  MQDISC(&Hconn, &CompCode, &Reason);
} while (0);
```

Figura 45. IBM MQ publisher

Quando la pubblicazione arriva, il client MQTT richiama il metodo `messageArrived` della classe MQTT client delle applicazioni `MqttCallback` .

```
public class CallBack implements MqttCallback {
  public void messageArrived(MqttTopic topic, MqttMessage message) {
    try {
      System.out.println("Message arrived: \"" + message.toString()
          + "\" on topic \"" + topic.toString() + "\"");
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
// ... Other callback methods
}
```

Figura 46. `messageArrived` metodo

Figura 47 a pagina 124 mostra una pubblicazione MQTT v3 di un messaggio nella sottoscrizione creata in Figura 44 a pagina 123.

```
      String          address = "localhost";
      String         clientId = String.format("%-23.23s",
                                System.getProperty("user.name") + "_" +
                                (UUID.randomUUID().toString())).trim()).replace('-', '_');
      MqttClient       client = new MqttClient(address, clientId);
      String      topicString = "MQTT Examples";
      MqttTopic         topic = client.getTopic(Example.topicString);
      String     publication  = "Hello world";
      MqttMessage     message = new MqttMessage(publication.getBytes());
      MqttDeliveryToken token = topic.publish(message);
```

Figura 47. Programma di pubblicazione client MQTT v3

Linux Windows AIX Pubblicazione di un messaggio nel programma di utilità del client MQTT da IBM MQ Explorer

Attenersi alla procedura descritta in questa attività per pubblicare un messaggio utilizzando IBM MQ Explorere sottoscriverlo con il programma di utilità del client MQTT . Un'attività aggiuntiva mostra come

configurare un alias del gestore code piuttosto che impostare la coda di trasmissione predefinita su SYSTEM.MQTT.TRANSMIT.QUEUE.

Prima di iniziare

L'attività presuppone che l'utente abbia dimestichezza con IBM MQ e IBM MQ Explorere che IBM MQ e MQ Telemetry siano installati.

L'utente che crea le risorse del gestore code per questa attività deve disporre di autorizzazioni sufficienti. A scopo dimostrativo, si presuppone che l'ID utente IBM MQ Explorer sia membro del gruppo mqm .

Informazioni su questa attività

Nell'attività, si crea un topic in IBM MQ e si sottoscrive l'topic utilizzando il programma di utilità del client MQTT . Quando si pubblica l'argomento utilizzando IBM MQ Explorer, il client MQTT riceve la pubblicazione.

Procedura

Effettuare una delle seguenti attività:

- È stato installato MQ Telemetry, ma non è stato ancora avviato. Eseguire l'attività: "Avvia attività senza alcun servizio di telemetria (MQXR) ancora definito" a pagina 125.
- Si è già eseguita la telemetria IBM MQ , ma si desidera utilizzare un nuovo gestore code per eseguire la dimostrazione. Eseguire l'attività: "Avvia attività senza alcun servizio di telemetria (MQXR) ancora definito" a pagina 125.
- Si desidera eseguire l'attività utilizzando un gestore code esistente che non dispone di risorse di telemetria definite. Non si desidera eseguire la procedura guidata **Definisci configurazione di esempio** .
  a. Eseguire una delle seguenti attività per configurare la telemetria:
     – Configurazione di un gestore code per la telemetria su Linux e AIX
     – Configurazione di un gestore code per la telemetria su Windows
  b. Eseguire l'attività: "Avvia attività con un servizio di telemetria in esecuzione (MQXR)" a pagina 126
- Se si desidera eseguire l'attività utilizzando un gestore code esistente che dispone già di risorse di telemetria definite, eseguire l'attività: "Avvia attività con un servizio di telemetria in esecuzione (MQXR)" a pagina 126.

Operazioni successive

Eseguire "Invio di un messaggio a un client MQTT utilizzando IBM MQ Explorer" a pagina 129 per inviare un messaggio direttamente al programma di utilità client.

Avvia attività senza alcun servizio di telemetria (MQXR) ancora definito

Creare un gestore code ed eseguire **Definisci configurazione di esempio** per definire le risorse di telemetria di esempio per il gestore code. Pubblicare un messaggio utilizzando IBM MQ Explorere sottoscriverlo con il programma di utilità client MQTT .

Informazioni su questa attività

Quando si impostano le risorse di telemetria di esempio utilizzando **Definisci configurazione di esempio**, la procedura guidata imposta le autorizzazioni ID utente guest. Considerare attentamente se si desidera che l'ID utente guest sia autorizzato in questo modo. A guest su Windowse a nobody su Linuxviene concessa l'autorizzazione a pubblicare e sottoscrivere la root della struttura ad albero degli argomenti e a inserire i messaggi in SYSTEM.MQTT.TRANSMIT.QUEUE.

La procedura guidata imposta anche la coda di trasmissione predefinita su SYSTEM.MQTT.TRANSMIT.QUEUE, che potrebbe interferire con le applicazioni in esecuzione su un gestore code esistente. È possibile, ma laborioso, configurare la telemetria e non utilizzare la coda di

trasmissione predefinita; eseguire la seguente attività: “Utilizzo di un alias del gestore code” a pagina 127. In questa attività, si crea un gestore code per evitare la possibilità di interferire con qualsiasi coda di trasmissione predefinita esistente.

Procedura

1. Utilizzando IBM MQ Explorer, creare e avviare un nuovo gestore code.
   a) Fare clic con il tasto destro del mouse su `Queue Managers` cartella> **Nuovo** > **Gestore code ...**. Immettere un nome gestore code> **Fine**.

      Creare un nome gestore code; ad esempio, `MQTTQMGR`.
2. Creare e avviare il servizio di telemetria (MQXR) e creare un canale di telemetria di esempio.
   a) Aprire la cartella `Queue Managers\`*`QmgrName`*`\Telemetry` .
   b) Fare clic su **Definisci configurazione di esempio ...** > **Fine**

      Lasciare la casella di spunta **Avvia MQTT Programma di utilità client** selezionata.
3. Creare una sottoscrizione per `MQTT Example` utilizzando il programma di utilità client MQTT .
   a) Fare clic su **Connetti**.

      La **Cronologia client** registra un evento `Connected` .
   b) Immettere `MQTT Example` nel campo **Sottoscrizione \ Argomento** > **Sottoscrivi**.

   La **Cronologia client** registra un evento `Subscribed` .
4. Creare `MQTTExampleTopic` in IBM MQ.
   a) Fare clic con il tasto destro del mouse sulla cartella `Queue Managers\`*`QmgrName`*`\Topics` in **MQ Explorer**> **Nuovo** > **Argomento**.
   b) Immettere `MQTTExampleTopic` come **Nome** > **Avanti**.
   c) Immettere `MQTT Example` come **Stringa argomento** > **Fine**.
   d) Fare clic su **OK** per chiudere la finestra di conferma.
5. Pubblicare `Hello World!` nell'argomento `MQTT Example` utilizzando IBM MQ Explorer.
   a) Fare clic sulla cartella `Queue Managers\`*`QmgrName`*`\Topics` in IBM MQ Explorer.
   b) Fare clic con il tasto destro del mouse su `MQTTExampleTopic` > **Verifica pubblicazione ...**
   c) Immettere `Hello World!` nel campo **Dati del messaggio** > **Pubblica messaggio** > Passa alla finestra Programma di utilità del client MQTT .

   La **Cronologia client** registra un evento `Received` .

Avvia attività con un servizio di telemetria in esecuzione (MQXR)

Creare un canale di telemetria e un argomento. Autorizzare l'utente ad utilizzare l'argomento e la coda di trasmissione della telemetria. Pubblicare un messaggio utilizzando IBM MQ Explorere sottoscriverlo con il programma di utilità client MQTT .

Prima di iniziare

In questa versione dell'attività, un gestore code, *`QmgrName`*, è definito e in esecuzione. Un servizio di telemetria (MQXR) è definito e in esecuzione. Il servizio di telemetria (MQXR) potrebbe essere stato creato manualmente oppure eseguendo la procedura guidata **Definisci configurazione di esempio** .

Informazioni su questa attività

In questa attività si configurerà un gestore code esistente per inviare una pubblicazione al programma di utilità del client MQTT .

Il passo “1” a pagina 127 dell'attività imposta la coda di trasmissione predefinita su `SYSTEM.MQTT.TRANSMIT.QUEUE`, che potrebbe interferire con le applicazioni in esecuzione su un gestore code esistente. È possibile, ma laborioso, configurare la telemetria e non utilizzare la coda di

trasmissione predefinita; eseguire la seguente attività: “Utilizzo di un alias del gestore code” a pagina 127.

Procedura

1. Impostare SYSTEM.MQTT.TRANSMIT.QUEUE come coda di trasmissione predefinita.
   a) Fare clic con il tastino destro del mouse su Queue Managers*QmgrName* folder > **Proprietà**
   b) Fare clic su **Comunicazione** nel navigatore.
   c) Fare clic su **Seleziona ...** > Seleziona SYSTEM.MQTT.TRANSMIT.QUEUE > **OK** > **OK**.
2. Creare un canale di telemetria MQTTExampleChannel per connettere il programma di utilità client MQTT a IBM MQe avviare il programma di utilità client MQTT .
   a) Fare clic con il tasto destro del mouse sulla cartella Queue Managers*QmgrName* \Telemetry\Channels in **MQ Explorer**> **Nuovo** > **Canale di telemetria ...**.
   b) Immettere MQTTExampleChannel nel campo **Nome canale** > **Avanti** > **Avanti**.
   c) Modificare l' **ID utente fisso** nel pannello di autorizzazione del client nell'ID utente che pubblicherà e sottoscrivi MQTTExample > **Avanti**.
   d) Lasciare selezionato **Avvia programma di utilità client** > **Fine**.
3. Creare una sottoscrizione per MQTT Example utilizzando il programma di utilità client MQTT .
   a) Fare clic su **Connetti**.

      La **Cronologia client** registra un evento Connected .
   b) Immettere MQTT Example nel campo **Sottoscrizione \ Argomento** > **Sottoscrivi**.

   La **Cronologia client** registra un evento Subscribed .
4. Creare MQTTExampleTopic in IBM MQ.
   a) Fare clic con il tasto destro del mouse sulla cartella Queue Managers*QmgrName*\Topics in **MQ Explorer**> **Nuovo** > **Argomento**.
   b) Immettere MQTTExampleTopic come **Nome** > **Avanti**.
   c) Immettere MQTT Example come **Stringa argomento** > **Fine**.
   d) Fare clic su **OK** per chiudere la finestra di conferma.
5. Se si desidera che un utente, non appartenente al gruppo mqm , pubblichi e sottoscriva l'argomento MQTTExample , effettuare le seguenti operazioni:
   a) Autorizzare l'utente a pubblicare e sottoscrivere l'argomento MQTTExampleTopic:

      ```
      setmqaut -m qMgrName -t topic -n MQTTExampleTopic -p User ID -all +pub +sub
      ```

   b) Autorizzare l'utente a inserire un messaggio in SYSTEM.MQTT.TRANSMIT.QUEUE:

      ```
      setmqaut -m qMgrName -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p User ID -all +put
      ```

6. Pubblicare Hello World! nell'argomento MQTT Example utilizzando IBM MQ Explorer.
   a) Fare clic sulla cartella Queue Managers*QmgrName*\Topics in IBM MQ Explorer.
   b) Fare clic con il tasto destro del mouse su MQTTExampleTopic > **Verifica pubblicazione ...**
   c) Immettere Hello World! nel campo **Dati del messaggio** > **Pubblica messaggio** > Passa alla finestra Programma di utilità del client MQTT .

   La **Cronologia client** registra un evento Received .

Utilizzo di un alias del gestore code

Pubblicare un messaggio nel programma di utilità del client MQTT utilizzando IBM MQ Explorer senza impostare la coda di trasmissione predefinita su SYSTEM.MQTT.TRANSMIT.QUEUE.

L'attività è una continuazione dell'attività precedente e utilizza un alias del gestore code per evitare di impostare la coda di trasmissione predefinita su SYSTEM.MQTT.TRANSMIT.QUEUE.

Prima di iniziare

Completare l'attività, "Avvia attività senza alcun servizio di telemetria (MQXR) ancora definito" a pagina 125 o l'attività, "Avvia attività con un servizio di telemetria in esecuzione (MQXR)" a pagina 126.

Informazioni su questa attività

Quando un MQTT client crea una sottoscrizione, IBM MQ invia la risposta utilizzando `ClientIdentifier`, come nome del gestore code remoto. In questa attività, utilizza `ClientIdentifier`, `MyClient`.

Se non esiste una coda di trasmissione o un alias del gestore code denominato `MyClient`, la risposta viene inserita nella coda di trasmissione predefinita. Impostando la coda di trasmissione predefinita su SYSTEM.MQTT.TRANSMIT.QUEUE, il client MQTT ottiene la risposta.

È possibile evitare di impostare la coda di trasmissione predefinita su SYSTEM.MQTT.TRANSMIT.QUEUE utilizzando gli alias del gestore code. È necessario impostare un alias del gestore code per ogni `ClientIdentifier`. Di solito, ci sono troppi client per rendere pratico l'utilizzo degli alias dei gestori code. Spesso `ClientIdentifier` è imprevedibile, rendendo impossibile configurare la telemetria in questo modo.

Tuttavia, in alcune circostanze potrebbe essere necessario configurare la coda di trasmissione predefinita su un valore diverso da SYSTEM.MQTT.TRANSMIT.QUEUE. I passaggi in Procedura configurano un alias del gestore code invece di impostare la coda di trasmissione predefinita su SYSTEM.MQTT.TRANSMIT.QUEUE.

Procedura

1. Rimuovere SYSTEM.MQTT.TRANSMIT.QUEUE come coda di trasmissione predefinita.
   a) Fare clic con il tastino destro del mouse su `Queue Managers\`*`QmgrName`* `folder` > **Proprietà**
   b) Fare clic su **Comunicazione** nel navigatore.
   c) Rimuovere SYSTEM.MQTT.TRANSMIT.QUEUE dal campo **Coda di trasmissione predefinita** > **OK**.
2. Verificare che non sia più possibile creare una sottoscrizione con l'utilità client MQTT :
   a) Fare clic su **Connetti**.

      La **Cronologia client** registra un evento `Connected` .
   b) Immettere `MQTT Example` nel campo **Sottoscrizione \ Argomento** > **Sottoscrivi**.

   La **Cronologia client** registra un `Subscribe failed` e un evento `Connection lost` .
3. Creare un alias del gestore code per `ClientIdentifier`, `MyClient`.
   a) Fare clic con il pulsante destro del mouse sulla cartella `Queue Managers\`*`QmgrName`*`\Queues` > **Nuovo** > **Definizione coda remota**.
   b) Denominare la definizione, `MyClient` > **Avanti**.
   c) Immettere `MyClient` nel campo **Gestore code remoto** .
   d) Immettere SYSTEM.MQTT.TRANSMIT.QUEUE nel campo **Coda di trasmissione** > **Fine**.
4. Connettere nuovamente il programma di utilità del client MQTT .
   a) Verificare che **Identificativo client** sia impostato su `MyClient`.
   b) **Connetti**

   La **Cronologia client** registra un evento `Connected` .
5. Creare una sottoscrizione per `MQTT Example` utilizzando il programma di utilità client MQTT .
   a) Fare clic su **Connetti**.

La **Cronologia client** registra un evento `Connected` .

b) Immettere `MQTT Example` nel campo **Sottoscrizione \ Argomento** > **Sottoscrivi**.

La **Cronologia client** registra un evento `Subscribed` .

6. Pubblicare `Hello World!` nell'argomento `MQTT Example` utilizzando IBM MQ Explorer.
   a) Fare clic sulla cartella `Queue Managers\`*`QmgrName`*`\Topics` in IBM MQ Explorer.
   b) Fare clic con il tasto destro del mouse su `MQTTExampleTopic` > **Verifica pubblicazione ...**
   c) Immettere `Hello World!` nel campo **Dati del messaggio** > **Pubblica messaggio** > Passa alla finestra Programma di utilità del client MQTT .

   La **Cronologia client** registra un evento `Received` .

Linux Windows AIX Invio di un messaggio a un client MQTT utilizzando IBM MQ Explorer

Inviare un messaggio al programma di utilità del client MQTT inserendo un messaggio in una coda IBM MQ utilizzando IBM MQ Explorer. L'attività mostra come configurare una definizione di coda remota per l'invio di messaggi direttamente a un client MQTT .

Prima di iniziare

Eseguire l'attività, "Pubblicazione di un messaggio nel programma di utilità del client MQTT da IBM MQ Explorer" a pagina 124. Lasciare connesso il programma di utilità del client MQTT .

Informazioni su questa attività

L'attività dimostra l'invio di un messaggio a un client MQTT utilizzando la coda piuttosto che la pubblicazione in un argomento. Non si crea una sottoscrizione nel client. Il passo "2" a pagina 129 dell'attività dimostra che la precedente sottoscrizione è stata eliminata.

Procedura

1. Eliminare eventuali sottoscrizioni esistenti disconnettendo e ricollegando il programma di utilità del client MQTT .

   La sottoscrizione viene eliminata perché, a meno che non si modifichino i valori predefiniti, il programma di utilità del client MQTT si connette con una sessione pulita; consultare Sessione pulita.

   Per rendere più semplice l'esecuzione dell'attività, immettere il proprio `ClientIdentifier`, piuttosto che utilizzare il `ClientIdentifier` generato dal programma di utilità client MQTT .

   a) Fare clic su **Disconnetti** per disconnettere il programma di utilità del client MQTT dal canale di telemetria.

      La **Cronologia client** registra un evento `Disconnected`

   b) Modificare l' **Identificativo client** in `MyClient`.

   c) Fare clic su **Connetti**.

   La **Cronologia client** registra un evento `Connected`

2. Controllare che l'utilità client MQTT non riceva più la pubblicazione per `MQTTExampleTopic`.
   a) Fare clic sulla cartella `Queue Managers\`*`QmgrName`*`\Topics` in IBM MQ Explorer.
   b) Fare clic con il tasto destro del mouse su `MQTTExampleTopic` > **Verifica pubblicazione ...**
   c) Immettere `Hello World!` nel campo **Dati del messaggio** > **Pubblica messaggio** > Passa alla finestra Programma di utilità del client MQTT .

   Nessun evento viene registrato nella **Cronologia client**.

3. Creare una definizione di coda remota per il client.

Impostare `ClientIdentifier`, `MyClient`, come nome del gestore code remoto nella definizione della coda remota. Utilizzare qualsiasi nome che si desidera come nome della coda remota. Il nome della coda remota viene inoltrato a un client MQTT come nome argomento.

a) Fare clic con il pulsante destro del mouse sulla cartella `Queue Managers\`*`QmgrName`*`\Queues` > **Nuovo** > **Definizione coda remota**.

b) Denominare la definizione, `MyClientRemoteQueue` > **Avanti**.

c) Immettere `MQTTExampleQueue` nel campo **Coda remota** .

d) Immettere `MyClient` nel campo **Gestore code remoto** .

e) Immettere `SYSTEM.MQTT.TRANSMIT.QUEUE` nel campo **Coda di trasmissione** > **Fine**.

4. Inserire un messaggio di test in `MyClientRemoteQueue`.

a) Fare clic con il tasto destro del mouse su **MyClientRemoteQueue** > **Inserisci messaggio di prova ...**

b) Immettere `Hello queue!` nel campo Dati messaggio> **Inserisci messaggio** > **Chiudi**

La **Cronologia client** registra un evento `Received` .

5. Rimuovere `SYSTEM.MQTT.TRANSMIT.QUEUE` come coda di trasmissione predefinita.

a) Fare clic con il tastino destro del mouse su `Queue Managers\`*`QmgrName`* `folder` > **Proprietà**

b) Fare clic su **Comunicazione** nel navigatore.

c) Rimuovere `SYSTEM.MQTT.TRANSMIT.QUEUE` dal campo **Coda di trasmissione predefinita** > **OK**.

6. Ripetere il passo "4" a pagina 130.

`MyClientRemoteQueue` è una definizione di coda remota che denomina esplicitamente la coda di trasmissione. non è necessario definire una coda di trasmissione predefinita per inviare un messaggio a `MyClient`.

Operazioni successive

Con la coda di trasmissione predefinita non più impostata su `SYSTEM.MQTT.TRANSMIT.QUEUE`, il programma di utilità client MQTT non è in grado di creare una nuova sottoscrizione a meno che non sia stato definito un alias del gestore code per `ClientIdentifier`, `MyClient`. Ripristinare la coda di trasmissione predefinita su `SYSTEM.MQTT.TRANSMIT.QUEUE`.

Linux Windows AIX Pubblicazione di un messaggio in un client MQTT v3 specifico

Pubblicare un messaggio da un client MQTT v3 ad un altro, utilizzando `ClientIdentifier` come nome argomento e IBM MQ come broker di pubblicazione / sottoscrizione.

Prima di iniziare

Eseguire l'attività, "Pubblicazione di un messaggio nel programma di utilità del client MQTT da IBM MQ Explorer" a pagina 124. Lasciare connesso il programma di utilità del client MQTT .

Informazioni su questa attività

L'attività mostra due cose:

1. Sottoscrizione di un argomento in un client MQTT e ricezione di una pubblicazione da un altro client MQTT .
2. Impostazione delle sottoscrizioni "point - to - point" utilizzando `ClientIdentifier` come stringa argomento.

Procedura

1. Eliminare eventuali sottoscrizioni esistenti disconnettendo e ricollegando il programma di utilità del client MQTT .

   La sottoscrizione viene eliminata perché, a meno che non si modifichino i valori predefiniti, il programma di utilità del client MQTT si connette con una sessione pulita; consultare Sessione pulita.

   Per rendere più semplice l'esecuzione dell'attività, immettere il proprio `ClientIdentifier`, piuttosto che utilizzare il `ClientIdentifier` generato dal programma di utilità client MQTT .

   a) Fare clic su **Disconnetti** per disconnettere il programma di utilità del client MQTT dal canale di telemetria.

      La **Cronologia client** registra un evento `Disconnected`

   b) Modificare l' **Identificativo client** in `MyClient`.

   c) Fare clic su **Connetti**.

   La **Cronologia client** registra un evento `Connected`
2. Creare una sottoscrizione all'argomento, `MyClient`

   `MyClient` è `ClientIdentifier` di questo client.

   a) Immettere `MyClient` nel campo **Sottoscrizione \ Argomento** > **Sottoscrivi**.

   La **Cronologia client** registra un evento `Subscribed` .
3. Avviare un altro programma di utilità client MQTT .

   a) Aprire la cartella `Queue Managers\`*`QmgrName`*`\Telemetry\channels` .

   b) Fare clic con il tasto destro del mouse sul canale **PlainText** > **Esegui programma di utilità client MQTT ...**

   c) Fare clic su **Connetti**.

   La **Cronologia client** registra un evento `Connected`
4. Pubblicare `Hello MyClient!` nell'argomento `MyClient`.

   a) Copiare l'argomento della sottoscrizione, `MyClient`, dal programma di utilità client MQTT in esecuzione con `ClientIdentifier`, `MyClient`.

   b) Incollare `MyClient` nel campo **Pubblicazione / Argomento** di ciascuna istanza del programma di utilità client di MQTT .

   c) Immettere `Hello MyClient!` nel campo **Pubblicazione \ messaggio** .

   d) Fare clic su **Pubblica** in entrambe le istanze.

Risultati

La **Cronologia client** nel programma di utilità del client MQTT con `ClientIdentifier`, `MyClient`, registra due eventi **Ricevuti** e un evento **Pubblicato** . L'altra istanza del programma di utilità client MQTT registra un evento **Pubblicato** .

Se viene visualizzato un solo evento **Ricevuto** , verificare le seguenti possibili cause:

1. La coda di trasmissione predefinita per il gestore code è impostata su `SYSTEM.MQTT.TRANSMIT.QUEUE` ?
2. Sono stati creati alias del gestore code o definizioni di code remote che fanno riferimento a `MyClient` durante gli altri esercizi? Nel caso in cui si verifichi un problema di configurazione, eliminare le risorse che fanno riferimento a `MyClient`, come ad esempio gli alias di un gestore code o le code di trasmissione. Disconnettere le utilità client, arrestare e riavviare il servizio di telemetria (MQXR).

Linux Windows AIX Invio di un messaggio a un'applicazione IBM MQ da un client MQTT

Un'applicazione IBM MQ può ricevere un messaggio da un client MQTT v3 sottoscrivendo un argomento. Il client MQTT si connette a IBM MQ utilizzando un canale di telemetria e invia un messaggio all'applicazione IBM MQ pubblicando nello stesso argomento.

Eseguire l'attività “Pubblicazione di un messaggio in IBM MQ da un client MQTT” a pagina 132per informazioni su come inviare una pubblicazione da un client MQTT a una sottoscrizione definita in IBM MQ.

Se l'argomento è in cluster o distribuito utilizzando una gerarchia di pubblicazione / sottoscrizione, la sottoscrizione può trovarsi su un gestore code differente rispetto al gestore code a cui è connesso il client MQTT .

Linux Windows AIX Pubblicazione di un messaggio in IBM MQ da un client MQTT

Creare una sottoscrizione a un argomento utilizzando IBM MQ Explorer e pubblicarla nell'argomento utilizzando l'utilità del client MQTT .

Prima di iniziare

Eseguire l'attività, “Pubblicazione di un messaggio nel programma di utilità del client MQTT da IBM MQ Explorer” a pagina 124. Lasciare connesso il programma di utilità del client MQTT .

Informazioni su questa attività

L'attività illustra la pubblicazione di un messaggio con un cliente MQTT e la ricezione della pubblicazione utilizzando una sottoscrizione durevole non gestita creata utilizzando IBM MQ Explorer.

Procedura

1. Creare una sottoscrizione durevole alla stringa di argomenti `MQTT Example`.

   Effettuare le seguenti operazioni per creare la coda e la sottoscrizione utilizzando IBM MQ Explorer.

   a) Fare clic con il pulsante destro del mouse su `Queue Managers\`*`QmgrName`*`\Queues` in IBM MQ Explorer> **Nuovo** > **Coda locale ...**.

   b) Immettere `MQTTExampleQueue` come nome coda> **Fine**.

   c) Fare clic con il pulsante destro del mouse sulla cartella `Queue Managers\`*`QmgrName`*`\Subscriptions` in IBM MQ Explorer> **Nuovo** > **Sottoscrizione**.

   d) Immettere `MQTTExampleSubscription` come nome coda> **Avanti**.

   e) Fare clic su **Seleziona ...** > `MQTTExampleTopic` > **OK**.

   L'argomento è già stato creato, `MQTTExampleTopic` nel passo “4” a pagina 126 di “Pubblicazione di un messaggio nel programma di utilità del client MQTT da IBM MQ Explorer” a pagina 124.

   f) Immettere `MQTTExampleQueue` come nome destinazione> **Fine**.

2. Come passo facoltativo, impostare la coda per l'utilizzo da parte di un utente differente, senza l'autorità `mqm` .

   Se si sta configurando la configurazione per gli utenti con meno autorizzazioni di `mqm`, è necessario fornire l'autorizzazione `put` e `get` a `MQTTExampleQueue`. L'accesso all'argomento e alla coda di trasmissione è stato configurato in “Pubblicazione di un messaggio nel programma di utilità del client MQTT da IBM MQ Explorer” a pagina 124.

   a) Autorizzare un utente a inserire e richiamare la coda `MQTTExampleQueue`:

```
setmqaut -m qMgrName -t queue -n MQTTExampleQueue -p User ID -all +put +get
```

3. Pubblicare `Hello IBM MQ!` nell'argomento `MQTT Example` utilizzando l'utilità client MQTT .

   Se il programma di utilità del client MQTT non è stato ancora connesso, fare clic con il tasto destro del mouse sul canale **PlainText** > **Esegui programma di utilità del client MQTT ...** > **Connetti**.

   a) Immettere `MQTT Example` nel campo **Pubblicazione / Argomento** .

   b) Immettere `Hello IBM MQ!` nel campo **Pubblicazione \ Messaggio** > **Pubblica**.
4. Aprire la cartella `Queue Managers\`*`QmgrName`*`\Queues` e trovare `MQTTExampleQueue`.

   Il campo **Profondità coda corrente** è 1
5. Fare clic con il tasto destro del mouse su `MQTTExampleQueue` > **Sfoglia messaggi ...** ed esaminare la pubblicazione.

Linux Windows AIX MQTT applicazioni di pubblicazione/ sottoscrizione

Utilizzare la pubblicazione / sottoscrizione basata sull'argomento per scrivere applicazioni MQTT .

Quando il client MQTT è connesso, le pubblicazioni passano in entrambe le direzioni tra il client e il server. Le pubblicazioni vengono inviate dal client quando le informazioni vengono pubblicate sul client. Le pubblicazioni vengono ricevute sul client quando un messaggio viene pubblicato su un argomento che corrisponde a una sottoscrizione creata dal client.

Il broker di pubblicazione / sottoscrizione IBM MQ gestisce gli argomenti e le sottoscrizioni creati dai client MQTT . Gli argomenti creati dai clienti MQTT condividono lo stesso spazio argomenti creato dalle applicazioni IBM MQ .

Le pubblicazioni che corrispondono alla stringa argomento in una sottoscrizione client MQTT vengono collocate in `SYSTEM.MQTT.TRANSMIT.QUEUE` con il nome del gestore code remoto impostato su `ClientIdentifier` del client. Il servizio di telemetria (MQXR) inoltra le pubblicazioni al client che ha creato la sottoscrizione. Utilizza `ClientIdentifier`, che è stato impostato come nome del gestore code remoto per identificare il client.

Generalmente, `SYSTEM.MQTT.TRANSMIT.QUEUE` deve essere definito come coda di trasmissione predefinita. È possibile, ma oneroso, configurare MQTT in modo che non utilizzi la coda di trasmissione predefinita; consultare Configurazione dell'accodamento distribuito per inviare messaggi ai client MQTT.

Un client MQTT può creare una sessione persistente; consultare “MQTT sessioni stateless e stateful” a pagina 136. Le sottoscrizioni create in una sessione persistente sono durevoli. Le pubblicazioni che arrivano per un client con una sessione persistente vengono memorizzate in `SYSTEM.MQTT.TRANSMIT.QUEUE`e inoltrate al client quando si riconnette.

Un client MQTT può anche pubblicare e sottoscrivere pubblicazioni conservate; consultare Pubblicazioni conservate e client MQTT. Un sottoscrittore a un argomento di pubblicazione conservato riceve l'ultima pubblicazione per l'argomento. Il sottoscrittore riceve la pubblicazione conservata quando crea una sottoscrizione o quando si riconnette alla sessione precedente.

Linux Windows AIX Applicazioni di telemetria

Scrivere applicazioni di telemetria utilizzando flussi di messaggi IBM MQ o IBM Integration Bus .

Utilizzare JMS, MQI o altre interfacce di programmazione IBM MQ per programmare le applicazioni di telemetria in IBM MQ.

Il servizio di telemetria (MQXR) converte tra i messaggi MQTT v3 e i messaggi IBM MQ . Crea sottoscrizioni e pubblicazioni per conto dei client MQTT e inoltra le pubblicazioni ai client MQTT . Una pubblicazione è il payload di un messaggio MQTT v3 . Il payload comprende le intestazioni del messaggio e un array di byte in formato `jms-bytes` . Il server di telemetria associa le intestazioni tra un messaggio MQTT v3 e uno IBM MQ ; consultare “Integrazione di MQ Telemetry con i gestori code” a pagina 134.

Utilizzare i nodi Publication, MQInput e JMSInput per inviare e ricevere pubblicazioni tra client IBM Integration Bus e MQTT .

Utilizzando i flussi di messaggi è possibile integrare la telemetria con i siti Web utilizzando HTTPe con altre applicazioni utilizzando IBM MQ e WebSphere Adapters.

Linux Windows AIX Integrazione di MQ Telemetry con i gestori code

Il client MQTT è integrato con IBM MQ come applicazione di pubblicazione / sottoscrizione. Può pubblicare o sottoscrivere argomenti in IBM MQ, creare nuovi argomenti o utilizzare argomenti esistenti. Riceve pubblicazioni da IBM MQ come risultato di client MQTT , incluso se stesso, o altre applicazioni IBM MQ che pubblicano gli argomenti delle relative sottoscrizioni. Le regole vengono applicate per decidere gli attributi di una pubblicazione.

Molti degli attributi associati ad argomenti, pubblicazioni, sottoscrizioni e messaggi forniti da IBM MQnon sono supportati. "Client MQTT per IBM MQ broker di pubblicazione / sottoscrizione" a pagina 134 e "IBM MQ su un client MQTT" a pagina 135 descrivono come vengono impostati gli attributi delle pubblicazioni. Le impostazioni dipendono dal fatto che la pubblicazione venga eseguita o meno dal broker di pubblicazione / sottoscrizione IBM MQ .

In IBM MQ , gli argomenti di pubblicazione / sottoscrizione sono associati agli oggetti argomento di gestione. Gli argomenti creati dai client MQTT non sono diversi. Quando un client MQTT crea una stringa di argomento per una pubblicazione, il broker di pubblicazione / sottoscrizione IBM MQ la associa a un oggetto argomento di amministrazione. Il broker associa la stringa argomento nella pubblicazione all'oggetto argomento di gestione principale più vicino. L'associazione è uguale a quella delle applicazioni IBM MQ . Se non è presente alcun argomento creato dall'utente, l'argomento di pubblicazione viene associato a `SYSTEM.BASE.TOPIC`. Gli attributi applicati alla pubblicazione derivano dall'oggetto argomento.

Quando un'applicazione IBM MQ o un amministratore crea una sottoscrizione, la sottoscrizione viene denominata. Elencare le sottoscrizioni utilizzando IBM MQ Explorero utilizzando i comandi **runmqsc** o PCF. Tutte le sottoscrizioni dei client MQTT vengono denominate. Viene fornito un nome del formato: *`ClientIdentifier:Topic name`*

Client MQTT per IBM MQ broker di pubblicazione / sottoscrizione

Un client MQTT ha inviato una pubblicazione a IBM MQ. Il servizio di telemetria (MQXR) converte la pubblicazione in messaggio IBM MQ . Il messaggio IBM MQ contiene tre parti:

1. MQMD
2. RFH2
3. Messaggio

Le proprietà MQMD sono impostate sui valori predefiniti, tranne dove indicato in Tabella 9 a pagina 134.

Tabella 9. Impostazioni per proprietà MQMD

campo MQMD	Tipo	Valore
Format	MQCHAR8	MQFMT_RF_HEADER_2

UserIdentifier	MQCHAR12	Impostare su uno tra:
		`MqttClient.ClientIdentifier`
		`MqttConnectOptions.UserName`
		Un ID utente impostato dall'amministratore IBM MQ per il canale di telemetria.

Priority	MQLONG	MQPRI_PRIORITY_AS_Q_DEF (Impostazione predefinita per IBM MQ, che è diverso da JMS che ha un valore predefinito di 4.)

Tabella 9. Impostazioni per proprietà MQMD (Continua)		
campo MQMD	**Tipo**	**Valore**
Persistence	MQLONG	QoS=0→MQPER_NOT_PERSISTENT QoS=1→MQPER_PERSISTENT QoS=2→MQPER_PERSISTENT

L'intestazione RFH2 non contiene una cartella <msd> per definire il tipo di messaggio JMS . Il servizio di telemetria (MQXR) crea il messaggio IBM MQ come messaggio predefinito JMS . Il tipo di messaggio JMS predefinito è un jms-bytes . Un'applicazione può accedere ad ulteriori informazioni di intestazione come proprietà del messaggio; consultare Proprietà del messaggio.

I valori RFH2 sono impostati come mostrato in Tabella 10 a pagina 135. La proprietà Formato è impostata nell'intestazione fissa RFH2 e gli altri valori sono impostati nelle cartelle di RFH2 .

Tabella 10. Impostazioni per proprietà RFH2		
proprietà RFH2	**Tipo / Cartella**	**Intestazione**
Format	MQCHAR8	MQFMT_NONE
ClientIdentifier	mqtt/clientId	Copiare MqttClient.ClientIdentifier con una lunghezza di 1 ... 23 byte.
QoS	mqtt/qos	Copiare QoS dal messaggio MQTT in entrata.
ID messaggio	mqtt/msgid	Copiare ID messaggio dal messaggio MQTT in entrata, se QoS è 1 o 2.
MQIsRetained	mqps/Ret	Impostare se la pubblicazione MQTT originale è stata inviata con la proprietà RETAIN impostata e il messaggio viene ricevuto come pubblicazione conservata.
MQTopicString	mqps/Top	L'argomento in cui è pubblicato il messaggio MQTT .

Il payload in una pubblicazione MQTT viene associato al contenuto di un messaggio IBM MQ :

Tabella 11. Modalità di associazione del payload in una pubblicazione MQTT al contenuto del messaggio IBM MQ		
Contenuto messaggio	**Tipo**	**Contenuto del messaggio IBM MQ**
Memorizza nel buffer	MQBYTE *n*	Copia dei byte dal messaggio MQTT in entrata. La lunghezza può essere zero.

IBM MQ su un client MQTT

Un client ha sottoscritto un argomento di pubblicazione. Un'applicazione IBM MQ è stata pubblicata nell'argomento, determinando l'invio di una pubblicazione al sottoscrittore MQTT da parte del broker di pubblicazione / sottoscrizione IBM MQ . In alternativa, un'applicazione di IBM MQ ha inviato un messaggio non richiesto direttamente a un client MQTT . Tabella 12 a pagina 136 descrive il modo in cui le intestazioni del messaggio fisso sono impostate nel messaggio inviato al client MQTT . Tutti gli altri dati nell'intestazione del messaggio IBM MQ , o qualsiasi altra intestazione, vengono eliminati. I dati del messaggio nel messaggio IBM MQ vengono inviati come payload del messaggio nel messaggio MQTT , senza alcuna modifica. Il messaggio MQTT viene inviato al client MQTT dal servizio di telemetria (MQXR).

Tabella 12. Modalità di impostazione delle intestazioni di messaggi fissi in un messaggio IBM MQ inviato al client MQTT

MQTT campo	Tipo	Valore
DUP	booleano	Impostare se QoS = 1 o 2e il messaggio è stato inviato a questo client in una trasmissione precedente e il messaggio non è stato riconosciuto dopo un periodo di tempo.
QoS	int	Il modo in cui viene impostato il valore di QoS in una pubblicazione in uscita dal broker di pubblicazione / sottoscrizione in IBM MQ dipende dalla pubblicazione in entrata. Dipende se la pubblicazione in entrata è stata inviata da un client MQTT o da un'applicazione IBM MQ . **MQTT** Valore inferiore di QoS nella pubblicazione in entrata e in QoS richiesto dal sottoscrittore. **IBM MQ** Valore inferiore del QoS derivato dalla pubblicazione in entrata: MQPER_NOT_PERSISTENT→QoS=0 MQPER_PERSISTENT→QoS=2 e il QoS richiesto dal sottoscrittore. Se il messaggio viene inviato al client senza una sottoscrizione, per impostazione predefinita QoS è impostato su 2. Un client può modificare questo valore sottoscrivendo DEFAULT.QoS con un QoSdifferente.
RETAIN	booleano	Impostare se la pubblicazione in entrata ha la proprietà conservata impostata.

Tabella 13 a pagina 136 descrive il modo in cui sono impostate le intestazioni del messaggio variabile nel messaggio MQTT inviato al client MQTT .

Tabella 13. Modalità di impostazione delle proprietà dell'intestazione della variabile MQTT in un messaggio MQTT inviato al client MQTT

MQTT campo	Tipo	Valore
Topic name	Stringa	La stringa di argomenti con cui è stato pubblicato il messaggio.
Message ID	Stringa	Gli ultimi 2 byte di MQMD.MsgId proprietà della pubblicazione quando viene inserita in SYSTEM.MQTT.TRANSMIT.QUEUE.
Payload	byte[]	Copia diretta dei byte dalla pubblicazione in entrata al broker di pubblicazione / sottoscrizione. La lunghezza può essere zero.

Linux Windows AIX

MQTT sessioni stateless e stateful

I client MQTT possono creare una sessione con stato con il gestore code. Quando un client MQTT con stato si disconnette, il gestore code conserva le sottoscrizioni create dal client e i messaggi in corso. Quando il client si riconnette, risolve il messaggio incompleto. Invia tutti i messaggi che sono in coda per la consegna e riceve tutti i messaggi pubblicati per le sue sottoscrizioni mentre è stato disconnesso.

Quando un client MQTT si connette a un canale di telemetria, avvia una nuova sessione o riprende una vecchia sessione. Una nuova sessione non ha messaggi in sospeso che non sono stati riconosciuti,

nessuna sottoscrizione e nessuna pubblicazione in attesa di consegna. Quando un client si connette, specifica se iniziare con una sessione di ripulitura o riprendere una sessione esistente; consultare Sessione di ripulitura.

Se il client riprende una sessione esistente, continua come se la connessione non fosse stata interrotta. Le pubblicazioni in attesa di consegna vengono inviate al client e tutti i trasferimenti di messaggi di cui non è stato eseguito il commit vengono completati. Quando un client in una sessione persistente si disconnette dal servizio di telemetria (MQXR), tutte le sottoscrizioni create dal client rimangono. Le pubblicazioni per le sottoscrizioni vengono inviate al client quando si riconnette. Se si riconnette senza riprendere la vecchia sessione, le pubblicazioni vengono eliminate dal servizio di telemetria (MQXR).

Le informazioni sullo stato della sessione vengono salvate dal gestore code nella coda `SYSTEM.MQTT.PERSISTENT.STATE` .

L'amministratore IBM MQ può disconnettere ed eliminare una sessione.

Linux Windows AIX Quando un client MQTT non è connesso

Quando un client non è connesso, il gestore code può continuare a ricevere pubblicazioni per suo conto. Vengono inoltrati al client quando si riconnette. Un client può creare un "Ultimo testamento", che il gestore code pubblica per conto del client, se il client si disconnette inaspettatamente.

Se si desidera ricevere una notifica quando il client si disconnette inaspettatamente, è possibile registrare una pubblicazione Ultimo testamento e testamento; consultare Pubblicazione Ultimo testamento e testamento. Viene inviato dal servizio di telemetria (MQXR), se rileva che la connessione al client si è interrotta senza che il client lo richieda.

Un client può pubblicare una pubblicazione conservata in qualsiasi momento; consultare Pubblicazioni conservate e client MQTT. Una nuova sottoscrizione a un argomento può richiedere l'invio di qualsiasi pubblicazione conservata associata all'argomento. Se si crea l'ultimo testamento come pubblicazione conservata, è possibile utilizzarlo per monitorare lo stato di un client.

Ad esempio, il client pubblica una pubblicazione conservata, quando si collega, pubblicizzando la relativa disponibilità. Allo stesso tempo, crea una pubblicazione conservata di ultima volontà e testamento che annuncia la sua indisponibilità. Inoltre, prima di effettuare una disconnessione pianificata, pubblica la sua indisponibilità come pubblicazione conservata. Per scoprire se il client è disponibile, è necessario sottoscrivere l'argomento della pubblicazione conservata. Riceverai sempre una delle tre pubblicazioni.

Se il client deve ricevere i messaggi pubblicati quando è disconnesso, riconnettere il client alla sessione precedente; consultare “MQTT sessioni stateless e stateful” a pagina 136. Le relative sottoscrizioni sono attive fino a quando non vengono eliminate o fino a quando il client non crea una sessione pulita.

Linux Windows AIX Accoppiamento debole tra client di MQTT e applicazioni IBM MQ

Il flusso delle pubblicazioni tra client MQTT e applicazioni IBM MQ è accoppiato in modo flessibile. Le pubblicazioni potrebbero provenire da un client MQTT o da un'applicazione IBM MQ e non in un ordine impostato. Gli editori e i sottoscrittori sono debolmente accoppiati. Interagiscono tra loro indirettamente tramite pubblicazioni e sottoscrizioni. È anche possibile inviare messaggi direttamente a un client MQTT da un'applicazione IBM MQ .

I client MQTT e le applicazioni IBM MQ sono accoppiati in due modi:

1. Gli autori (publisher) e i sottoscrittori (subscriber) sono associati liberamente da una pubblicazione e da una sottoscrizione a un argomento. Gli editori e i sottoscrittori normalmente non sono a conoscenza dell'indirizzo o dell'identità dell'altra fonte di una pubblicazione o di un abbonamento.
2. I client MQTT pubblicano, sottoscrivono, ricevono pubblicazioni ed elaborano riconoscimenti di consegna su thread separati.

Un'applicazione client MQTT non attende la consegna di una pubblicazione. L'applicazione passa un messaggio al client MQTT , quindi l'applicazione continua sul proprio thread. Un token di consegna viene utilizzato per sincronizzare l'applicazione con la consegna di una pubblicazione; vedi Token di consegna.

Dopo aver passato un messaggio al client MQTT , l'applicazione ha la possibilità di attendere il token di consegna. Invece di attendere, il client può fornire un metodo di callback che viene richiamato quando la pubblicazione viene consegnata a IBM MQ. Può anche ignorare il token di consegna.

A seconda della qualità del servizio associata al messaggio, il token di consegna viene restituito immediatamente al metodo di callback o, eventualmente, dopo un periodo di tempo considerevole. Il token di consegna potrebbe essere restituito anche dopo che il client si è scollegato e riconnesso. Se la QoS (quality of service) è *fire and forget*, il token di distribuzione viene restituito immediatamente. Negli altri due casi, il token di consegna viene restituito solo quando il client riceve il riconoscimento che la pubblicazione è stata inviata ai sottoscrittori.

Le pubblicazioni inviate a un client MQTT come risultato di una sottoscrizione client vengono consegnate al metodo di callback `messageArrived` . `messageArrived` viene eseguito su un thread differente rispetto all'applicazione principale.

Invio di messaggi direttamente a un client MQTT

È possibile inviare un messaggio a uno specifico client MQTT in due modi.

1. Un'applicazione IBM MQ può inviare un messaggio direttamente a un client MQTT senza una sottoscrizione; consultare Invio di un messaggio direttamente a un client.
2. Un approccio alternativo consiste nell'utilizzare la convenzione di denominazione `ClientIdentifier` . Fare in modo che tutti i sottoscrittori MQTT creino sottoscrizioni utilizzando il proprio `ClientIdentifier` univoco come argomento. Pubblicare in *ClientIdentifier*. La pubblicazione viene inviata al client che ha sottoscritto l'argomento *ClientIdentifier*. Utilizzando questa tecnica, è possibile inviare una pubblicazione a un particolare sottoscrittore MQTT .

Linux Windows AIX Sicurezza di MQ Telemetry

Proteggere i dispositivi di telemetria può essere importante, in quanto è probabile che siano portatili e utilizzati in luoghi che non possono essere controllati con attenzione. Puoi utilizzare la VPN per proteggere la connessione dal dispositivo MQTT al servizio di telemetria (MQXR). MQ Telemetry fornisce altri due meccanismi di sicurezza, TLS e JAAS.

TLS viene utilizzato principalmente per codificare le comunicazioni tra il dispositivo e il canale di telemetria e per autenticare il dispositivo che si connette al server corretto; consultare Autenticazione del canale di telemetria utilizzando TLS. Puoi inoltre utilizzare TLS per controllare che la periferica client sia autorizzata a connettersi al server; vedi Autenticazione clientMQTT tramite TLS.

JAAS viene utilizzato principalmente per controllare che l'utente della periferica sia autorizzato a utilizzare un'applicazione server; consultare MQTT autenticazione client utilizzando una parola d'ordine. JAAS può essere utilizzato con LDAP per controllare una parola d'ordine utilizzando una directory SSO (single sign - on).

TLS e JAAS possono essere utilizzati insieme per fornire l'autenticazione a due fattori. È possibile limitare le cifrature utilizzate da TLS alle cifrature che soddisfano standard FIPS.

Con almeno decine di migliaia di utenti, non è sempre pratico fornire profili di sicurezza individuali. Inoltre, non è sempre pratico utilizzare i profili per autorizzare i singoli utenti ad accedere agli oggetti IBM MQ . Raggruppare invece gli utenti in classi per autorizzare la pubblicazione e la sottoscrizione agli argomenti e inviare pubblicazioni ai client.

Configurare ciascun canale di telemetria per associare i client agli ID utente client comuni. Utilizzare un ID utente comune per ogni client che si connette su un canale specifico; consultare Identità e autorizzazione clientMQTT.

L'autorizzazione di gruppi di utenti non compromette l'autenticazione di ogni individuo. Ogni singolo utente può essere autenticato, sul client o sul server, con i relativi `Nome utente` e `Password`e quindi autorizzato sul server utilizzando un ID utente comune.

Linux Windows AIX MQ Telemetry globalizzazione

Il payload del messaggio nel protocollo MQTT v3 è codificato come array di byte. In genere, le applicazioni che gestiscono il testo creano il payload del messaggio in UTF-8. Il canale di telemetria descrive il payload del messaggio come UTF-8, ma non esegue alcuna conversione di codepage. La stringa dell'argomento di pubblicazione deve essere UTF-8.

L'applicazione è responsabile della conversione dei dati alfabetici nella codepage corretta e dei dati numerici nella codifica numerica corretta.

Il client MQTT Java ha un metodo `MqttMessage.toString` conveniente. Il metodo considera il payload del messaggio come codificato nella serie di caratteri predefinita della piattaforma locale, generalmente UTF-8. Converte il payload in una stringa Java . Java dispone di un metodo String, `getBytes` che converte una stringa in una schiera di byte codificata utilizzando la serie di caratteri predefinita della piattaforma locale. Due programmi MQTT Java che scambiano testo nel payload del messaggio, tra piattaforme con la stessa serie di caratteri predefinita, lo fanno in modo semplice ed efficiente in UTF-8.

Se la serie di caratteri predefinita di una delle piattaforme non è UTF-8, le applicazioni devono stabilire una convenzione per lo scambio di messaggi. Ad esempio, il publisher specifica la conversione da una stringa a UTF-8 utilizzando il metodo `getBytes("UTF8")` . Per ricevere il testo di un messaggio, l'utente assume che il messaggio sia codificato nella serie di caratteri UTF-8 .

Il servizio di telemetria (MQXR) descrive la codifica di tutte le pubblicazioni in entrata dai messaggi dei client MQTT come UTF-8. Imposta `MQMD.CodedCharSetId` su UTF-8e `RFH2.CodedCharSetId` su `MQCCSI_INHERIT` ; consultare “Integrazione di MQ Telemetry con i gestori code” a pagina 134.
Il formato della pubblicazione è impostato su `MQFMT_NONE`, quindi nessuna conversione può essere eseguita dai canali o da `MQGET`.

Linux Windows AIX Prestazioni e scalabilità di MQ Telemetry

Considerare i seguenti fattori quando si gestiscono un numero elevato di client e si migliora la scalabilità di MQ Telemetry.

Pianificazione della capacità

Per informazioni sui report delle prestazioni per MQ Telemetry, consultare MQ Documenti delle prestazioni.

Connessioni

I costi relativi alle connessioni includono:

- Il costo dell'impostazione di una connessione in termini di tempo e utilizzo del processore.
- Costi di rete.
- Memoria utilizzata quando si mantiene una connessione aperta ma non la si utilizza.

C'è un carico aggiuntivo che si verifica quando i client rimangono connessi. Se una connessione viene mantenuta aperta, i flussi TCP/IP e i messaggi MQTT utilizzano la rete per controllare che la connessione sia ancora presente. Inoltre, la memoria viene utilizzata nel server per ogni connessione client mantenuta aperta.

Se si stanno inviando messaggi più di uno al minuto, mantenere la connessione aperta per evitare il costo di avviare una nuova connessione. Se si inviano messaggi meno di uno ogni 10-15 minuti, si consiglia di eliminare la connessione per evitare il costo di mantenerla aperta. Potresti voler mantenere una connessione TLS aperta, ma inattiva, per periodi più lunghi perché è più costosa da configurare.

Inoltre, considerare la capacità del client. Se sul client è presente una funzione di memorizzazione e inoltro, è possibile eseguire il batch dei messaggi ed eliminare la connessione tra l'invio dei batch. Tuttavia, se il client è disconnesso, non è possibile per il client ricevere un messaggio dal server. Pertanto, lo scopo della tua domanda ha un impatto sulla decisione.

Se il sistema dispone di un client che invia molti messaggi, ad esempio i trasferimenti file, non attendere una risposta del server per messaggio. Invece, inviare tutti i messaggi e verificare alla fine che siano stati tutti ricevuti. In alternativa, utilizzare Quality of Service (QoS).

È possibile variare il QoS in base al messaggio, consegnando messaggi non importanti utilizzando QoS 0 e messaggi importanti utilizzando un QoS di 2. La velocità di trasmissione dei messaggi può essere circa il doppio di quella con un QoS pari a 0 rispetto a QoS pari a 2.

Convenzioni di denominazione

Se si sta progettando l'applicazione per molti client, implementare una convenzione di denominazione efficace. Per associare ciascun client al `ClientIdentifier`corretto, rendere `ClientIdentifier` significativo. Una buona convenzione di denominazione rende più semplice per l'amministratore capire quali client sono in esecuzione. Una convenzione di denominazione aiuta l'amministratore a filtrare un lungo elenco di client in IBM MQ Explorer e aiuta nella determinazione dei problemi; consultare Identificatore client.

Velocità di trasmissione

La lunghezza dei nomi argomento influisce sul numero di byte che passano attraverso la rete. Durante la pubblicazione o la sottoscrizione, il numero di byte in un messaggio potrebbe essere importante. Pertanto, limitare il numero di caratteri in un nome argomento. Quando un MQTT client effettua la sottoscrizione per un argomento IBM MQ gli fornisce un nome del formato:

```
ClientIdentifier: TopicName
```

Per visualizzare tutte le sottoscrizioni per un client MQTT , è possibile utilizzare il comando IBM MQ MQSC **DISPLAY** :

```
DISPLAY SUB(' ClientID1:*')
```

Definizione delle risorse in IBM MQ per l'utilizzo da parte dei client MQTT

Un client MQTT si connette a un gestore code remoto IBM MQ . Esistono due metodi di base per un'applicazione IBM MQ per inviare i messaggi a un client MQTT : impostare la coda di trasmissione predefinita su `SYSTEM.MQTT.TRANSMIT.QUEUE` o utilizzare gli alias del gestore code. Definire la coda di trasmissione predefinita di un gestore code, se è presente un numero elevato di client MQTT . L'uso dell'impostazione della coda di trasmissione predefinita semplifica l'amministrazione; consultare Configurazione dell'accodamento distribuito per inviare messaggi ai client MQTT ..

Migliorare la scalabilità evitando le sottoscrizioni.

Quando un client MQTT V3 sottoscrive un argomento, viene creata una sottoscrizione dal servizio di telemetria (MQXR) in IBM MQ. La sottoscrizione instrada le pubblicazioni per il client su `SYSTEM.MQTT.TRANSMIT.QUEUE`. Il nome del gestore code remoto nell'intestazione di trasmissione di ogni pubblicazione è impostato su `ClientIdentifier` del client MQTT che ha eseguito la sottoscrizione. Se vi sono molti client, ognuno dei quali effettua le proprie sottoscrizioni, ciò comporta la gestione di molte sottoscrizioni proxy in tutta la gerarchia o il cluster di pubblicazione / sottoscrizione IBM MQ . Per informazioni su come non utilizzare la pubblicazione / sottoscrizione, ma utilizzare invece una soluzione basata punto a punto, consultare Invio di un messaggio direttamente a un client.

Gestione di un gran numero di client

Per supportare molti client connessi simultaneamente, aumentare la memoria disponibile per il servizio di telemetria (MQXR) impostando i parametri JVM **-Xms** e **-Xmx**. Eseguire queste operazioni:

1. Individuare il file `java.properties` nella directory di configurazione del servizio di telemetria; consultare Directory di configurazione del servizio di telemetria (MQXR) su Windows o Directory di configurazione del servizio di telemetria su Linux .
2. Seguire le istruzioni nel file; un heap di 1 GB è sufficiente per 50.000 client connessi contemporaneamente.

```
# Heap sizing options - uncomment the following lines to set the heap to 1G
#-Xmx1024m
#-Xms1024m
```

3. Aggiungere altri argomenti della riga comandi da passare alla JVM che esegue il servizio di telemetria (MQXR) nel file `java.properties` ; consultare Passaggio dei parametri JVM al servizio di telemetria (MQXR).

Per aumentare il numero di descrittori di file aperti su Linux, aggiungere le seguenti righe a `/etc/security/limits.conf`/ed eseguire nuovamente l'accesso.

```
@mqm soft nofile 65000
@mqm hard nofile 65000
```

Ogni socket richiede un descrittore di file. Il servizio di telemetria richiede alcuni descrittori di file aggiuntivi, quindi questo numero deve essere maggiore del numero di socket aperti richiesti.

Il gestore code utilizza un handle di oggetto per ogni sottoscrizione non durevole. Per supportare molte sottoscrizioni attive e non durevoli, aumentare il numero massimo di handle attivi nel gestore code; ad esempio:

```
echo ALTER QMGR MAXHANDS(999999999) | runmqsc qMgrName
```

Figura 48. Modifica numero massimo di handle su Windows

```
echo "ALTER QMGR MAXHANDS(999999999)" | runmqsc qMgrName
```

Figura 49. Modifica numero massimo di handle su Linux

Altre considerazioni

Quando si pianificano i requisiti di sistema, considerare il tempo impiegato per riavviare il sistema. Il tempo di inattività pianificato potrebbe avere implicazioni per il numero di messaggi in coda, in attesa di essere elaborati. Configurare il sistema in modo che i messaggi possano essere elaborati correttamente in un tempo accettabile. Esaminare l'archiviazione disco, la memoria e la potenza di elaborazione. Con alcune applicazioni client, è possibile eliminare i messaggi quando il client si riconnette. Per eliminare i messaggi, impostare `CleanSession` nei parametri di collegamento client; consultare Sessione pulita. In alternativa, pubblica e sottoscrivi utilizzando il QoS (Quality of Service) più efficace, `0`, in un client MQTT ; vedi QoS (Quality of service). Utilizzare messaggi non `persistenti` quando si inviano messaggi da IBM MQ. I messaggi con queste QoS (quality of service) non vengono ripristinati al riavvio del sistema o della connessione.

Linux Windows AIX Dispositivi supportati da MQ Telemetry

I clienti MQTT possono essere eseguiti su una vasta gamma di dispositivi, da sensori e attuatori, a dispositivi portatili e sistemi di veicoli.

I client MQTT sono di piccole dimensioni e vengono eseguiti su dispositivi con poca memoria e bassa potenza di elaborazione. Il MQTT protocol è affidabile e ha intestazioni piccole, che si adattano alle reti limitate da una larghezza di banda bassa, costi elevati e disponibilità intermittente.

MQ Telemetry comunica con i dispositivi di telemetria tramite applicazioni client MQTT . Queste applicazioni utilizzano le seguenti risorse, che implementano il protocollo MQTT v3 :

- Le seguenti librerie client:
  - *MQTT client for Java*, utilizzato per la creazione di applicazioni native per (ad esempio) dispositivi Android, OS X, Linux o Windows . Le applicazioni che utilizzano questa libreria client possono essere eseguite su tutte le varianti di Java dal più piccolo CLDC (Connected Limited Device Configuration) / MIDP (Mobile Information Device Profile) al CDC (Connected Device Configuration) /Foundation, J2SE (Java Platform, Standard Edition) e J2EE (Java Platform, Enterprise Edition). È supportata anche la libreria di classi personalizzata IBM jclRM . La piattaforma Java ME viene generalmente utilizzata su dispositivi di piccole dimensioni, come attuatori, sensori, telefoni cellulari e altri dispositivi incorporati. La piattaforma Java SE è generalmente installata su dispositivi integrati di fascia superiore, come computer desktop e server.
  - *MQTT client for C*, utilizzato per la creazione di applicazioni native per (ad esempio) periferiche iOS, OS X, Linux o Windows . Questa libreria del client fornisce un'implementazione di riferimento C insieme al client nativo precostruito per sistemi Windows e Linux . L'implementazione del riferimento C consente a MQTT di essere portato su una vasta gamma di dispositivi e piattaforme. Alcuni sistemi Windows su Intel, inclusi Windows 7, RedHat, Ubuntue alcuni sistemi Linux su piattaforme ARM come Eurotech Viper implementano versioni di Linux che eseguono il client C, ma IBM non fornisce il supporto di servizio per le piattaforme. È necessario riprodurre i problemi con il client su una piattaforma supportata se si intende chiamare il centro di supporto IBM .
  - *MQTT client for Java*, utilizzato per la creazione di applicazioni Web basate su browser.

Le librerie client MQTT sono disponibili gratuitamente da Eclipse Paho e MQTT.org. Vedere IBM MQ Telemetry Transport programmi di esempio.

Sicurezza in IBM MQ

In IBM MQ, esistono diversi metodi per fornire la sicurezza: l'interfaccia del servizio di autorizzazione, le uscite del canale scritte dall'utente o di terze parti, la sicurezza del canale utilizzando TLS (Transport Layer Security), i record di autenticazione del canale e la sicurezza dei messaggi.

Interfaccia servizio di autorizzazione

L'autorizzazione per l'utilizzo di chiamate MQI, comandi e accesso agli oggetti è fornita da **gestore autorizzazioni oggetto** (OAM), che per impostazione predefinita è abilitata. L'accesso alle entità IBM MQ è controllato tramite i gruppi di utenti IBM MQ e OAM. Gli amministratori possono utilizzare una CLI (command - line interface) per concedere o revocare le autorizzazioni come richiesto.

Per ulteriori informazioni sulla creazione dei componenti del servizio di autorizzazione, consultare Impostazione della sicurezza sui sistemi AIX, Linux, and Windows.

Uscite canale di terze parti o scritte dall'utente

I canali possono utilizzare uscite di canali scritte dall'utente o di terze parti. Per ulteriori informazioni, consultare Channel - exit programs for messaging channels.

Sicurezza del canale utilizzando TLS

Il protocollo TLS (Transport Layer Security) fornisce la sicurezza del canale standard del settore, con protezione da intercettazione, manomissione e impersonificazione.

TLS utilizza tecniche di chiave pubblica e simmetriche per fornire riservatezza e integrità dei messaggi e autenticazione reciproca.

Per una revisione completa della sicurezza in IBM MQ incluse informazioni dettagliate su TLS, vedi Sicurezza. Per una panoramica di TLS, inclusi i puntatori ai comandi descritti in questa sezione, consultare Protocolli di sicurezza crittografici: TLS.

Record di autenticazione di canale

Utilizzare i record di autenticazione di canale per esercitare un controllo preciso sull'accesso concesso ai sistemi di collegamento a livello di canale. Per ulteriori informazioni, consultare Record di autenticazione di canale.

Sicurezza messaggi

Utilizzare Advanced Message Security, che è un componente con licenza e installato separatamente di IBM MQ, per fornire una protezione crittografica ai messaggi inviati e ricevuti utilizzando IBM MQ. Vedere Advanced Message Security.

Attività correlate

Protezione

Pianificazione dei requisiti di sicurezza

Supporto TLS del client gestito IBM MQ.NET

Il client IBM MQ.NET completamente gestito fornisce il supporto TLS (Transport Layer Security) basato sul kit Microsoft.NET SSLStreams. È diverso dagli altri client IBM MQ , che si basano su IBM Global Security Kit (GSKit).

È possibile sviluppare applicazioni IBM MQ.NET da eseguire in modalità gestita o non gestita.

- In modalità gestita, le applicazioni .NET funzionano all'interno di .NET CLR (Common Language Runtime) senza alcun richiamo tra piattaforme, come il richiamo di C MQI.
- In modalità non gestita, C MQI viene richiamata per le operazioni MQI sottostanti. In pratica, l'interfaccia in modalità non gestita comprende le classi wrapper .NET in cima alla C MQI.

Il client IBM MQ.NET gestito utilizza le librerie Microsoft.NET Framework per implementare i protocolli socket sicuri TLS. Il sistema.NET.Security.SSLStream da Microsoft viene utilizzata per implementare la sicurezza (TLS) in IBM MQ.NET.

La modalità client IBM MQ.NET non gestita supporta già la funzione TLS, che è basata su C MQ (e GSKit). In altre parole, le operazioni TLS sono gestite da C MQI. In questo caso, GSKit implementa i protocolli socket protetti TLS.

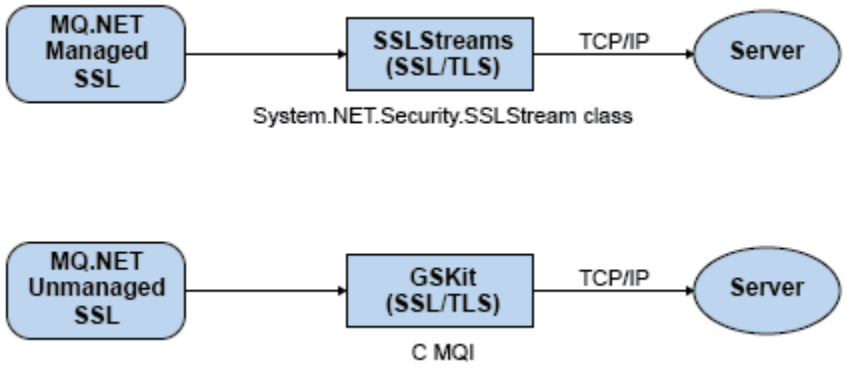


Figura 50. Confronto TLS gestito e non gestito di IBM MQ.NET

La seguente tabella riepiloga le differenze tra le implementazioni gestite e non gestite:

Tabella 14. Differenze tra implementazioni gestite e non gestite			
Modalità	**Protocolli**	**Implementazione**	**Commenti**
SSL gestito da IBM MQ.NET	TLS	Sistema.NET.Security.SSLStream SSLStream La classe SSLStream funziona come un flusso su un socket TCP connesso	TLS 1.0 TLS 1.2 (solo con Microsoft.NET Framework v4.5)
IBM MQ.NET SSL non gestito	TLS	GSKIT e C-MQI	Protocolli socket sicuri TLS

Concetti correlati

Supporto SSL (Secure Sockets Layer) e TLS (Transport Layer Security) per .NET

IBM MQ MQI clients

Un IBM MQ MQI client è un componente del prodotto IBM MQ che può essere installato su un sistema su cui non è in esecuzione alcun gestore code.

Un IBM MQ client MQI è un componente che permette a un'applicazione in esecuzione su un sistema di emettere chiamate MQI a un gestore code in esecuzione su un altro sistema. L'output della chiamata viene inviato di nuovo al client, che lo restituisce all'applicazione.

Utilizzando un IBM MQ MQI client, un'applicazione in esecuzione sullo stesso sistema del client può connettersi a un gestore code in esecuzione su un altro sistema. L'applicazione può emettere chiamate MQI a tale gestore code. Tale applicazione è denominata IBM MQ MQI client e il gestore code è denominato *gestore code server*.

Un server IBM MQ è un gestore code che fornisce servizi di accodamento a uno o più client. Tutti gli oggetti IBM MQ , ad esempio le code, esistono solo sulla macchina del gestore code (la macchina server IBM MQ) e non sul client. Un server IBM MQ può anche supportare applicazioni IBM MQ locali.

La differenza tra un server IBM MQ e un gestore code ordinario è che un server ha un collegamento di comunicazioni dedicato con ciascun client. Per ulteriori informazioni sulla creazione di canali per client e server, consultare Configurazione dell'accodamento distribuito.

Un'applicazione IBM MQ MQI client e un gestore code del server comunicano tra loro utilizzando un canale *MQI*. Un canale MQI viene avviato quando l'applicazione client emette una chiamata **MQCONN** o **MQCONNX** per connettersi al gestore code e termina quando l'applicazione client emette una chiamata **MQDISC** per disconnettersi dal gestore code. I parametri di input di un flusso di chiamate MQI in una direzione su un canale MQI e i parametri di output nella direzione opposta.

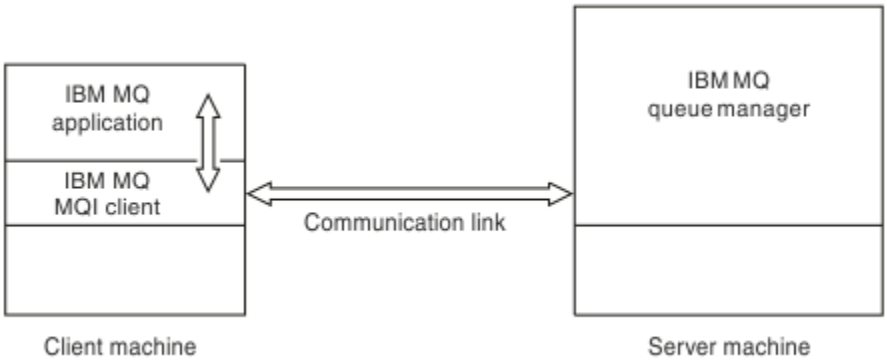


Figura 51. Collegamento tra un client e un server

È possibile utilizzare le piattaforme riportate di seguito. Le combinazioni dipendono da quale prodotto IBM MQ si sta utilizzando e sono descritte in “Supporto della piattaforma per client IBM MQ” a pagina 146.

IBM MQ MQI client	IBM MQ server
AIX and Linux Windows IBM i	AIX and Linux Windows IBM i z/OS

La MQI è disponibile per le applicazioni in esecuzione sulla piattaforma client; le code e gli altri oggetti IBM MQ si trovano su un gestore code installato su un server.

Un'applicazione che si desidera eseguire nell'ambiente IBM MQ MQI client deve essere prima collegata alla libreria client pertinente. Quando l'applicazione emette una chiamata MQI, IBM MQ MQI client indirizza la richiesta a un gestore code, dove viene elaborata e da dove viene inviata una risposta a IBM MQ MQI client.

Il collegamento tra l'applicazione e IBM MQ MQI client viene stabilito dinamicamente in fase di runtime.

È anche possibile sviluppare applicazioni client utilizzando IBM MQ classes for .NET, IBM MQ classes for Java o IBM MQ classes for Java Message Service (JMS). È possibile utilizzare i client Java e JMS sulle seguenti piattaforme:

- IBM i
- AIX
- Linux
- Windows

L'utilizzo di Java e JMS non è descritto qui. Per i dettagli completi su come installare, configurare e utilizzare IBM MQ classes for Java e IBM MQ classes for JMS consultare Utilizzo di IBM MQ classes for Java e Utilizzo di IBM MQ classes for JMS.

Applicazioni IBM MQ in un ambiente client-server

Quando sono collegati a un server, le applicazioni IBM MQ del client possono emettere la maggior parte delle chiamate MQI allo stesso modo delle applicazioni locali. L'applicazione client emette una chiamata `MQCONN` per connettersi a un gestore code specificato. Tutte le chiamate MQI aggiuntive che specificano l'handle di connessione restituito dalla richiesta di connessione vengono quindi elaborate da questo gestore code.

È necessario collegare le applicazioni alle librerie client appropriate. Consultare Creazione di applicazioni per IBM MQ MQI clients.

Concetti correlati

“Perché utilizzare i client IBM MQ ?” a pagina 146
L'uso dei client IBM MQ è un modo efficiente di implementare la messaggistica e l'accodamento IBM MQ .

“Cos' è un client transazionale esteso?” a pagina 147
Un client transazionale esteso IBM MQ può aggiornare le risorse gestite da un altro gestore risorse, sotto il controllo di un gestore transazioni esterno.

“Modalità di connessione del client al server” a pagina 149
Un client si connette a un server utilizzando MQCONN o MQCONNX e comunica attraverso un canale.

“Gestione e supporto delle transazioni” a pagina 150
Introduzione alla gestione delle transazioni e al modo in cui IBM MQ supporta le transazioni.

“Estensione delle funzioni del gestore code” a pagina 151

È possibile estendere le funzioni del gestore code utilizzando le uscite utente, le uscite API o i servizi installabili.

Informazioni correlate

Come configurare un IBM MQ MQI client

Perché utilizzare i client IBM MQ ?

L'uso dei client IBM MQ è un modo efficiente di implementare la messaggistica e l'accodamento IBM MQ .

È possibile avere un'applicazione che utilizza MQI in esecuzione su una macchina e il gestore code in esecuzione su una macchina differente (fisica o virtuale). I vantaggi di questa operazione sono:

- Non è necessaria un'implementazione IBM MQ completa sulla macchina client.
- I requisiti hardware sul sistema client sono ridotti.
- I requisiti di amministrazione del sistema sono ridotti.
- Un'applicazione IBM MQ in esecuzione su un client può connettersi a più gestori code su sistemi differenti.
- Possono essere utilizzati canali alternativi che utilizzano protocolli di trasmissione differenti.

Supporto della piattaforma per client IBM MQ

IBM MQ su tutte le piattaforme server supportate accetta le connessioni client da IBM MQ MQI clients su un numero di piattaforme.

IBM MQ installato come prodotto di base *e server* su tutte le piattaforme server supportate può accettare connessioni da IBM MQ MQI clients sulle piattaforme seguenti:

- IBM i
- AIX
- Linux
- Windows

I collegamenti client sono soggetti a differenze nel CCSID (Coded Character Set Identifier) e nel protocollo di comunicazioni.

Quali applicazioni vengono eseguite su un IBM MQ MQI client?

MQI completo è supportato nell'ambiente client. Ciò consente a quasi tutte le applicazioni IBM MQ di essere configurate per essere eseguite su un sistema IBM MQ MQI client collegando l'applicazione su IBM MQ MQI client alla libreria MQIC, piuttosto che alla libreria MQI. Le eccezioni sono:

- MQGET con segnale
- Un'applicazione che richiede il coordinamento del punto di sincronizzazione con altri gestori risorse deve utilizzare un client transazionale esteso

Se la lettura anticipata è abilitata, per migliorare le prestazioni della messaggistica non persistente, non sono disponibili tutte le opzioni MQGET. La tabella mostra le opzioni consentite e se è possibile modificarle tra le chiamate MQGET.

Tabella 15. Opzioni MQGET consentite quando la lettura anticipata è abilitata

Valori	**Consentito quando la lettura anticipata è abilitata e può essere modificato tra le chiamate MQGET**	**Consentito quando la lettura anticipata è abilitata ma non può essere modificato tra chiamate MQGET** [1]	**Le opzioni MQGET non consentite quando la lettura anticipata è abilitata** [2]
Valori MQGET MD	MsgId [3] CorrelId [3]	Codifica CodedCharSetId	

Tabella 15. Opzioni MQGET consentite quando la lettura anticipata è abilitata (Continua)			
Valori	**Consentito quando la lettura anticipata è abilitata e può essere modificato tra le chiamate MQGET**	**Consentito quando la lettura anticipata è abilitata ma non può essere modificato tra chiamate MQGET [1]**	**Le opzioni MQGET non consentite quando la lettura anticipata è abilitata [2]**
Opzioni MQGET MQGMO	MQGMO_WAIT MQGMO_NO_WAIT MQGMO_FAIL_IF_QUIESCING MQGMO_BROWSE_FIRST [4] MQGMO_BROWSE_NEXT [4] MESSAGGIO_BROWSE_MQGMO_ _UNDER_CURSOR [4]	MQGMO_SYNCPOINT_IF_PERSISTENT MQGMO_NO_SYNCPOINT MQGMO_ACCEPT_TRUNCATED_MSG MQGMO_CONVERT ORDER LOGICAL_MQGMO_ MQGMO_COMPLETE_MSG MQGMO_ALL_MSGS_AVAILABLE MQGMO_ALL_SEGMENTS_AVAILABLE MQGMO_MARK_BROWSE_HANDLE MQGMO_MARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_HANDLE MQGMO_UNMARKED_BROWSE_MSG MQGMO_PROPERTIES_FORCE_MQRFH2 MQGMO_NO_PROPERTIES MQGMO_PROPERTIES_IN_HANDLE MQGMO_PROPERTIES_COMPATIBILITY	MQGMO_SET_SIGNAL SYNCPOINT MQGMO SKIP_MARK_MQGMO _BACKOUT MQGMO_MSG_UNDER _CURSORE [4] LOCK_MQGMO MQGMO_UNLOCK
Valori MQGMO		MsgHandle	

1. Se queste opzioni vengono modificate tra le chiamate MQGET, viene restituito un codice motivo MQRC_OPTIONS_CHANGED.
2. Se queste opzioni vengono specificate nella prima chiamata MQGET, il read ahead è disabilitato. Se queste opzioni vengono specificate in una successiva chiamata MQGET, viene restituito il codice motivo MQRC_OPTIONS_ERROR.
3. Le applicazioni client devono essere consapevoli che se i valori MsgId e CorrelId vengono modificati tra le chiamate MQGET, i messaggi con i valori precedenti potrebbero essere già stati inviati al client e rimanere nel buffer di lettura anticipata del client finché non vengono consumati (o automaticamente eliminati).
4. La prima chiamata MQGET determina se i messaggi devono essere ricercati o richiamati da una coda quando il read ahead è abilitato. Se l'applicazione tenta di utilizzare una combinazione di BROWSE e GET, viene restituito il codice motivo MQRC_OPTIONS_CHANGED.
5. MQGMO_MSG_UNDER_CURSOR non è consentito con il read ahead. Quando il read ahead è abilitato, i messaggi possono essere sfogliati o richiamati, ma non entrambe le cose.

Un'applicazione in esecuzione su un IBM MQ MQI client può connettersi a più di un gestore code contemporaneamente oppure utilizzare un nome gestore code con un asterisco (*) su una chiamata MQCONN o MQCONNX (consultare gli esempi in Connessione di applicazioni IBM MQ MQI client ai gestori code).

Cos' è un client transazionale esteso?

Un client transazionale esteso IBM MQ può aggiornare le risorse gestite da un altro gestore risorse, sotto il controllo di un gestore transazioni esterno.

Se non si ha familiarità con i concetti di gestione delle transazioni, consultare “Gestione e supporto delle transazioni” a pagina 150.

Notare che il client transazionale XA viene ora fornito come parte di IBM MQ.

Un'applicazione client può partecipare a un'unità di lavoro gestita da un gestore code a cui è connessa. All'interno dell'unità di lavoro, l'applicazione client può inserire e richiamare messaggi dalle code di proprietà di tale gestore code. L'applicazione client può quindi utilizzare la chiamata **MQCMIT** per eseguire il commit dell'unità di lavoro o la chiamata **MQBACK** per eseguire il backout dell'unità di lavoro. Tuttavia, all'interno della stessa unità di lavoro, l'applicazione client non può aggiornare le risorse di un altro gestore risorse, ad esempio le tabelle di un database Db2 . L'uso di un client transazionale esteso IBM MQ elimina questa limitazione.

Un client transazionale esteso IBM MQ è un IBM MQ MQI client con alcune funzioni aggiuntive. Utilizzando questa funzione un'applicazione client, all'interno della stessa unità di lavoro, può eseguire le seguenti attività:

- Inserire e richiamare i messaggi dalle code di proprietà del gestore code a cui è connesso
- Aggiornare le risorse di un gestore risorse diverso da un gestore code IBM MQ

Questa unità di lavoro deve essere gestita da un gestore transazioni esterno in esecuzione sullo stesso sistema dell'applicazione client. L'unità di lavoro non può essere gestita dal gestore code a cui è connessa l'applicazione client. Ciò significa che il gestore code può agire solo come gestore risorse e non come gestore transazioni. Ciò significa anche che l'applicazione client può eseguire il commit o il backout dell'unità di lavoro utilizzando solo l'API (application programming interface) fornita dal gestore transazioni esterno. L'applicazione client, pertanto, non può utilizzare le chiamate MQI, **MQBEGIN**, **MQCMIT**e **MQBACK**.

Il gestore transazioni esterno comunica con il gestore code come gestore risorse utilizzando lo stesso canale MQI utilizzato dall'applicazione client connessa al gestore code. Tuttavia, in una situazione di ripristino in seguito a un errore, quando nessuna applicazione è in esecuzione, il gestore transazioni può utilizzare un canale MQI dedicato per ripristinare le unità di lavoro incomplete a cui il gestore code partecipava al momento dell'errore.

In questa sezione, un IBM MQ MQI client che non dispone della funzione transazionale estesa viene indicato come client di base IBM MQ . È possibile considerare, quindi, un client transazionale esteso IBM MQ composto da un client di base IBM MQ con aggiunta della funzione transazionale estesa.

Nota: IBM i IBM MQ MQI client su IBM i non supporta la funzione transazionale estesa IBM MQ .

Supporto della piattaforma per i client transazionali estesi

Multi

I client transazionali estesi sono disponibili per tutte le piattaforme multiple che supportano un client di base. I client non sono disponibili per z/OS.

Un'applicazione client che utilizza un client transazionale esteso può connettersi solo a un gestore code dei seguenti prodotti IBM MQ 9.0o successivi:

- AIX IBM MQ for AIX
- IBM i IBM MQ for IBM i
- Linux IBM MQ per Linux
- Windows IBM MQ for Windows

z/OS Anche se non ci sono client transazionali estesi in esecuzione su z/OS, un'applicazione client che utilizza un client transazionale esteso può connettersi a un gestore code in esecuzione su z/OS.

Per ogni piattaforma, i requisiti hardware e software per il client transazionale esteso corrispondono a quelli per il client di base IBM MQ . Un linguaggio di programmazione è supportato da un client transazionale esteso se è supportato dal client di base IBM MQ e dal gestore transazioni che si sta utilizzando.

Per informazioni sui gestori transazioni esterni per tutte le piattaforme, consultare Requisiti di sistema per IBM MQ.

Modalità di connessione del client al server

Un client si connette a un server utilizzando MQCONN o MQCONNX e comunica attraverso un canale.

Un'applicazione in esecuzione nell'ambiente client IBM MQ deve mantenere una connessione attiva tra le macchine client e server.

La connessione viene effettuata da un'applicazione che emette una chiamata MQCONN o MQCONNX. I client e i server comunicano tramite i *canali MQI*oppure, quando si utilizzano conversazioni condivise, ogni conversazione condivide un'istanza del canale MQI. Quando la chiamata ha esito positivo, la conversazione o l'istanza del canale MQI rimane connessa finché l'applicazione non emette una chiamata MQDISC. Questo è il caso di ogni gestore code a cui un'applicazione deve connettersi.

Client e gestore code sulla stessa macchina

È inoltre possibile eseguire un'applicazione nell'ambiente IBM MQ MQI client quando sulla macchina è installato anche un gestore code.

In questa situazione, si ha la possibilità di collegarsi alle librerie del gestore code o alle librerie del client, ma tenere presente che se si collega alle librerie del client, è comunque necessario definire le connessioni del canale. Ciò può essere utile durante la fase di sviluppo di una applicazione. È possibile testare il programma sulla propria macchina, senza dipendere da altri, ed essere certi che funzionerà ancora quando lo si sposta in un ambiente IBM MQ MQI client indipendente.

Client su diverse piattaforme

In questo esempio, la macchina server comunica con tre IBM MQ MQI clients su piattaforme differenti.

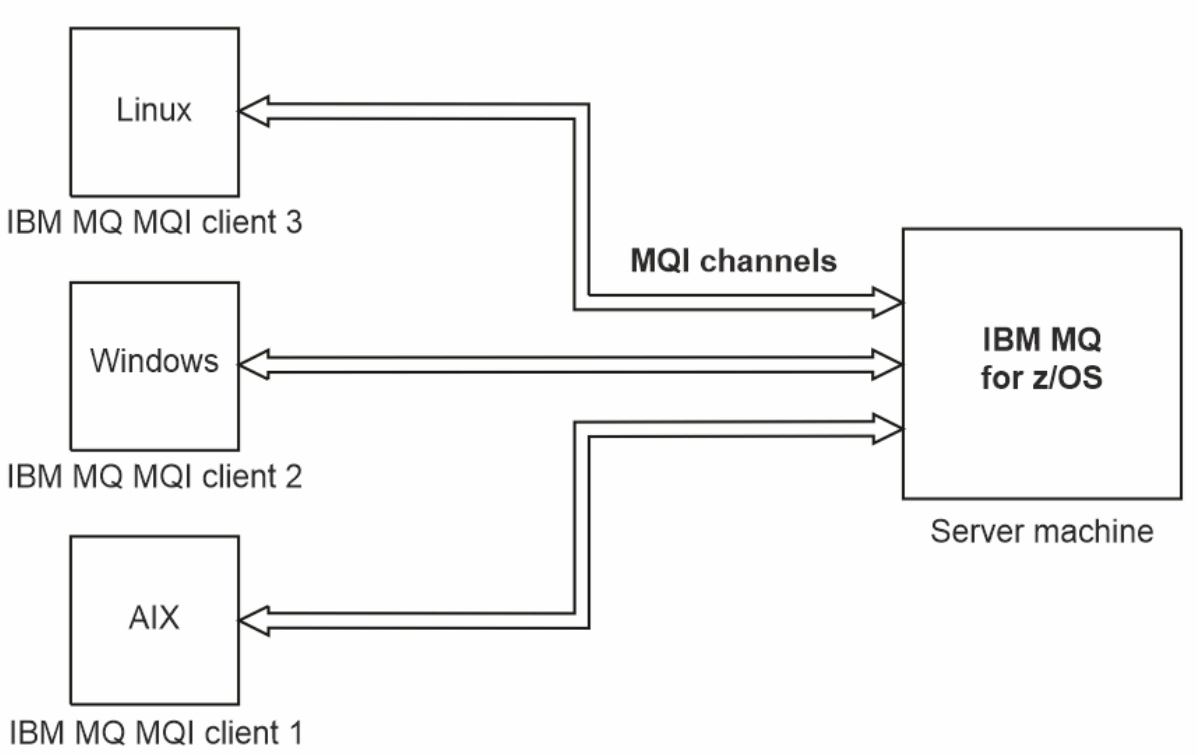


Figura 52. Server IBM MQ connesso a client su piattaforme differenti

Altri ambienti più complessi sono possibili. Ad esempio, un client IBM MQ può connettersi a più di un gestore code o a qualsiasi numero di gestori code connessi come parte di un gruppo di condivisione code.

Utilizzo di versioni differenti di software client e server

Se si stanno utilizzando versioni precedenti di prodotti IBM MQ , assicurarsi che la conversione del codice dal CCSID del proprio client sia supportato dal server.

Un client IBM MQ può connettersi a tutte le versioni supportate del gestore code. Se ci si connette a un gestore code di una versione precedente, non è possibile utilizzare le funzioni e le strutture di una versione successiva del prodotto nell'applicazione IBM MQ sul client.

Un gestore code IBM MQ può comunicare con i client di versioni diverse con se stesso negoziando fino al livello di protocollo supportato reciprocamente più alto. Ciò significa che i client più vecchi possono essere utilizzati con livelli di gestore code successivi. Si consiglia che sia il client che il server siano alle versioni di IBM MQ attualmente supportate per facilitare la diagnostica dei problemi e abilitare il supporto da parte di IBM.

Per ulteriori informazioni, consultare i linguaggi di programmazione supportati in Sviluppo delle applicazioni.

Gestione e supporto delle transazioni

Introduzione alla gestione delle transazioni e al modo in cui IBM MQ supporta le transazioni.

Un *gestore risorse* è un sottosistema di computer che possiede e gestisce le risorse a cui le applicazioni possono accedere e che possono essere aggiornate. Di seguito sono riportati esempi di gestori risorse:

- Un gestore code IBM MQ , con le relative risorse
- Un database Db2 , con le risorse che sono le relative tabelle

Quando un'applicazione aggiorna le risorse di uno o più gestori risorse, potrebbe esserci un requisito di business per garantire che alcuni aggiornamenti vengano completati correttamente come un gruppo o che nessuno di essi venga completato. Il motivo per questo tipo di requisito è che i dati di business verrebbero lasciati in uno stato incongruente se alcuni di questi aggiornamenti fossero stati completati correttamente, ma altri no.

Gli aggiornamenti alle risorse gestite in questo modo si verificano all'interno di un' *unità di lavoro*o di *transazione*. Un programma applicativo può raggruppare una serie di aggiornamenti in un'unità di lavoro.

Durante un'unità di lavoro, un'applicazione invia richieste ai gestori risorse per aggiornare le relative risorse. L'unità di lavoro termina quando l'applicazione emette una richiesta di commit di tutti gli aggiornamenti. Fino a quando non viene eseguito il commit degli aggiornamenti, nessuna di esse diventa visibile alle altre applicazioni che accedono alle stesse risorse. In alternativa, se l'applicazione decide di non poter completare l'unità di lavoro per qualsiasi motivo, può emettere una richiesta di backout di tutti gli aggiornamenti che ha richiesto fino a quel momento. In questo caso, nessuno degli aggiornamenti diventa mai visibile ad altre applicazioni. Questi aggiornamenti sono di solito correlati logicamente e devono essere tutti corretti per preservare l'integrità dei dati. Se un aggiornamento ha esito positivo mentre un altro ha esito negativo, l'integrità dei dati viene persa.

Quando un'unità di lavoro viene completata correttamente, viene detto *commit*. Una volta eseguito il commit, tutti gli aggiornamenti effettuati all'interno di tale unità di lavoro sono resi permanenti e irreversibili. Tuttavia, se l'unità di lavoro non riesce, tutti gli aggiornamenti vengono *ripristinati*. Questo processo, in cui le unità di lavoro vengono sottoposte a commit o a backout con integrità, è noto come *coordinamento del punto di sincronizzazione*.

Il momento in cui viene eseguito il commit o il backout di tutti gli aggiornamenti all'interno di un'unità di lavoro viene definito *punto di sincronizzazione*. Un aggiornamento all'interno di un'unità di lavoro si verifica *all'interno del controllo del punto di sincronizzazione*. Se un'applicazione richiede un aggiornamento *esterno al controllo del punto di sincronizzazione*, il gestore risorse esegue il commit dell'aggiornamento immediatamente, anche se è presente un'unità di lavoro in corso e non è possibile eseguire il backout dell'aggiornamento in un secondo momento.

Il sottosistema del computer che gestisce le unità di lavoro è denominato *gestore transazioni*o *coordinatore del punto*.

Un'unità di lavoro *locale* è un'unità in cui le uniche risorse aggiornate sono quelle del gestore code IBM MQ . Qui il coordinamento del punto di sincronizzazione viene fornito dal gestore code stesso utilizzando un processo di commit a fase singola.

Un'unità di lavoro *globale* è un'unità di lavoro in cui vengono aggiornate anche le risorse appartenenti ad altri gestori risorse, ad esempio i database compatibili con XA. Qui, è necessario utilizzare una procedura di commit a due fasi e l'unità di lavoro può essere coordinata dal gestore code stesso o esternamente da un altro gestore transazioni compatibile con XA come IBM TXSerieso BEA Tuxedo.

Un gestore transazioni è responsabile di garantire che tutti gli aggiornamenti alle risorse all'interno di un'unità di lavoro vengano completati correttamente o che nessuno di essi venga completato. È a un gestore transazioni che un'applicazione emette una richiesta per eseguire il commit o il backout di un'unità di lavoro. Esempi di gestori transazioni sono CICS e WebSphere Application Server, sebbene entrambi dispongano di altre funzioni.

Alcuni gestori risorse forniscono la propria funzione di gestione transazioni. Ad esempio, un gestore code IBM MQ può gestire unità di lavoro che includono aggiornamenti alle proprie risorse e aggiornamenti alle tabelle Db2 . Il gestore code non ha bisogno di un gestore transazioni separato per eseguire questa funzione, anche se è possibile utilizzarne uno se si tratta di un requisito utente. Se viene utilizzato un gestore transazioni separato, viene indicato come *gestore transazioni esterno*.

Affinché un gestore transazioni esterno gestisca un'unità di lavoro, deve esistere un'interfaccia standard tra il gestore transazioni e ogni gestore risorse che partecipa all'unità di lavoro. Questa interfaccia consente al gestore transazioni e a un gestore risorse di comunicare tra loro. Una di queste interfacce è l' *Interfaccia XA*, che è un'interfaccia standard supportata da diversi gestori transazioni e gestori risorse. L'interfaccia XA viene pubblicata da Open Group in *Distributed Transaction Processing: The XA Specification*.

Quando più di un gestore risorse partecipa a un'unità di lavoro, un gestore transazioni deve utilizzare un protocollo di *commit a due fasi* per garantire che tutti gli aggiornamenti all'interno dell'unità di lavoro vengano completati correttamente o nessuno di essi venga completato, anche se si verifica un errore di sistema. Quando un'applicazione invia una richiesta a un gestore transazioni per eseguire il commit di un'unità di lavoro, il gestore transazioni effettua le seguenti operazioni:

Fase 1 (preparazione al commit)
: Il gestore transazioni richiede a ciascun gestore risorse che partecipa all'unità di lavoro di verificare che tutte le informazioni relative agli aggiornamenti previsti per le relative risorse siano in uno stato ripristinabile. Un gestore risorse normalmente esegue questa operazione scrivendo le informazioni in un log e verificandone la scrittura sul disco fisso. La fase 1 viene completata quando il gestore transazioni riceve la notifica da ciascun gestore risorse che le informazioni relative agli aggiornamenti previsti per le relative risorse si trovano in uno stato ripristinabile.

Fase 2 (commit)
: Al termine della fase 1, il gestore della transazione prende la decisione irrevocabile di impegnare l'unità di lavoro. Chiede a ciascun gestore risorse che partecipa all'unità di lavoro di eseguire il commit degli aggiornamenti alle proprie risorse. Quando un gestore risorse riceve questa richiesta, deve eseguire il commit degli aggiornamenti. Non ha la possibilità di eseguirne il backout in questo momento. La fase 2 viene completata quando il gestore transazioni riceve una notifica da ciascun gestore risorse che ha eseguito il commit degli aggiornamenti alle proprie risorse.

L'interfaccia XA utilizza un protocollo di commit a due fasi.

Per ulteriori informazioni, consultare Scenari di supporto transazionale.

IBM MQ fornisce inoltre il supporto per Microsoft Transaction Server (COM +). L'utilizzo di Microsoft Transaction Server (COM +) fornisce informazioni su come configurare IBM MQ per sfruttare il supporto COM +.

Estensione delle funzioni del gestore code

È possibile estendere le funzioni del gestore code utilizzando le uscite utente, le uscite API o i servizi installabili.

Uscite Utente

Le uscite utente forniscono un meccanismo per inserire il proprio codice in una funzione del gestore code. Le uscite utente supportate includono:

Uscite canale
Queste uscite cambiano il modo in cui operano i canali. Le uscite canale sono descritte in Programmi di uscita canale per i canali di messaggistica.

Uscite di conversione dati
Queste uscite creano frammenti di codice sorgente che possono essere inseriti in programmi applicativi per convertire i dati da un formato all'altro. Le uscite di conversione dati sono descritte in Scrittura delle uscite di conversione dati.

L'uscita del workload del cluster
La funzione eseguita da questa uscita è definita dal provider dell'uscita. Le informazioni sulla definizione della chiamata vengono fornite in MQ_CLUSTER_WORKLOAD_EXIT - Descrizione chiamata.

Uscite API

Le uscite API consentono di scrivere il codice che modifica il funzionamento delle chiamate API IBM MQ , come MQPUT e MQGET, e quindi inserire tale codice immediatamente prima o subito dopo tali chiamate. L'inserimento è automatico; il gestore code guida il codice di uscita nei punti registrati. Per ulteriori informazioni sulle uscite API, vedi Utilizzo e scrittura delle uscite API.

Servizi installabili

I servizi installabili hanno interfacce formalizzate (un'API) con più punti di ingresso.

Un'implementazione di un servizio installabile è denominata *componente del servizio*. È possibile utilizzare i componenti forniti con IBM MQoppure è possibile scrivere il proprio componente per eseguire le funzioni richieste.

Attualmente, vengono forniti i seguenti servizi installabili:

Servizio di autorizzazione
Il servizio di autorizzazione consente di creare la propria funzione di protezione.

Il componente di servizio predefinito che implementa il servizio è OAM (object authority manager). Per impostazione predefinita, OAM è attivo e non è necessario eseguire alcuna operazione per configurarlo. È possibile utilizzare l'interfaccia del servizio di autorizzazione per creare altri componenti per sostituire o aumentare l'OAM. Per ulteriori informazioni su OAM, vedi Configurazione della sicurezza sui sistemi AIX, Linux, and Windows.

Servizio di denominazione
Il servizio dei nomi consente alle applicazioni di condividere le code identificando le code remote come se fossero code locali.

È possibile scrivere il proprio componente servizio nomi. Si potrebbe voler eseguire questa operazione se si intende utilizzare il servizio nomi con IBM MQ, ad esempio. Per utilizzare il servizio nomi, è necessario disporre di un componente scritto dall'utente o fornito da un altro fornitore di software. Per impostazione predefinita, il servizio nomi è inattivo.

Concetti correlati
Uscite utente, uscite API e servizi installabili IBM MQ

Interfacce di lingua IBM MQ Java

IBM MQ fornisce tre API (application programming interface) da utilizzare nelle applicazioni Java : IBM MQ classes for Jakarta Messaging, IBM MQ classes for JMSe IBM MQ classes for Java.

IBM supporta ed è un partecipante attivo in standard aperti.

- Da IBM MQ 8.0, il prodotto implementa lo standard JMS 2.0 , che ha introdotto una nuova API semplificata insieme a funzioni come le sottoscrizioni condivise.
- V 9.3.0 V 9.3.0 Da IBM MQ 9.3.0, è supportato anche Jakarta Messaging 3.0 .
- Inoltre, WebSphere Liberty supporta JMS 2.0 e Jakarta Messaging 3.0 con IBM MQ.

All'interno di IBM MQ ci sono tre API da usare nelle applicazioni Java :

V 9.3.0 JM 3.0 V 9.3.0 **IBM MQ classes for Jakarta Messaging**
IBM MQ classes for Jakarta Messaging è un provider Jakarta Messaging che implementa le interfacce Jakarta Messaging per IBM MQ come sistema di messaggistica. Jakarta Connectors Architecture fornisce una modalità standard di connessione delle applicazioni in esecuzione in un ambiente Jakarta EE a un EIS (Enterprise Information System) come IBM MQ o Db2.

JMS 2.0 **IBM MQ classes for JMS**
IBM MQ classes for JMS è un provider JMS che implementa le interfacce JMS per IBM MQ come sistema di messaggistica. Java Platform, Enterprise Edition Connector Architecture (JCA) fornisce un modo standard per collegare le applicazioni in esecuzione in un ambiente Java EE a un EIS (Enterprise Information System) come IBM MQ o Db2.

IBM MQ classes for Java
IBM MQ classes for Java consente di utilizzare IBM MQ in un ambiente Java . IBM MQ classes for Java consenti a un'applicazione Java di connettersi a IBM MQ come client IBM MQ o di connettersi direttamente a un gestore code IBM MQ .

Nota:

- V 9.3.0 V 9.3.0 JMS 2.0 è stato sostituito da Jakarta Messaging. IBM MQ classes for JMS continua a supportare lo standard JMS 2.0 , ma i futuri miglioramenti alla messaggistica Java emergeranno solo in Jakarta Messaging, quindi in IBM MQ classes for Jakarta Messaging. IBM MQ classes for JMS sono consigliati solo per la manutenzione e l'estensione di applicazioni JMS 2.0 esistenti. IBM MQ classes for Jakarta Messaging dovrebbe essere la tecnologia preferita per il nuovo sviluppo.
- Stabilized IBM MQ classes for Java sono funzionalmente stabilizzati al livello fornito in IBM MQ 8.0. Le applicazioni esistenti che utilizzano il IBM MQ classes for Java continueranno ad essere completamente supportate, ma questa API è stabilizzata, quindi le nuove funzioni non saranno aggiunte e le richieste di miglioramenti rifiutate. Completamente supportato significa che i difetti verranno corretti insieme a tutte le modifiche richieste dalle modifiche ai requisiti di sistema IBM MQ .

V 9.3.0 JM 3.0 V 9.3.0 Da IBM MQ 9.3, IBM MQ classes for Java, IBM MQ classes for JMSe IBM MQ classes for Jakarta Messaging vengono creati con Java 8. Gli ambienti di runtime Java a questi livelli o superiori devono essere utilizzati per eseguire le applicazioni utilizzando queste interfacce.

Concetti correlati
Accesso a IBM MQ da Java - Scelta dell'API
V 9.3.0 V 9.3.0 Perché utilizzare le classi IBM MQ per Jakarta Messaging?
Perché dovrei utilizzare IBM MQ classes for JMS?
Perché dovrei utilizzare IBM MQ classes for Java?

IBM MQ classes for JMS/Jakarta Messaging

IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging sono provider di messaggistica forniti con IBM MQ. Ciascuno di questi provider fornisce anche due serie di estensioni all'API di messaggistica. Le applicazioni Java Platform, Standard Edition (Java SE) e Java Platform, Enterprise Edition (Java EE) possono utilizzare questi fornitori di messaggistica.

V 9.3.0 JM 3.0 V 9.3.0 IBM MQ 9.3.0 introduce il supporto per Jakarta Messaging 3.0. JMS 2.0 è ancora completamente supportato.

Le specifiche JMS e Jakarta Messaging definiscono una serie di interfacce che le applicazioni possono utilizzare per eseguire operazioni di messaggistica. Da IBM MQ 8.0, il prodotto supporta la versione JMS 2.0 dello standard JMS . Questa implementazione offre tutte le funzioni dell'API classica ma richiede meno interfacce ed è più semplice da utilizzare. Per ulteriori informazioni, consultare “Modello JMS e Jakarta Messaging” a pagina 157 e la specifica JMS 2.0 all'indirizzo Java.net. JM 3.0 Da IBM MQ 9.3.0, è supportato anche Jakarta Messaging .

Il package jakarta.jms (Jakarta Messaging 3.0) o javax.jms (JMS 2.0) specifica i dettagli delle interfacce di messaggistica e un provider di messaggistica implementa tali interfacce per un prodotto di messaggistica specifico. Ad esempio:

- IBM MQ classes for JMS è un fornitore JMS che implementa le interfacce JMS per IBM MQ e fornisce anche le seguenti due serie di estensioni all'API JMS :
  - Estensioni IBM MQ JMS
  - Estensioni IBM JMS
- Un oggetto argomento, coda o factory di connessione creato utilizzando javax.dimso jakarta.jms, le interfacce o una serie di estensioni JMS possono essere indirizzate utilizzando una qualsiasi di queste API; ovvero, può essere assegnato a una qualsiasi delle interfacce. Per mantenere la portabilità dell'applicazione al livello più alto, utilizzare l'API più generica adatta ai propri requisiti.

Poiché JMS e Jakarta Messaging condividono molto in comune, ulteriori riferimenti a JMS in questo argomento possono essere considerati come riferimenti ad entrambi. Eventuali differenze vengono evidenziate come necessario.

Estensioni IBM MQ JMS

IBM MQ classes for JMS fornisce inoltre le estensioni all'API JMS . IBM MQ classes for JMS contiene le estensioni implementate negli oggetti MQConnectionFactory, MQQueue e MQTopic. Questi oggetti hanno proprietà e metodi specifici di IBM MQ. Gli oggetti possono essere gestiti oppure un'applicazione può creare gli oggetti in modo dinamico al runtime. Queste estensioni sono definite estensioni IBM MQ JMS . Notare che, in questa documentazione, gli oggetti creati dinamicamente da un'applicazione in fase di runtime non sono considerati oggetti gestiti.

Estensioni IBM JMS

Oltre alle estensioni IBM MQ JMS , IBM MQ classes for JMS fornisce una serie più generica di estensioni all'API JMS o Java come linguaggio di programmazione utilizzato. Queste estensioni sono definite estensioni IBM JMS e hanno i seguenti obiettivi generali:

- Per fornire un livello maggiore di coerenza tra i provider IBM JMS .
- Per semplificare la scrittura di un'applicazione bridge tra due sistemi di messaggistica IBM .
- Per rendere più semplice la porta di un'applicazione da un provider IBM JMS a un altro.

Il focus principale di queste estensioni riguarda la creazione e la configurazione di factory di connessione e destinazioni in modo dinamico al runtime, ma le estensioni forniscono anche funzioni che non sono direttamente correlate alla messaggistica, come la funzione per la determinazione dei problemi.

Attività correlate

Utilizzo delle classi IBM MQ per JMS/Jakarta Messaging

Configurazione delle risorse JMS e Jakarta Messaging

V 9.3.0 JM 3.0 V 9.3.0 IBM MQ classes for Jakarta Messaging: una panoramica

IBM MQ 9.3.0 introduce il supporto per Jakarta Messaging. Per Jakarta Messaging 3.0, il controllo della specifica JMS è stato spostato da Oracle al processo della comunità Java . Tuttavia, Oracle conserva il controllo del nome "javax", che viene utilizzato in altre tecnologie Java . Quindi, sebbene Jakarta Messaging 3.0 sia funzionalmente equivalente a JMS 2.0, ci sono alcune differenze nella denominazione.

Il nome ufficiale per la versione 3.0 non è Java Message Service, ma Jakarta Messaging e i nomi del pacchetto e della costante hanno come prefisso `jakarta` piuttosto che `javax`.

Sfondo

Per molti anni, la piattaforma Java è disponibile in due formati: Standard Edition e Enterprise Edition.

Java Platform, Standard Edition (a volte abbreviato come Java SE) è la lingua principale e le librerie di classe, in grado di essere eseguite in un contesto autonomo. La maggior parte dei package Java in Java SE ha nomi che iniziano con "java.".

Java Platform, Enterprise Edition (Java EE) estende questa funzione, aggiungendo funzionalità come messaggistica, vari bean, transazionalità e così via. Alcune di queste tecnologie possono essere utilizzate anche in un contesto Java SE . La maggior parte dei pacchetti Java in Java EE storicamente hanno nomi che iniziano con "javax." - c'è qualche crossover, tuttavia, alcuni pacchetti Java SE hanno "javax." come prefisso del nome.

Java Message Service (JMS) fa parte di Java Platform, Enterprise Edition. Java EE 7 integra JMS 2.0.

Fino a Java EE 7, le tecnologie erano sotto la direzione di Oracle.

Le tecnologie Java EE sono state recentemente trasferite dalla gestione di Oracle a un processo della community supervisionato da Eclipse Foundation.

Come il "javax". non è stato possibile spostare il nome nel nuovo progetto, è stato adottato un nuovo nome - tutti i package e i nomi delle proprietà hanno ora il prefisso "jakarta". e Java Platform, Enterprise Edition verrà chiamato "Jakarta EE" in futuro. La numerazione della versione è continuata: la versione 8 era una versione provvisoria che può essere in gran parte ignorata, e Jakarta EE 9 è il punto in cui il "jakarta". il prefisso ha effetto.

La tecnologia Jakarta EE principale che si applica nel contesto IBM MQ è Jakarta Messaging 3.0 - il successore di Java Message Service (JMS) 2.0. Quindi Jakarta EE 9 incorpora Jakarta Messaging 3.0.

IBM MQ continua a supportare Java EE 7 e JMS 2.0, introducendo il supporto per Jakarta EE 9 e Jakarta Messaging 3.0.

Cosa viene consegnato: Java SE

Per Java Platform, Standard Edition, oltre a IBM MQ classes for JMS (che supporta le operazioni JMS 2.0 con IBM MQ) IBM MQ 9.3.0 fornisce IBM MQ classes for Jakarta Messaging. Queste classi forniscono un provider Jakarta Messaging 3.0 che si integra con IBM MQ, consentendo l'uso dei gestori code IBM MQ per facilitare le operazioni Jakarta Messaging .

Tali file vengono forniti come file JAR standard, `com.ibm.mq.jakarta.client.jar`, nella directory secondaria `java/lib` dell'installazione IBM MQ .

Per l'utilizzo in contenitori OSGi, come Apache Felix o Eclipse Equinox, IBM MQ fornisce anche una coppia di bundle OSGi:

- `com.ibm.mq.osgi.jms30.clientprereqs_`*`V.R.M.F`*`.jar`
- `com.ibm.mq.osgi.jms30.client_`*`V.R.M.F`*`.jar`

dove *V.R.M.F* rappresenta la versione di IBM MQ, ad esempio 9.3.0.0. Questi bundle si trovano nella sottodirectory `java/lib/OSGi` dell'installazione IBM MQ .

Cosa viene consegnato: Jakarta EE 9

Per supportare la messaggistica basata su IBM MQin un server delle applicazioni compatibile con Jakarta EE 9 , IBM MQ fornisce un adattatore di risorse compatibile con Jakarta EE 9: `wmq.jakarta.jmsra.rar`. Si trova nella sottodirectory `java/lib/jca` dell'installazione di IBM MQ .

IBM MQ continua a fornire un adattatore di risorse compatibile con Java EE 7 , `wmq.jmsra.rar`, nella directory secondaria `java/lib/jca` dell'installazione di IBM MQ .

Modalità di consegna di queste risorse utente

Questi JAR e il file RAR per l'adattatore risorse sono forniti con le risorse preesistenti nel normale supporto di installazione di IBM MQ - sia il supporto di installazione specifico della piattaforma, come i file "`.rpm`", che il supporto ridistribuibile, come i file JAR del client ridistribuibili autoestraenti.

Cosa è cambiato tra JMS 2.0 e Jakarta Messaging 3.0

Jakarta EE 9 e Jakarta Messaging 3.0 non introducono nuove funzioni. Tutto ciò che cambia sono i nomi. Ad esempio, quando si utilizza "javax.jms.Connection" in JMS 2.0, si utilizza "jakarta.jms.Connection" in Jakarta Messaging 3.0.

Poiché Eclipse Foundation porta avanti la piattaforma Jakarta EE , si baserà su questa base e questa convenzione di denominazione verrà utilizzata per le nuove funzionalità introdotte in futuro.

Cosa è cambiato tra IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging

Riepilogo

IBM MQ classes for JMS, che fornisce supporto per JMS 2.0, rimane disponibile e si consiglia principalmente di gestire ed estendere le applicazioni esistenti. Essi sono pienamente supportati.

IBM MQ classes for Jakarta Messaging, che fornisce supporto per Jakarta Messaging 3.0, è consigliato per il nuovo sviluppo.

In IBM MQ 9.3.0, queste due offerte sono funzionalmente equivalenti. Solo la denominazione è diversa. Tuttavia, è più probabile che la nuova funzionalità di messaggistica emerga in IBM MQ classes for Jakarta Messaging che in IBM MQ classes for JMS.

Le due offerte sono interoperabili. I messaggi prodotti da IBM MQ classes for JMS possono essere utilizzati da IBM MQ classes for Jakarta Messaginge viceversa. Ma le due offerte non devono coesistere in un'unica applicazione.

Modifiche di denominazione

Tabella 16. Modifiche ai nomi dei package

IBM MQ classes for JMS nome pacchetto	**IBM MQ classes for Jakarta Messaging nome pacchetto**
com.ibm.mq.jms[.*]	com.ibm.mq.jakarta.jms[.*]
com.ibm.jms	com.ibm.jakarta.jms
com.ibm.msg.client.jms.*	com.ibm.msg.client.jakarta.jms.*
com.ibm.msg.client.wmq.*	com.ibm.msg.client.jakarta.wmq.*

I pacchetti relativi ai servizi comuni (traccia, registrazione, supporto lingua nazionale ecc.) e le implementazioni JMQI (locale e remota) sono comuni sia a IBM MQ classes for JMS che a IBM MQ classes for Jakarta Messaging, pertanto non sono necessarie modifiche in queste aree.

Si noti che anche i nomi delle proprietà sono stati modificati. Ad esempio, la proprietà per abilitare le estensioni IBM MQ in IBM MQ classes for Jakarta Messaging è **`com.ibm.mq.jakarta.jms.SupportMQExtensions`**.

I nomi delle proprietà che sono indipendenti da IBM MQ classes for JMS o IBM MQ classes for Jakarta Messaging, come le diverse proprietà **`com.ibm.msg.client.commonservices.trace.*`** , si applicano ugualmente a entrambe le offerte.

Programmi di utilità amministrativi

I programmi di utilità **`crtmqenv`** e **`setmqenv`** accettano ora un'opzione per specificare se il percorso classi deve essere configurato per IBM MQ classes for JMS (`-j 2.0`) o IBM MQ classes for Jakarta

Messaging (`-j 3.0`) e vi sono IBM MQ classes for Jakarta Messaging varianti dei programmi di utilità **runjms** , denominati **runjms30** e nomi simili.

Il programma di utilità **dspmqver** , quando viene richiesto di creare un report sui componenti Java , include IBM MQ classes for Jakarta Messaging nel relativo output.

Per configurare gli oggetti IBM MQ classes for Jakarta Messaging da recuperare tramite JNDI, il nuovo programma di utilità **JMS30Admin** è equivalente al programma di utilità **JMSAdmin** per IBM MQ classes for JMS.

Notare che gli oggetti sottostanti provengono da pacchetti differenti. le definizioni JNDI create da **JMSAdmin** non possono essere utilizzate da IBM MQ classes for Jakarta Messaging, né quelle create da **JMS30Admin** possono essere utilizzate da IBM MQ classes for JMS.

Nota: Non vi è alcun supporto per gli oggetti IBM MQ classes for Jakarta Messaging forniti da IBM MQ Explorer; la sua integrazione JNDI è solo per IBM MQ classes for JMS .

Concetti correlati

Perché utilizzare le classi IBM MQ per Jakarta Messaging?

Modello JMS e Jakarta Messaging

Il modello JMS e Jakarta Messaging definisce una serie di interfacce che le applicazioni Java possono utilizzare per eseguire operazioni di messaggistica. IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging sono provider di messaggistica che definiscono il modo in cui gli oggetti di messaggistica Java sono correlati ai concetti IBM MQ . Le specifiche JMS e Jakarta Messaging prevedono che alcuni oggetti di messaggistica siano oggetti gestiti.

Da IBM MQ 8.0, il prodotto supporta la versione JMS 2.0 dello standard JMS, che ha introdotto un'API semplificata, conservando anche l'API classica, da JMS 1.1.

V 9.3.0 JM 3.0 V 9.3.0 IBM MQ 9.3.0 introduce il supporto per Jakarta Messaging 3.0. JMS 2.0 è ancora completamente supportato. Poiché JMS e Jakarta Messaging condividono molto in comune, ulteriori riferimenti a JMS in questo argomento possono essere considerati come riferimenti ad entrambi. Eventuali differenze vengono evidenziate come necessario.

API semplificata

JMS 2.0 ha introdotto l'API semplificata, mantenendo anche le interfacce specifiche del dominio e indipendenti dal dominio da JMS 1.1. L'API semplificata riduce il numero di oggetti necessari per inviare e ricevere messaggi e consiste nelle seguenti interfacce:

ConnectionFactory
Un ConnectionFactory è un oggetto gestito utilizzato da un client JMS per creare una connessione. Questa interfaccia viene utilizzata anche nell'API classica.

JMSContesto
Questo oggetto combina gli oggetti Connection e Session dell'API classica. JMSGli oggetti di contesto possono essere creati da altri JMSoggetti di contesto, con la connessione sottostante duplicata.

JMSMittente
Un Producer JMSviene creato da un contesto JMSe viene utilizzato per inviare messaggi a una coda o a un argomento. L'oggetto Producer JMSprovoca la creazione degli oggetti richiesti per inviare il messaggio.

JMSDestinatario
Un consumer JMSviene creato da un contesto JMSe viene utilizzato per ricevere messaggi da un argomento o da una coda.

L'API semplificata ha una serie di effetti:

- L'oggetto Context JMSavvia sempre automaticamente la connessione sottostante.
- JMSI produttori e JMSi consumatori possono ora lavorare direttamente con il corpo del messaggio, senza dover ottenere l'intero oggetto del messaggio, utilizzando il metodo getBody del messaggio.

- Le proprietà dei messaggi possono essere impostati sull'oggetto JMSProducer, utilizzando il concatenamento di metodo, prima di inviare un 'corpo ', un contenuto di messaggi. Il Producer JMSgestirà la creazione di tutti gli oggetti necessari per inviare il messaggio. Utilizzando JMS 2.0, è possibile impostare le proprietà e inviare un messaggio nel modo seguente:

```
context.createProducer().
setProperty("foo", "bar").
setTimeToLive(10000).
setDeliveryMode(NON_PERSISTENT).
setDisableMessageTimestamp(true).
send(dataQueue, body);
```

JMS 2.0 ha anche introdotto sottoscrizioni condivise in cui i messaggi possono essere condivisi tra più consumer. Tutte le sottoscrizioni JMS 1.1 vengono considerate sottoscrizioni non condivise.

API classica

Il seguente elenco riepiloga le principali interfacce JMS dell'API classica:

Destinazione
Una destinazione è il punto in cui un'applicazione invia i messaggi o è un'origine da cui un'applicazione riceve i messaggi o entrambi.

ConnectionFactory
Un oggetto ConnectionFactory contiene una serie di proprietà di configurazione per una connessione. Un'applicazione utilizza una produzione connessioni per creare una connessione.

Connessione
Un oggetto Connection incapsula una connessione attiva dell'applicazione a un server di messaggistica. Un'applicazione utilizza una connessione per creare sessioni.

Sessione
Una sessione è un contesto a thread singolo per inviare e ricevere messaggi. Un'applicazione utilizza una sessione per creare messaggi, produttori di messaggi e utenti di messaggi. Una sessione è sottoposta a transazione o non è stata sottoposta a transazione.

Messaggio
Un oggetto Message incapsula un messaggio che un'applicazione invia o riceve.

MessageProducer
Un'applicazione utilizza un produttore di messaggi per inviare messaggi a una destinazione.

MessageConsumer
Un'applicazione utilizza un utente di messaggi per ricevere i messaggi inviati a una destinazione.

Figura 53 a pagina 158 mostra questi oggetti e le relazioni.

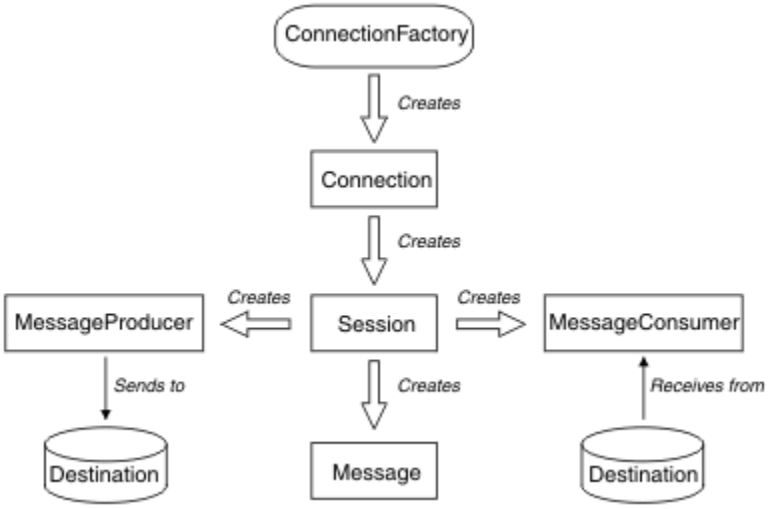


Figura 53. Oggetti JMS e relative relazioni

Il diagramma mostra le interfacce principali: ConnectionFactory, Connection, Session, MessageProducer, MessageConsumer, Message e Destination. Un'applicazione utilizza una produzione connessioni per creare una connessione e una connessione per creare sessioni. L'applicazione può quindi utilizzare una sessione per creare messaggi, produttori di messaggi e utenti di messaggi. L'applicazione utilizza un producer di messaggi per inviare messaggi a una destinazione e utilizza un consumer di messaggi per ricevere i messaggi inviati a una destinazione.

Un oggetto di destinazione, ConnectionFactoryo Connection può essere utilizzato contemporaneamente da thread differenti di un'applicazione a più thread, ma un oggetto Session, MessageProducero MessageConsumer non può essere utilizzato contemporaneamente da thread differenti. Il modo più semplice per garantire che un oggetto Session, MessageProducero MessageConsumer non venga utilizzato contemporaneamente consiste nel creare un oggetto Session separato per ciascun thread.

JMS supporta due stili di messaggistica:

- Messaggistica point-to-point
- Pubblicazione/sottoscrizione della messaggistica

Questi stili di messaggistica vengono anche indicati come *domini di messaggistica*ed è possibile combinare entrambi gli stili di messaggistica in un'applicazione. Nel dominio point-to-point, una destinazione è una coda e, nel dominio di pubblicazione / sottoscrizione, una destinazione è un argomento.

Con le versioni di JMS precedenti a JMS 1.1, la programmazione per il dominio point-to-point utilizza una serie di interfacce e metodi, mentre la programmazione per il dominio di pubblicazione / sottoscrizione utilizza un'altra serie. I due set sono simili, ma separati. Da JMS 1.1, è possibile utilizzare una serie comune di interfacce e metodi che supporta entrambi i domini di messaggistica. Le interfacce comuni forniscono una vista indipendente dal dominio di ciascun dominio di messaggistica. Tabella 17 a pagina 159 elenca le interfacce indipendenti del dominio JMS e le relative interfacce specifiche del dominio corrispondenti.

Tabella 17. Il dominio JMS indipendente e le relative interfacce specifiche del dominio corrispondenti

Interfacce indipendenti dal dominio	**Interfacce specifiche del dominio per il dominio point - to - point**	**Interfacce specifiche del dominio per il dominio di pubblicazione / sottoscrizione**
ConnectionFactory	Factory QueueConnection	Factory TopicConnection
Connessione	QueueConnection	TopicConnection
Destinazione	Coda	Argomento
Sessione	QueueSession	TopicSession
MessageProducer	QueueSender	TopicPublisher
MessageConsumer	QueueReceiver QueueBrowser	TopicSubscriber

JMS 2.0 IBM MQ classes for JMS 2.0 supporta sia le precedenti interfacce specifiche del dominio JMS 1.1 che l'API semplificata di JMS 2.0. IBM MQ classes for JMS 2.0 può quindi essere utilizzato per gestire le applicazioni esistenti, incluso lo sviluppo di nuove funzioni in applicazioni esistenti.

JM 3.0 IBM MQ classes for Jakarta Messaging 3.0 supporta le versioni Jakarta Messaging delle stesse interfacce ed è consigliato per lo sviluppo di nuove applicazioni.

In IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging, gli oggetti JMS sono correlati a concetti IBM MQ nei seguenti modi:

- Un oggetto Connection ha proprietà derivate dalle proprietà del factory di connessione utilizzato per creare la connessione. Queste proprietà controllano il modo in cui un'applicazione si connette a un gestore code. Esempi di queste proprietà sono il nome del gestore code e, per un'applicazione che

si connette al gestore code in modalità client, il nome host o l'indirizzo IP del sistema su cui è in esecuzione il gestore code.

- Un oggetto Session incapsula un handle di connessione IBM MQ , che definisce quindi l'ambito transazionale della sessione.
- Un oggetto MessageProducer e un oggetto MessageConsumer incapsulano un handle di oggetto IBM MQ .

Quando si utilizza IBM MQ classes for JMS o IBM MQ classes for Jakarta Messaging, vengono applicate tutte le normali regole di IBM MQ . Notare, in particolare, che un'applicazione può inviare un messaggio a una coda remota, ma può ricevere un messaggio solo da una coda di proprietà del gestore code a cui è connessa l'applicazione.

La specifica JMS prevede che gli oggetti ConnectionFactory e Destination siano oggetti gestiti. Un amministratore crea e gestisce gli oggetti gestiti in un repository centrale e un'applicazione JMS richiama tali oggetti utilizzando JNDI (Java Naming and Directory Interface).

In IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging, l'implementazione dell'interfaccia Destinazione è una superclasse astratta di Coda e Argomento, e quindi un'istanza di Destinazione è un oggetto Coda o un oggetto Argomento. Le interfacce indipendenti dal dominio trattano una coda o un argomento come una destinazione. Il dominio di messaggistica per un oggetto MessageProducer o MessageConsumer è determinato se la destinazione è una coda o un argomento.

In IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging , quindi, gli oggetti dei seguenti tipi possono essere oggetti gestiti:

- ConnectionFactory
- Factory QueueConnection
- Factory TopicConnection
- Coda
- Argomento
- XAConnectionFactory
- Factory XAQueueConnection
- Factory XATopicConnection

Architettura IBM MQ classes for JMS/Jakarta Messaging

IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging hanno un'architettura a livelli. Il livello di codice più alto è un livello comune che può essere utilizzato da qualsiasi provider di messaggistica IBM Java .

V 9.3.0 JM 3.0 V 9.3.0 IBM MQ 9.3.0 introduce il supporto per Jakarta Messaging 3.0. JMS 2.0 è ancora completamente supportato.

IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging hanno un'architettura a livelli come mostrato nel diagramma Figura 54 a pagina 161. Il livello più alto di codice è un livello comune che può essere utilizzato da qualsiasi provider IBM JMS o Jakarta Messaging. Quando un'applicazione chiama un metodo JMS o Jakarta Messaging, qualsiasi elaborazione della chiamata che non è specifica di un sistema di messaggistica viene eseguita dal livello comune, che fornisce anche una risposta congruente alla chiamata. Qualsiasi elaborazione della chiamata specifica di un sistema di messaggistica viene delegata ad un livello inferiore. Nel seguente diagramma, il provider di messaggistica IBM MQ viene visualizzato nel livello inferiore, insieme a due ulteriori provider di messaggistica (provider di messaggistica A e provider di messaggistica B.)

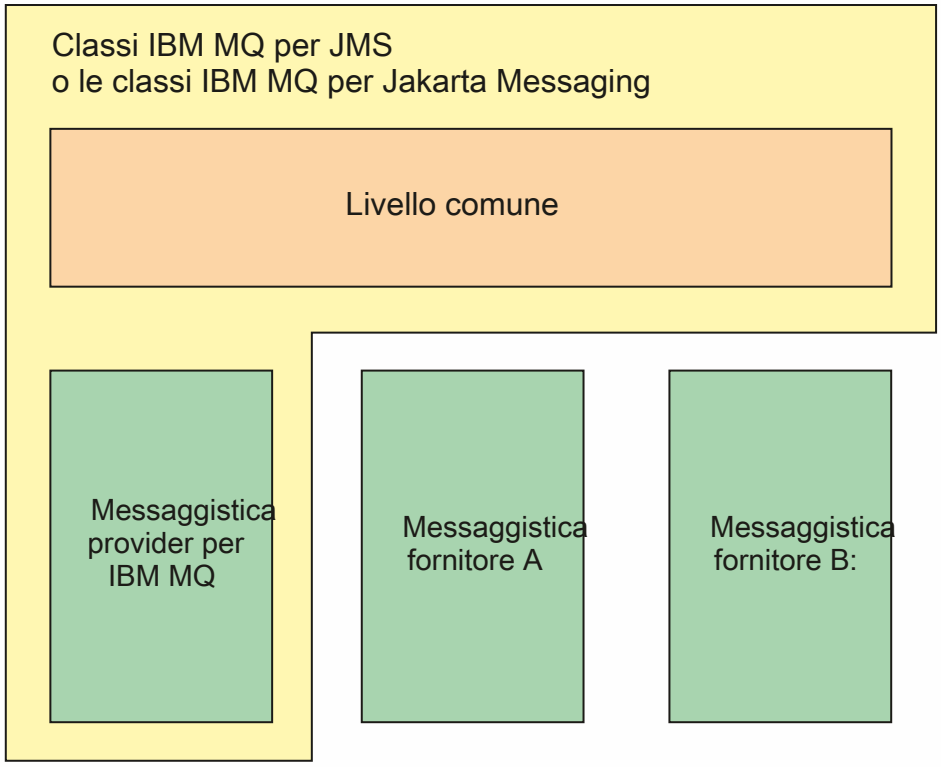


Figura 54. L'architettura a livelli per provider IBM JMS e Jakarta Messaging

Un'architettura a livelli soddisfa i seguenti obiettivi:

- Per migliorare la coerenza del comportamento dei diversi provider IBM JMS e Jakarta Messaging
- Per semplificare la scrittura di un'applicazione bridge tra due sistemi di messaggistica IBM
- Per semplificare la porta di un'applicazione da un provider IBM JMS o Jakarta Messaging a un altro

Attività correlate
Utilizzo delle classi IBM MQ per JMS/Jakarta Messaging

Supporto per gli oggetti gestiti

IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging supportano l'utilizzo di oggetti gestiti.

V 9.3.0 JM 3.0 V 9.3.0 Da IBM MQ 9.3.0, Jakarta Messaging 3.0 è supportato per lo sviluppo di nuove applicazioni. IBM MQ 9.3.0 continua a supportare JMS 2.0 per applicazioni esistenti. Non è supportato utilizzare sia l'API Jakarta Messaging 3.0 che l'API JMS 2.0 nella stessa applicazione. Per ulteriori informazioni, consultare Utilizzo delle classi IBM MQ per JMS/Jakarta Messaging.

Il flusso di logica all'interno di un'applicazione JMS o IBM MQ classes for Jakarta Messaging inizia con gli oggetti ConnectionFactory e Destination. L'applicazione utilizza un oggetto ConnectionFactory per creare un oggetto Connection, che rappresenta la connessione attiva dall'applicazione a un server di messaggistica. L'applicazione utilizza l'oggetto Connection per creare un oggetto Session, un contesto a thread singolo per la produzione e l'utilizzo di messaggi. L'applicazione può quindi utilizzare l'oggetto Session e un oggetto Destination per creare un oggetto MessageProducer , che l'applicazione utilizza per inviare messaggi alla destinazione specificata. La destinazione è una coda o un argomento nel sistema di messaggistica ed è incapsulata dall'oggetto Destinazione. L'applicazione può anche utilizzare l'oggetto Session e un oggetto Destination per creare un oggetto MessageConsumer , che l'applicazione utilizza per ricevere i messaggi inviati alla destinazione specificata.

Le specifiche JMS e Jakarta Messaging prevedono che gli oggetti gestiti siano ConnectionFactory e Destination. Un amministratore crea e gestisce oggetti gestiti in un repository centrale e un'applicazione

JMS o Jakarta Messaging richiama tali oggetti utilizzando Java Naming Directory Interface (JNDI). Il repository di oggetti gestiti può variare da un file semplice a una directory LDAP (Lightweight Directory Access Protocol).

IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging supportano l'utilizzo di oggetti gestiti. Un'applicazione può utilizzare tutte le funzioni di IBM MQ classes for JMS o IBM MQ classes for Jakarta Messaging esposte tramite IBM MQ senza avere alcuna informazione specifica di IBM MQcodificata nell'applicazione stessa. Questa disposizione fornisce all'applicazione un grado di indipendenza dalla configurazione IBM MQ sottostante.

Per ottenere questa indipendenza, l'applicazione può utilizzare JNDI per richiamare factory di connessione e destinazioni memorizzate come oggetti gestiti e utilizzare solo le interfacce definite nel pacchetto `javax.jms` (JMS 2.0) o `jakarta.jms` (Jakarta Messaging 3.0) per eseguire operazioni di messaggistica.

JMS 2.0 Per JMS 2.0, un amministratore può utilizzare lo strumento di amministrazione IBM MQ JMS **JMSAdmin**o IBM MQ Explorer, per creare e gestire gli oggetti gestiti in un repository centrale.

JM 3.0 Per Jakarta Messaging 3.0, non è possibile gestire JNDI utilizzando IBM MQ Explorer. La gestione JNDI è supportata dalla variante Jakarta Messaging 3.0 di **JMSAdmin**, che è **JMS30Admin**.

Un server delle applicazioni generalmente fornisce il proprio repository per gli oggetti gestiti e i propri strumenti per la creazione e la gestione degli oggetti. Un'applicazione Java EE JM 3.0 o Jakarta EE può quindi utilizzare JNDI per richiamare gli oggetti gestiti dal repository del server delle applicazioni o da un repository centrale.

Attività correlate

Configurazione delle risorse JMS e Jakarta Messaging

Tipi di comunicazione supportati su piattaforme Java EE e Jakarta EE

Sulle piatteforme Java EE e Jakarta EE , IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging supportano due tipi di comunicazione tra un componente di un'applicazione e un gestore code IBM MQ .

V 9.3.0 JM 3.0 V 9.3.0 IBM MQ 9.3.0 introduce il supporto per Jakarta Messaging 3.0. JMS 2.0 è ancora completamente supportato. Poiché JMS e Jakarta Messaging condividono molto in comune, ulteriori riferimenti a JMS in questo argomento possono essere considerati come riferimenti ad entrambi. Eventuali differenze vengono evidenziate come necessario.

Sono supportati i seguenti due tipi di comunicazione tra un componente di un'applicazione e un gestore code IBM MQ :

- Comunicazione in uscita
- Comunicazione in ingresso

Comunicazione in uscita

Utilizzando direttamente l'API JMS o Jakarta Messaging , un componente dell'applicazione crea una connessione a un gestore code, quindi invia e riceve messaggi.

Ad esempio, il componente dell'applicazione pu essere un client dell'applicazione, un servlet, un JSP (Java Server Page), un EJB (enterprise Java bean) o un MDB (message driven bean). In questo tipo di comunicazione, il contenitore del server delle applicazioni fornisce solo funzioni di basso livello a supporto delle operazioni di messaggistica, come il pool di connessioni e la gestione thread.

Comunicazione in ingresso

Nel caso di una comunicazione in entrata, un messaggio che arriva a una destinazione viene consegnato a un MDB, che quindi elabora il messaggio.

Le applicazioni Java EE JM 3.0 e Jakarta EE utilizzano MDB per elaborare i messaggi in modo asincrono. Un MDB funge da listener di messaggi JMS e viene implementato da un metodo onMessage(), che definisce il modo in cui un messaggio viene elaborato. Un MDB viene distribuito nel contenitore EJB di un application server. Il modo preciso in cui un MDB è configurato dipende dal server delle applicazioni che si sta utilizzando, ma le informazioni di configurazione devono specificare a quali gestori code connettersi, come connettersi al gestore code, quale destinazione monitorare per i messaggi e il comportamento transazionale di MDB. Queste informazioni vengono quindi utilizzate dal contenitore EJB. Quando un messaggio che soddisfa i criteri di selezione dell'MDB arriva alla destinazione specificata, il contenitore EJB utilizza IBM MQ classes for JMS o IBM MQ classes for Jakarta Messaging per richiamare il messaggio dal gestore code e quindi consegna il messaggio all'MDB richiamandone il metodo onMessage().

Relazione con IBM MQ classes for Java

IBM MQ classes for Java, IBM MQ classes for Jakarta Messaging e IBM MQ classes for JMS sono peer che utilizzano un'interfaccia Java comune per MQI.

Figura 55 a pagina 163 mostra la relazione tra IBM MQ classes for JMS, IBM MQ classes for Jakarta Messaging e IBM MQ classes for Java.

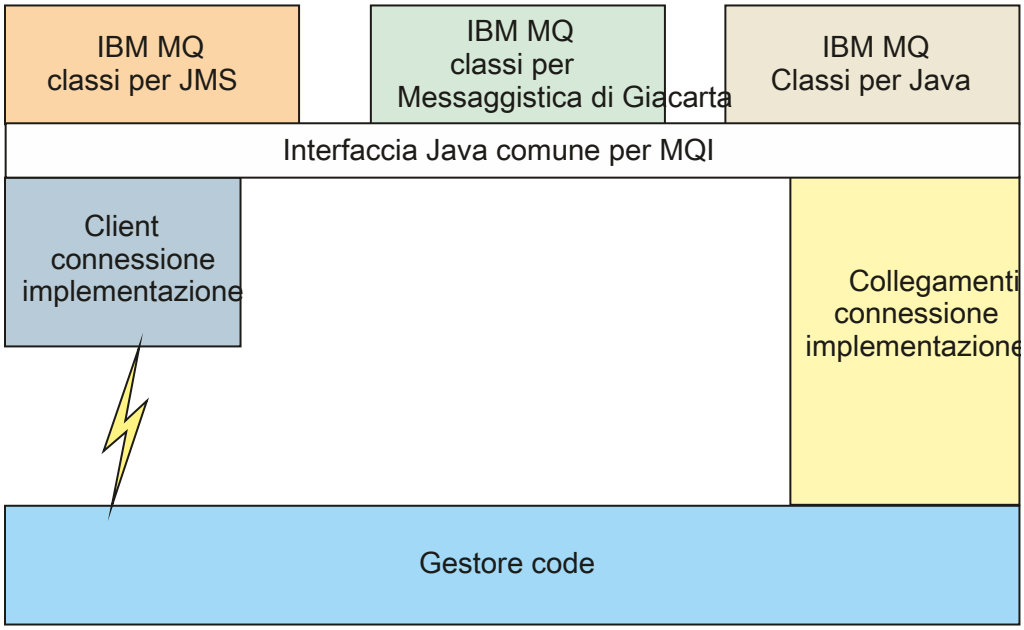


Figura 55. La relazione tra IBM MQ classes for JMS, IBM MQ classes for Jakarta Messaging e IBM MQ classes for Java

In generale, i programmi Java devono utilizzare una sola interfaccia per interfacciarsi con IBM MQ - IBM MQ classes for Java, IBM MQ classes for Jakarta Messaging o IBM MQ classes for JMS. La combinazione di interfacce non è supportata, con un'eccezione. Per mantenere la compatibilità con le release precedenti a IBM WebSphere MQ 7.0, le classi di uscita del canale scritte in Java possono ancora utilizzare le interfacce IBM MQ classes for Java , anche se le classi di uscita del canale vengono richiamate da IBM MQ classes for JMS. Tuttavia, l'utilizzo delle interfacce IBM MQ classes for Java significa che le applicazioni dipendono ancora da:

- JMS 2.0 Il file JAR IBM MQ classes for Java , `com.ibm.mq.jar`. Se non si desidera `com.ibm.mq.jar` nel percorso classe, è possibile utilizzare la serie di interfacce nel pacchetto `com.ibm.mq.exits` .
- V 9.3.0 JM 3.0 V 9.3.0 Utilizzo di `com.ibm.mq.jakarta.client.jar`, quando si interagisce con IBM MQ classes for Jakarta Messaging.

Concetti correlati

V 9.3.0 V 9.3.0 Perché utilizzare le classi IBM MQ per Jakarta Messaging?

Perché utilizzare le classi IBM MQ per JMS?

Perché utilizzare le classi IBM MQ per Java?

IBM MQ Provider di messaggistica

Il provider di messaggistica IBM MQ ha tre modalità operative: modalità normale, modalità normale con limitazioni e modalità di migrazione.

Il provider di messaggistica IBM MQ ha tre modi di funzionamento:

- Modalità normale del provider di messaggistica IBM MQ
- Modalità normale del provider di messaggistica IBM MQ con restrizioni
- Modalità di migrazione del provider di messaggistica IBM MQ

La modalità normale del provider di messaggistica IBM MQ utilizza tutte le funzioni di un gestore code IBM MQ per implementare JMS. Questa modalità è ottimizzata per utilizzare l'API e la funzionalità di JMS 2.0 V 9.3.0 JM 3.0 V 9.3.0 o Jakarta Messaging 3.0 .

If:

- Il client specifica una versione del provider 6 su un **ConnectionFactory**, il client si comporta in modo compatibile con il client fornito con IBM WebSphere MQ 6.0. Sono supportate solo le interfacce JMS 1.1 e JMS 2, ma alcune funzionalità JMS 2, come le sottoscrizioni condivise, il ritardo di consegna e l'invio asincrono, sono disabilitate. Non esiste alcuna connessione condivisa.
- Il client specifica una versione del provider 7 su un **ConnectionFactory**, entrambe le interfacce JMS 1.1 e JMS 2 sono completamente supportate.
- Non è stata specificata alcuna versione del provider, viene effettuato un tentativo di connessione con il provider versione 7. Se ciò non riesce, viene effettuato un ulteriore tentativo con il fornitore versione 6.

Se si desidera connettersi a IBM Integration Bus utilizzando IBM MQ Enterprise Transport, utilizzare la modalità di migrazione. Se si utilizza IBM MQ Real-Time Transport, la modalità di migrazione viene selezionata automaticamente perché sono state selezionate esplicitamente le proprietà nell'oggetto factory di connessione. La connessione a IBM Integration Bus utilizzando IBM MQ Enterprise Transport segue le regole generali per selezionare la modalità descritte in Configurazione della proprietà JMS **PROVIDERVERSION**.

Attività correlate

Configurazione delle risorse JMS

z/OS Concetti di IBM MQ for z/OS

Alcuni dei concetti utilizzati da IBM MQ for z/OS sono univoci per la piattaforma z/OS . Ad esempio, il meccanismo di registrazione, le tecniche di gestione della memoria, la disposizione dell'unità di ripristino e i gruppi di condivisione code vengono forniti solo con IBM MQ for z/OS. Utilizzare questo argomento come introduzione per ulteriori informazioni su questi concetti.

I concetti includono una panoramica degli oggetti utilizzati da IBM MQ for z/OS, inclusi:

- Il gestore code
- Il programma di avvio dei canali
- Code condivise e gruppi di condivisione code
- Accodamento all'interno del gruppo

I seguenti argomenti riguardano anche varie procedure necessarie, tra cui:

- Definizioni di sistema su z/OS
- Gestione dello storage

- Ripristino e riavvio
- Concetti di sicurezza in IBM MQ for z/OS

Concetti correlati

“Il gestore code su z/OS” a pagina 166
Prima di consentire ai propri programmi applicativi di utilizzare IBM MQ sul proprio sistema z/OS , è necessario installare il prodotto IBM MQ for z/OS e avviare un gestore code. Il gestore code possiede e gestisce la serie di risorse utilizzate da IBM MQ.

“L'iniziatore di canali su z/OS” a pagina 167
L'iniziatore di canali fornisce e gestisce le risorse che abilitano l'accodamento distribuito IBM MQ . IBM MQ utilizza *Message Channel Agents* (MCA) per inviare messaggi da un gestore code a un altro.

“Termini e attività per la gestione di IBM MQ for z/OS” a pagina 168
Utilizzare questo argomento come introduzione alla terminologia e alle attività specifiche di IBM MQ for z/OS.

“Code condivise e gruppi di condivisione code” a pagina 171
È possibile utilizzare code condivise e gruppi di condivisione code per implementare l'alta disponibilità delle risorse IBM MQ . Le code condivise e i gruppi di condivisione code sono funzioni univoche per IBM MQ for z/OS sulla piattaforma z/OS .

“Accodamento all'interno del gruppo” a pagina 217
Questa sezione descrive l'accodamento all'interno del gruppo, una funzione IBM MQ for z/OS univoca per la piattaforma z/OS . Questa funzione è disponibile solo per i gestori code definiti per un gruppo di condivisione code.

“Gestione della memoria su z/OS” a pagina 231
IBM MQ for z/OS richiede strutture di dati permanenti e temporanee e utilizza serie di pagine e buffer di memoria per memorizzare questi dati. Questi argomenti forniscono ulteriori dettagli su come IBM MQ utilizza queste serie di pagine e buffer.

“accessoIBM MQ for z/OS” a pagina 236
IBM MQ conserva i *log* delle modifiche dei dati e degli eventi significativi quando si verificano. Questi log possono essere utilizzati per ripristinare i dati ad uno stato precedente, se necessario.

“Ripristino e riavvio su z/OS” a pagina 258
Utilizzare i link in questo argomento per informazioni sulle funzioni di IBM MQ for z/OS per il riavvio e il recupero.

“Concetti di sicurezza in IBM MQ for z/OS” a pagina 275
Utilizzare questo argomento per comprendere l'importanza della sicurezza per IBM MQe le implicazioni di non disporre di impostazioni di sicurezza adeguate sul sistema.

“Disponibilità su z/OS” a pagina 282
IBM MQ for z/OS ha molte funzioni per l'alta disponibilità. Questo argomento descrive alcune delle considerazioni sulla disponibilità.

“Disposizione dell'unità di recupero su z/OS” a pagina 286
Alcune applicazioni transazionali possono utilizzare un'unità di disposizione di ripristino GROUP, anziché QMGR, quando sono connesse a un gestore code in un gruppo di condivisione code (QSG), specificando il nome QSG quando si connettono invece del nome del gestore code. Ciò consente al recupero delle transazioni di essere più flessibile e robusto, rimuovendo il requisito di riconnettersi allo stesso gestore code in QSG.

Riferimenti correlati

“Definizione di sistema su z/OS” a pagina 247
IBM MQ for z/OS utilizza molte definizioni di oggetti predefinite e fornisce un JCL di esempio per creare tali oggetti predefiniti. Utilizzare questo argomento per comprendere questi oggetti predefiniti e il JCL di esempio.

“Monitoraggio e statistiche su IBM MQ for z/OS” a pagina 285
IBM MQ for z/OS dispone di una serie di funzionalità per il controllo del gestore code e la raccolta di statistiche.

z/OS Il gestore code su z/OS

Prima di consentire ai propri programmi applicativi di utilizzare IBM MQ sul proprio sistema z/OS , è necessario installare il prodotto IBM MQ for z/OS e avviare un gestore code. Il gestore code possiede e gestisce la serie di risorse utilizzate da IBM MQ.

Il gestore code

Un *gestore code* è un programma che fornisce servizi di messaggistica alle applicazioni. Le applicazioni che utilizzano l'interfaccia MQI (Message Queue Interface) possono inserire e richiamare i messaggi dalle code. Il gestore code garantisce che i messaggi vengano inviati alla coda appropriata oppure indirizzati verso un altro gestore code. Il gestore code elabora entrambe le chiamate MQI che vengono emesse e i comandi inoltrati (da qualsiasi origine). Il gestore code genera i codici di completamento appropriati per ogni chiamata o comando.

Le risorse gestite dal gestore code includono:

- Serie di pagine che contengono le definizioni dell'oggetto IBM MQ e i dati del messaggio
- Log utilizzati per ripristinare i messaggi e gli oggetti in caso di errore del gestore code
- Memoria processore
- Connessioni tramite cui diversi ambienti dell'applicazione (CICS, IMSe Batch) possono accedere all'API IBM MQ
- L'iniziatore di canali IBM MQ , che consente la comunicazione tra IBM MQ sul sistema z/OS e altri sistemi

Il gestore code ha un nome e le applicazioni possono connettersi utilizzando questo nome.

Figura 56 a pagina 166 illustra un gestore code, mostrando le connessioni a diversi ambienti di applicazione e l'iniziatore di canali.

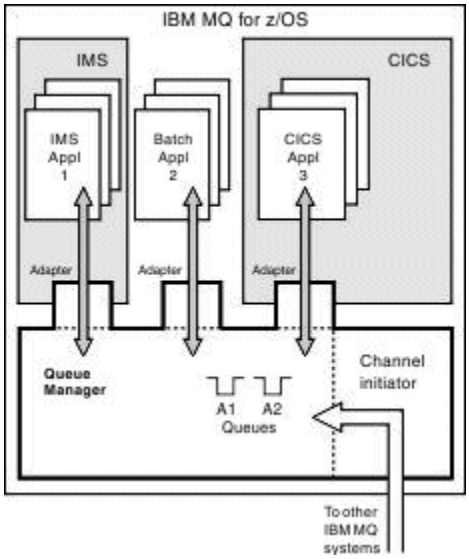


Figura 56. Panoramica di IBM MQ for z/OS

Il sottosistema del gestore code su z/OS

Su z/OS, IBM MQ viene eseguito come un sottosistema z/OS avviato al momento dell'IPL. Nel sottosistema, il gestore code viene avviato eseguendo una procedura JCL che specifica i dataset z/OS che contengono informazioni sui log e che contengono definizioni di oggetti e dati di messaggi (le serie di pagine). Il sottosistema e gestore code hanno lo stesso nome, con un massimo di quattro caratteri. Tutti

i gestori code nella rete devono avere nomi univoci, anche se si trovano su sistemi, sysplex o piattaforme differenti.

z/OS L'iniziatore di canali su z/OS

L'iniziatore di canali fornisce e gestisce le risorse che abilitano l'accodamento distribuito IBM MQ . IBM MQ utilizza *Message Channel Agents* (MCA) per inviare messaggi da un gestore code a un altro.

Per inviare messaggi dal gestore code A al gestore code B, un *mittente* MCA sul gestore code A deve configurare un link di comunicazioni al gestore code B. Un MCA di *ricezione* deve essere avviato sul gestore code B per ricevere messaggi dal collegamento di comunicazioni. Questo percorso unidirezionale composto dall'MCA mittente, dal collegamento di comunicazioni e dall'MCA ricevente è noto come *canale*. L'MCA mittente prende i messaggi da una coda di trasmissione e li invia attraverso un canale all'MCA ricevente. L'MCA ricevente riceve i messaggi e li inserisce nelle code di destinazione.

In IBM MQ for z/OS, gli MCA di invio e ricezione vengono tutti eseguiti all'interno dell'iniziatore di canali (l'iniziatore di canali è noto anche come *mover*). L'iniziatore di canali viene eseguito come uno spazio di indirizzo z/OS sotto il controllo del gestore code. Ci può essere solo un singolo iniziatore di canali connesso a un gestore code e viene eseguito nella stessa immagine z/OS del gestore code. Ci possono essere migliaia di processi MCA in esecuzione contemporaneamente all'interno dell'iniziatore di canali.

Figura 57 a pagina 167 mostra due gestori code all'interno di un sysplex. Ogni gestore code ha un iniziatore di canali e una coda locale. I messaggi inviati dai gestori code su AIX e Windows vengono collocati nella coda locale, da dove vengono richiamati da un'applicazione. I messaggi di risposta vengono restituiti da un instradamento simile.

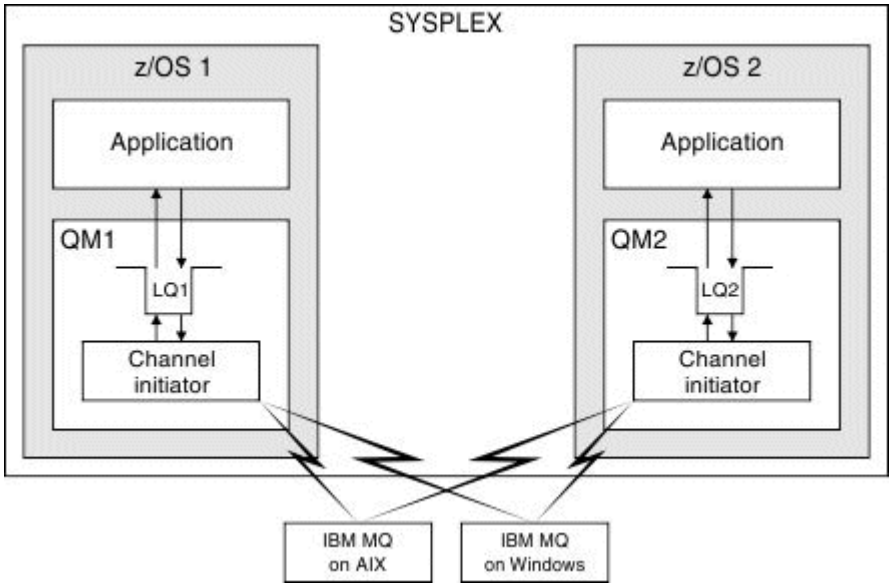


Figura 57. Comunicazione tra gestori code

L'iniziatore del canale contiene anche altri processi relativi alla gestione dei canali. Questi processi includono:

Listener
Questi processi sono in attesa di richieste di canali in entrata su un sottosistema di comunicazione come TCP e avviano un MCA denominato quando viene ricevuta una richiesta in entrata.

Supervisore
Questo gestisce lo spazio di indirizzo dell'iniziatore di canali, ad esempio è responsabile del riavvio dei canali dopo un errore.

Definisci server
Viene utilizzato per risolvere i nomi TCP in indirizzi.

Attività TLS
Questi vengono utilizzati per eseguire la codifica e la decrittografia e controllare gli elenchi di revoca dei certificati.

z/OS Record SMF per l'iniziatore di canali

L'iniziatore di canali (CHINIT) può produrre record di statistiche SMF e record di account con informazioni su attività e canali.

Il CHINIT può produrre record di statistiche SMF e record di contabilità con i seguenti tipi di informazioni:

- Le attività: dispatcher, adattatore, DNS (Domain Name Server) e SSL. Queste attività formano le cosiddette statistiche CHINIT.
- Canali: fornisce informazioni di account simili a quelle disponibili con il comando DIS CHSTATUS. Questo è chiamato account di canale.

IBM MQ for Multiplatforms fornisce informazioni simili scrivendo messaggi PCF nel SISTEMA SYSTEM.ADMIN.STATISTICS.QUEUE. Consultare Dati del messaggio delle statistiche del canale per ulteriori informazioni su come vengono registrate le informazioni statistiche su IBM MQ for Multiplatforms.

Dati statistiche

È possibile utilizzare queste informazioni per trovare le seguenti informazioni:

- Indica se sono necessarie più attività CHINIT, ad esempio il numero di TCB SSL e la quantità di CPU utilizzata da tali attività.
- Il tempo medio per le richieste su queste attività.
- La richiesta di durata più lunga nell'intervallo e l'ora del giorno in cui ciò si è verificato, per le attività DNS e SSL. È possibile correlare questa ora del giorno con i problemi che potrebbero verificarsi con il canale.

Dati contabili

È possibile utilizzare queste informazioni per monitorare l'utilizzo del canale e trovare le seguenti informazioni:

- I canali con la velocità di trasmissione più elevata.
- La frequenza con cui i messaggi sono stati inviati e la velocità di invio dei dati in MB/secondo.
- La dimensione batch raggiunta. Se la dimensione batch raggiunta è vicina alla dimensione batch specificata per il canale, il canale potrebbe essere vicino al limite per l'invio dei messaggi.

Utilizzare i comandi START TRACE e STOP TRACE per controllare la raccolta della traccia di account e della traccia delle statistiche. È possibile utilizzare le opzioni STATCHL e STATACLS sul canale e sul gestore code per controllare se i canali producono dati SMF.

z/OS Termini e attività per la gestione di IBM MQ for z/OS

Utilizzare questo argomento come introduzione alla terminologia e alle attività specifiche di IBM MQ for z/OS.

Alcuni dei termini e delle attività richiesti per la gestione di IBM MQ for z/OS sono specifici della piattaforma z/OS . Il seguente elenco contiene alcuni di questi termini e attività.

- Code condivise
- Serie di pagine e pool di buffer
- Registrazione
- Adattamento dell'ambiente del gestore code
- Riavvio e ripristino

- Sicurezza
- Disponibilità
- Manipolazione degli oggetti
- Monitoraggio e statistiche
- ambienti di applicazione

Code condivise

Le code possono essere *non condivise*, di proprietà e accessibili a un solo gestore code o *condivise*, di proprietà di un *gruppo di condivisione code*. Un gruppo di condivisione code è costituito da un numero di gestori code, in esecuzione all'interno di un singolo sysplex z/OS , che possono accedere simultaneamente alle stesse definizioni di oggetto IBM MQ e ai dati di messaggio. All'interno di un gruppo di condivisione code, le definizioni di oggetto condivisibili vengono memorizzate in un database Db2 condiviso. I messaggi della coda condivisa vengono conservati all'interno di una o più strutture CF (coupling facility structure). Se i dati del messaggio sono troppo grandi per essere memorizzati direttamente nella struttura (più di 63 KB di dimensione) o se il messaggio è abbastanza grande che le regole definite dall'installazione lo selezionano per l'offload, le informazioni di controllo del messaggio vengono ancora memorizzate nella voce CF (coupling facility), ma i dati del messaggio vengono scaricati in un SMDS (shared message data set) o in un database Db2 condiviso. I dataset di messaggi condivisi, il database Db2 condiviso e le strutture CF (Coupling Facility) sono risorse gestite congiuntamente da tutti i gestori code del gruppo.

Serie di pagine e pool di buffer

Quando un messaggio viene inserito in una coda non condivisa, il gestore code memorizza i dati in una serie di pagine in modo che possano essere recuperati quando un'operazione successiva richiama un messaggio dalla stessa coda. Se il messaggio viene rimosso dalla coda, lo spazio nella serie di pagine che contiene i dati viene successivamente liberato per il riutilizzo. Man mano che il numero di messaggi conservati su una coda aumenta, la quantità di spazio utilizzato nella serie di pagine aumenta e il numero di messaggi su una coda si riduce, lo spazio utilizzato nella serie di pagine si riduce.

Per ridurre il costo delle prestazioni di scrittura e lettura dei dati dalle serie di pagine, il gestore code memorizza nel buffer gli aggiornamenti nella memoria del processore. La quantità di memoria utilizzata per memorizzare nel buffer l'accesso alla serie di pagine è controllata tramite IBM MQ oggetti denominati *pool di buffer*.

Per ulteriori informazioni sulle serie di pagine e sui pool di buffer, consultare Gestione memoria.

Registrazione

Le modifiche agli oggetti contenuti nelle serie di pagine e le operazioni sui messaggi persistenti vengono registrate come record di log. Questi record di log vengono scritti in un dataset di log denominato *log attivo*. Il nome e la dimensione del dataset del log attivo si trovano in un dataset denominato *dataset bootstrap* (BSDS).

Quando il dataset del log attivo si riempie, il gestore code passa a un'altro dataset di log in modo che la registrazione possa continuare e copia il contenuto del dataset del log attivo completo in un data set *log archivio* . Le informazioni su queste azioni, incluso il nome del dataset del log di archivio, sono contenute nel dataset di bootstrap. Concettualmente, esiste un anello di dataset di log attivi in cui il gestore code esegue il ciclo; quando un log attivo viene riempito, i dati di log vengono scaricati in un log di archivio e il dataset di log attivo è disponibile per il riutilizzo.

Per ulteriori informazioni sui dataset di log e bootstrap, consultare “accessoIBM MQ for z/OS” a pagina 236.

Adattamento dell'ambiente del gestore code

Quando il gestore code viene avviato, viene letta una serie di parametri di inizializzazione che controllano il funzionamento del gestore code. Inoltre, vengono letti i dataset contenenti i comandi IBM MQ e vengono eseguiti i comandi contenuti. Generalmente, questi dataset contengono le definizioni degli oggetti di sistema richiesti per l'esecuzione di IBM MQ ed è possibile personalizzarli per definire o inizializzare gli oggetti IBM MQ necessari per il proprio ambiente operativo. Quando questi dataset sono stati letti, tutti gli oggetti da essi definiti vengono memorizzati, su una serie di pagine o in Db2.

Per ulteriori informazioni sui parametri di inizializzazione e sugli oggetti di sistema, consultare “Definizione di sistema su z/OS” a pagina 247.

Ripristino e riavvio

In qualsiasi momento durante il funzionamento di IBM MQ, potrebbero essere presenti modifiche nella memoria del processore che non sono state ancora scritte nella serie di pagine. Queste modifiche vengono scritte nella serie di pagine che è la meno recente utilizzata da un'attività in background all'interno del gestore code.

Se il gestore code viene terminato in modo anomalo, la fase di ripristino del riavvio del gestore code può ripristinare le modifiche perse della serie di pagine poiché i dati dei messaggi persistenti sono conservati nei record di log. Ciò significa che IBM MQ può ripristinare i dati dei messaggi persistenti e le modifiche degli oggetti fino al punto di errore.

Se un gestore code che è un membro di un gruppo di condivisione code rileva un errore CF (Coupling Facility), i messaggi persistenti su tale coda possono essere recuperati solo se è stato eseguito il backup della struttura CF (Coupling Facility).

Per ulteriori informazioni sul ripristino e il riavvio, consultare “Ripristino e riavvio su z/OS” a pagina 258.

Sicurezza

È possibile utilizzare un gestore della sicurezza esterno, ad esempio Security Server (precedentemente noto come RACF) per proteggere le risorse che IBM MQ possiede e gestisce dall'accesso da parte di utenti non autorizzati. È anche possibile utilizzare TLS (Transport Layer Security) per la sicurezza del canale. TLS è incluso come parte del prodotto IBM MQ .

Per ulteriori informazioni sulla sicurezza IBM MQ , consultare “Concetti di sicurezza in IBM MQ for z/OS” a pagina 275.

Disponibilità

Ci sono diverse funzioni di IBM MQ progettate per aumentare la disponibilità del sistema in caso di errore del gestore code o del sottosistema di comunicazione. Per ulteriori informazioni su queste funzioni, consultare “Disponibilità su z/OS” a pagina 282.

Manipolazione di oggetti

Quando il gestore code è in esecuzione, è possibile manipolare gli oggetti IBM MQ tramite un'interfaccia della console z/OS o tramite un programma di utilità di gestione che utilizza i servizi ISPF in TSO. Entrambi i meccanismi consentono di definire, modificare o eliminare gli oggetti IBM MQ . È inoltre possibile controllare e visualizzare lo stato di varie IBM MQ e funzioni del gestore code.

Per ulteriori informazioni su queste funzioni, consultare Origini da cui è possibile immettere i comandi MQSC e PCF su IBM MQ for z/OS.

È inoltre possibile modificare gli oggetti IBM MQ utilizzando IBM MQ Explorer, una GUI (graphical user interface) che fornisce un modo visivo di gestire code, gestori code e altri oggetti.

Monitoraggio e statistiche

Sono disponibili diverse funzioni per monitorare i gestori code e gli iniziatori di canali. È inoltre possibile raccogliere statistiche per la valutazione delle prestazioni e la contabilità.

Per ulteriori informazioni su queste funzioni, consultare “Monitoraggio e statistiche su IBM MQ for z/OS” a pagina 285.

Ambienti di applicazione

Una volta avviato il gestore code, le applicazioni possono connettersi ad esso e iniziare a utilizzare l'API IBM MQ . Possono essere applicazioni CICS, IMS, Batch o WebSphere Application Server . Le applicazioni IBM MQ possono anche accedere alle applicazioni su sistemi CICS e IMS che non sono a conoscenza di IBM MQ, utilizzando i bridge CICS e IMS .

Per ulteriori informazioni su queste funzioni, consultare “IBM MQ e altri prodotti z/OS” a pagina 289.

Per informazioni sulla scrittura di applicazioni IBM MQ , consultare la seguente documentazione:

- Sviluppo delle applicazioni
- Utilizzo di C++
- Utilizzo di IBM MQ classes for Java

z/OS Code condivise e gruppi di condivisione code

È possibile utilizzare code condivise e gruppi di condivisione code per implementare l'alta disponibilità delle risorse IBM MQ . Le code condivise e i gruppi di condivisione code sono funzioni univoche per IBM MQ for z/OS sulla piattaforma z/OS .

Questa sezione descrive gli attributi e i vantaggi e offre informazioni sul modo in cui diversi gestori code possono condividere le stesse code e i messaggi su tali code.

Cos' è una coda condivisa?

Una coda condivisa è un tipo di coda locale. Ai messaggi su tale coda possono accedere uno o più gestori code che si trovano in un sysplex.

Gruppo di condivisione code

I gestori code che accedono alla stessa serie di code condivise formano un gruppo denominato *gruppo di condivisione code*.

Qualsiasi gestore code può accedere ai messaggi

Qualsiasi gestore code nel gruppo di condivisione code può accedere a una coda condivisa. Ciò significa che è possibile inserire un messaggio in una coda condivisa su un gestore code e ricevere lo stesso messaggio dalla coda da un gestore code differente. Questo fornisce un meccanismo rapido per la comunicazione all'interno di un gruppo di condivisione code che non richiede canali attivi tra gestori code.

IBM MQ supporta lo scaricamento dei messaggi in Db2 o in un SMDS (shared message data set). L'offload di messaggi di qualsiasi dimensione è configurabile.

Figura 58 a pagina 172 mostra tre gestori code e una CF (coupling facility), che formano un gruppo di condivisione code. Tutti e tre i gestori code possono accedere alla coda condivisa nella CF (coupling facility).

Un'applicazione può connettersi a qualsiasi gestore code all'interno del gruppo di condivisione code. Poiché tutti i gestori code nel gruppo di condivisione code possono accedere a tutte le code condivise, l'applicazione non dipende dalla disponibilità di un gestore code specifico; qualsiasi gestore code nel gruppo di condivisione code può servire la coda.

Ciò fornisce una maggiore disponibilità in quanto tutti gli altri gestori code nel gruppo di condivisione code possono continuare l'elaborazione della coda se uno dei gestori code ha un problema.

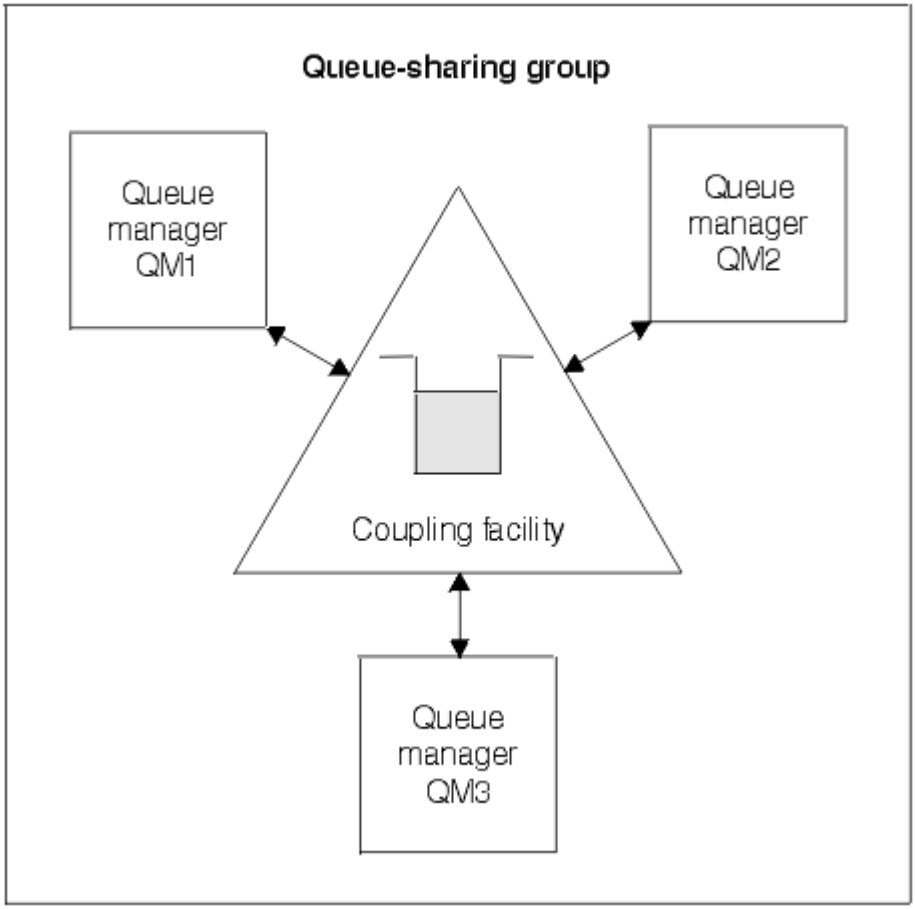


Figura 58. Gruppo di condivisione code

La definizione di coda è condivisa da tutti i gestori code

Le definizioni delle code condivise vengono memorizzate nella tabella del database Db2 OBJ_B_QUEUE. Per questo motivo, è necessario definire la coda una sola volta e quindi è possibile accedervi da tutti i gestori code nel gruppo di condivisione code. Ciò significa che ci sono meno definizioni da fare.

Al contrario, la definizione di una coda non condivisa viene memorizzata nella serie di pagine zero del gestore code che possiede la coda (come descritto in Serie di pagine).

Non è possibile definire una coda condivisa se una coda con tale nome è già stata definita nelle serie di pagine del gestore code di definizione. Allo stesso modo, non è possibile definire una versione locale di una coda nelle serie di pagine del gestore code se esiste già una coda condivisa con lo stesso nome.

Cos' è un gruppo di condivisione code?

Un gruppo di gestori code che possono accedere alle stesse code condivise è denominato gruppo di condivisione code. Ogni membro del gruppo di condivisione code ha accesso alla stessa serie di code condivise.

I gruppi di condivisione code hanno un nome composto da un massimo di quattro caratteri. Il nome deve essere univoco nella rete e diverso da qualsiasi altro nome di gestore code.

Figura 59 a pagina 173 illustra un gruppo di condivisione code che contiene due gestori code. Ogni gestore code ha un iniziatore di canali e le proprie serie di pagine locali e dataset di log.

Ogni membro del gruppo di condivisione code deve anche connettersi a un sistema Db2 . I sistemi Db2 devono essere tutti nello stesso gruppo di condivisione dati Db2 in modo che i gestori code possano accedere al repository condiviso Db2 utilizzato per conservare le definizioni di oggetti condivisi. Si tratta di definizioni di qualsiasi tipo di oggetto IBM MQ (ad esempio, code e canali) che vengono definite una sola volta e che possono essere utilizzate da qualsiasi gestore code del gruppo. Queste sono denominate definizioni *globali* e sono descritte in Definizioni private e globali.

Più di un gruppo di condivisione code può fare riferimento a un particolare gruppo di condivisione dati. Specificare il nome del sottosistema Db2 e il gruppo di condivisione dati che un gestore code utilizza nei parametri di sistema IBM MQ all'avvio.

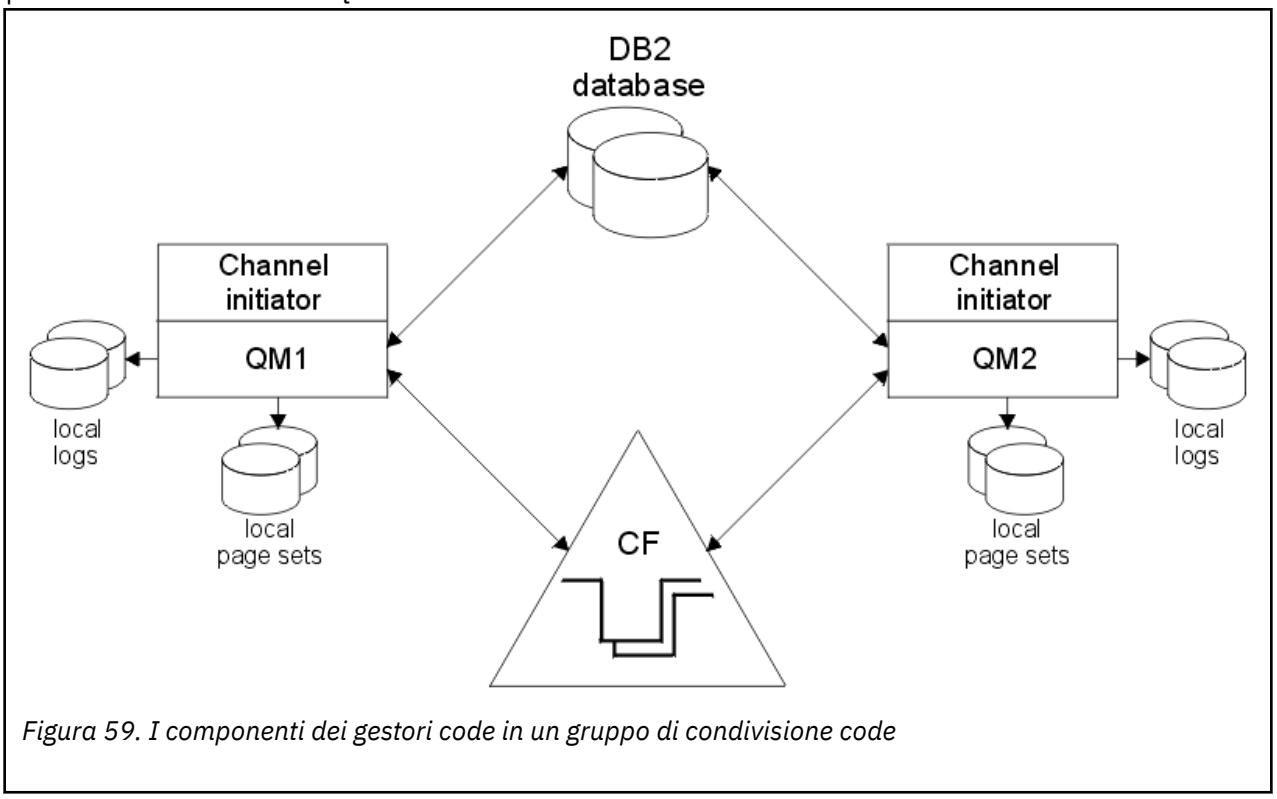


Figura 59. I componenti dei gestori code in un gruppo di condivisione code

Quando un gestore code si unisce ad un gruppo di condivisione code, ha accesso agli oggetti condivisi definiti per tale gruppo ed è possibile utilizzare tale gestore code per definire nuovi oggetti condivisi all'interno del gruppo. Se le code condivise sono definite all'interno del gruppo, è possibile utilizzare questo gestore code per inserire e ricevere messaggi da tali code condivise. Qualsiasi gestore code del gruppo può richiamare i messaggi contenuti in una coda condivisa.

È possibile immettere un comando MQSC una volta e farlo eseguire su tutti i gestori code all'interno del gruppo di condivisione code come se fosse stato immesso singolarmente su ciascun gestore code. L'attributo *ambito comando* viene utilizzato per questo. Questo attributo è descritto in Indirizzamento di comandi a gestori code differenti.

Quando un gestore code viene eseguito come membro di un gruppo di condivisione code, deve essere possibile distinguere tra gli oggetti IBM MQ definiti privatamente per tale gestore code e gli oggetti IBM MQ definiti globalmente che sono disponibili per tutti i gestori code nel gruppo di condivisione code. L'attributo *disposizione del gruppo di condivisione code* viene utilizzato per questo. Questo attributo è descritto in Definizioni private e globali.

È possibile definire una singola serie di profili di protezione che controllano l'accesso agli oggetti IBM MQ in qualsiasi punto del gruppo. Ciò significa che il numero di profili da definire è notevolmente ridotto.

Un gestore code può appartenere a un solo gruppo di condivisione code e tutti i gestori code del gruppo devono trovarsi nello stesso sysplex. Specificare a quale gruppo di condivisione code appartiene il gestore code nei parametri di sistema all'avvio.

Concetti correlati

“Dove vengono conservati i messaggi della coda condivisa?” a pagina 174
Ogni messaggio su una coda condivisa è rappresentato da una voce in una struttura di elenco CF (Coupling Facility) z/OS . Se i dati del messaggio sono troppo grandi per rientrare nella stessa voce, vengono scaricati in un SMDS (shared message data set) o in Db2.

“Vantaggi dell'utilizzo delle code condivise” a pagina 191
La coda condivisa consente alle applicazioni IBM MQ di essere scalabili, altamente disponibili e consente l'implementazione del bilanciamento del carico di lavoro.

“Accodamento distribuito e gruppi di condivisione code” a pagina 212
L'accodamento distribuito e i gruppi di condivisione code sono due tecniche che possono essere utilizzate per aumentare la disponibilità dei propri sistemi dell'applicazione. Utilizzare questo argomento per trovare ulteriori informazioni su queste tecniche.

“Influenza della distribuzione del carico di lavoro con le code condivise” a pagina 215
Utilizzare questo argomento per comprendere i fattori che influenzano la distribuzione del carico di lavoro con code condivise in un gruppo di condivisione code.

Riferimenti correlati

“Dove trovare ulteriori informazioni sulle code condivise e sui gruppi di condivisione code” a pagina 216
Utilizzare la tabella in questo argomento per trovare ulteriori informazioni su come IBM MQ for z/OS utilizza le code condivise e i gruppi di condivisione code.

z/OS Dove vengono conservati i messaggi della coda condivisa?

Ogni messaggio su una coda condivisa è rappresentato da una voce in una struttura di elenco CF (Coupling Facility) z/OS . Se i dati del messaggio sono troppo grandi per rientrare nella stessa voce, vengono scaricati in un SMDS (shared message data set) o in Db2.

Se la struttura CF è stata configurata per utilizzare SCM (System Class Memory), IBM MQ può utilizzarla senza alcuna configurazione aggiuntiva.

Importante: IBM z16 è pianificato per essere l'ultima generazione di IBM Z ® per supportare l'utilizzo di memoria flash virtuale (nota anche come SCM (Storage Class Memory) o SCM) per le immagini della CF (Coupling Facility). Per ulteriori informazioni, consultare IBM Z e IBM LinuxONE 4Q 2023 Statement of Direction.

In alternativa, è necessario utilizzare strutture più grandi o scaricare i messaggi su SMDS.

Memoria messaggi coda condivisa

I messaggi inseriti nelle code condivise non vengono memorizzati nelle serie di pagine e non utilizzano i pool di buffer.

I messaggi nelle code condivise hanno voci nelle strutture di elenco nella CF (Coupling Facility) z/OS . Molti gestori code nello stesso sysplex possono accedere a tali messaggi utilizzando la struttura di elenco CF.

I dati del messaggio per i piccoli messaggi della coda condivisa sono normalmente inclusi nella voce CF (Coupling Facility). Per messaggi di dimensioni maggiori, i dati del messaggio possono essere memorizzati in un SMDS (shared message data set) o come uno o più BLOB (binary large object) in una tabella Db2 condivisa da un gruppo di condivisione dati Db2 . I dati dei messaggi superiori a 63 KB vengono sempre scaricati su SMDS o Db2. I messaggi più piccoli possono anche essere scaricati nello stesso modo per risparmiare spazio nella struttura CFS (coupling facility structure). Consultare “Specifica delle opzioni di offload per i messaggi condivisi” a pagina 176 per maggiori dettagli.

I messaggi inseriti in una coda condivisa vengono indicati in una struttura CFS (coupling facility structure) fino a quando non vengono richiamati da un MQGET. Le operazioni CF vengono utilizzate per:

- Cerca il prossimo messaggio richiamabile
- Blocca i messaggi senza commit sulle code condivise

- Notifica ai gestori code interessati l'arrivo dei messaggi di cui è stato eseguito il commit

Le operazioni MQPUT e MQGET sui messaggi persistenti vengono registrate nel log del gestore code che esegue tale operazione. Ciò riduce il rischio di perdita di dati in caso di errore della CF (Coupling Facility).

CF (Coupling Facility)

I messaggi contenuti nelle code condivise vengono utilizzati come riferimento all'interno di una CF (coupling facility). La CF si trova all'esterno di qualsiasi immagine z/OS nel sysplex ed è generalmente configurata per essere eseguita su un alimentatore differente. La CF è quindi resiliente ai malfunzionamenti software ed è possibile configurarlo in modo che sia resiliente ai malfunzionamenti hardware o alle interruzioni di alimentazione. Ciò significa che i messaggi memorizzati nella CF (Coupling Facility) sono altamente disponibili.

Ogni struttura di elenco CF utilizzata da IBM MQ è dedicata a uno specifico gruppo di condivisione code, ma una CF può contenere strutture per più di un gruppo di condivisione code. I gestori code in gruppi di condivisione code diversi non possono condividere dati. Fino a 32 gestori code in un gruppo di condivisione code possono connettersi a una struttura di elenco CF (Coupling Facility) contemporaneamente.

Una singola struttura dell'elenco CF (Coupling Facility) può contenere fino a 512 code condivise. La quantità totale di dati del messaggio memorizzati nella struttura è limitata dalla capacità della struttura. Tuttavia, con **CFLEVEL(5)** è possibile utilizzare i parametri di offload per eseguire l'offload dei dati per i messaggi inferiori a 63 KB, aumentando così il numero di messaggi che possono essere memorizzati nella struttura, anche se ogni messaggio richiede ancora almeno una voce Coupling facility più almeno 768 byte di dati, composti da 256 byte per la voce e 512 byte per i due elementi di intestazione e descrittore.

La dimensione della struttura dell'elenco è limitata dai fattori riportati di seguito:

- Deve trovarsi all'interno di un'unica struttura di accoppiamento.
- Potrebbe condividere la memoria CF disponibile con altre strutture per IBM MQ e altri prodotti.

Le strutture dell'elenco CF possono avere la memoria della classe di memoria associata. In alcune situazioni questa memoria della classe di memoria può essere utile quando viene utilizzata con le code condivise. Per ulteriori informazioni, consultare “Utilizzo della memoria della classe storage con code condivise” a pagina 192.

Pianificazione della dimensione della struttura CF

Se si richiede una guida per il dimensionamento delle strutture CF, è possibile utilizzare il supporto MP16: IBM MQ for z/OS Capacity planning and tuning . Puoi anche utilizzare lo strumento basato sul web CFSizerfornito da IBM per assistere con le dimensioni della CF.

L'oggetto struttura CF

L'utilizzo da parte del gestore code di una struttura CFS (Coupling Facility Structure) è specificato in un oggetto IBM MQ CFSTRUCT (CF structure).

Questi oggetti di struttura sono memorizzati in Db2.

Quando si utilizzano i comandi o le definizioni z/OS relativi a una struttura CFS (coupling facility structure), sono richiesti i primi quattro caratteri del nome del gruppo di condivisione code. Tuttavia, un oggetto IBM MQ CFSTRUCT esiste sempre all'interno di un singolo gruppo di condivisione code e quindi il suo nome non include i primi quattro caratteri del nome del gruppo di condivisione code. Ad esempio, CFSTRUCT (MYDATA) definito nel gruppo di condivisione code che inizia con SQ03 utilizzerà la struttura dell'elenco CF (Coupling Facility) SQ03MYDATA.

Le strutture CF hanno un attributo CFLEVEL che ne determina la capacità funzionale:

- 1, 2 - può essere utilizzato per messaggi non persistenti inferiori a 63 KB
- 3 - può essere utilizzato per messaggi persistenti e non persistenti inferiori a 63 KB
- 4 - può essere utilizzato per messaggi persistenti e non persistenti fino a 100 MB
- 5 - può essere utilizzato per messaggi persistenti e non persistenti fino a 100 MB e scaricati selettivamente in SMDS (shared message data sets) o Db2.

Nota: Quando si utilizza IBM MQ è possibile codificare una struttura CFS (coupling facility structure). Per ulteriori informazioni, consultare Codifica dei dati della struttura CFS .

Backup e ripristino della CF (coupling facility)

È possibile eseguire il backup delle strutture dell'elenco CF utilizzando il comando IBM MQ BACKUP CFSTRUCT. Questa operazione inserisce una copia dei messaggi persistenti attualmente all'interno della struttura CF nel dataset del log attivo del gestore code che esegue il backup e scrive un record del backup in Db2.

Se la CF ha esito negativo, è possibile utilizzare il comando IBM MQ RECOVER CFSTRUCT. Questo utilizza il record di backup da Db2 per individuare e ripristinare i messaggi persistenti dal backup della struttura CF. Qualsiasi attività dall'ultimo backup viene rieseguita utilizzando i log di tutti i gestori code nel gruppo di condivisione code e la struttura della CF viene quindi ripristinata fino al momento precedente l'errore.

Per ulteriori dettagli, consultare i comandi BACKUP CFSTRUCT e RECOVER CFSTRUCT .

Concetti correlati

“Specifica delle opzioni di offload per i messaggi condivisi” a pagina 176
È possibile scegliere dove memorizzare i dati del messaggio per un messaggio della coda condivisa, in una tabella Db2 o in un SMDS (shared message data set). È anche possibile selezionare quali messaggi vengono scaricati, in base alla dimensione del messaggio e all'uso corrente della CF (Coupling Facility Structure).

“Gestione dell'ambiente SMDS (shared message data set)” a pagina 179
Se si selezionano i dataset di messaggi condivisi per scaricare i messaggi di grandi dimensioni, è necessario essere consapevoli delle informazioni che IBM MQ utilizza per gestire tali dataset e dei comandi utilizzati per gestire tali informazioni. Utilizzare questo argomento per comprendere come gestire i dataset di messaggi condivisi.

z/OS *Specifica delle opzioni di offload per i messaggi condivisi*

È possibile scegliere dove memorizzare i dati del messaggio per un messaggio della coda condivisa, in una tabella Db2 o in un SMDS (shared message data set). È anche possibile selezionare quali messaggi vengono scaricati, in base alla dimensione del messaggio e all'uso corrente della CF (Coupling Facility Structure).

I dati del messaggio per le code condivise possono essere scaricati dalla CF (Coupling Facility) e memorizzati in una tabella Db2 o in un dataset gestito IBM MQ denominato SMDS (*shared message data set*).

Per i messaggi più grandi della dimensione della voce CF di 63 KB, lo scaricamento dei dati dei messaggi in un SMDS può avere un miglioramento significativo delle prestazioni rispetto allo scaricamento in Db2.

Ogni messaggio della coda condivisa viene ancora gestito utilizzando una voce di elenco in una struttura CFS (coupling facility structure), ma quando i dati del messaggio vengono scaricati su SMDS, la voce CFS contiene solo alcune informazioni di controllo e un elenco di riferimenti ai blocchi disco rilevanti in cui è memorizzato il messaggio. Utilizzando questo meccanismo, la quantità di memoria dell'elemento CF richiesta per ciascun messaggio è solo una frazione della dimensione effettiva del messaggio.

Selezione della posizione in cui sono memorizzati i messaggi della coda condivisa

La selezione dell'archivio di messaggi condivisi SMDS o Db2 è controllata con il parametro **OFFLOAD(SMDS|DB2)** nella definizione **CFSTRUCT** . **OFFLOAD(SMDS)** è il valore predefinito.

Questo parametro richiede anche **CFSTRUCT** per utilizzare **CFLEVEL(5)** o superiore.

Il parametro **OFFLOAD** è valido solo da **CFLEVEL(5)**. Per ulteriori dettagli, consultare DEFINE CFSTRUCT .

OFFLOAD(DB2) è supportato principalmente per scopi di migrazione.

Selezione dei messaggi della coda condivisa di cui è stato eseguito l'offload

I dati del messaggio vengono scaricati in SMDS o Db2 in base alla dimensione dei dati del messaggio e all'utilizzo corrente della struttura CFS (coupling facility structure). Esistono tre regole e ciascuna regola specifica una coppia di parametri corrispondente. Questi parametri sono una corrispondente percentuale della soglia di utilizzo della struttura CFS (**OFFLDnTH**) e un limite di dimensione del messaggio (**OFFLDnSZ**).

L'implementazione corrente delle tre regole viene specificata utilizzando le seguenti coppie di parole chiave:

- OFFLD1TH e OFFLD1SZ
- OFFLD2TH e OFFLD2SZ
- OFFLD3TH e OFFLD3SZ

Coppia di regole	Valore predefinito	Descrizione
Coppia di regole 1	OFFLD1TH(70) e OFFLD1SZ(32K)	Se la struttura CFS (coupling facility structure) è più del 70% di dati di scaricamento completo per i messaggi che superano i 32 KB
Coppia di regole 2	OFFLD2TH(80) e OFFLD2SZ(4K)	Se la struttura CFS (coupling facility structure) è superiore all ' 80% di dati di scaricamento completo per i messaggi che superano i 4 KB
Coppia di regole 3	OFFLD3TH(90) e OFFLD3SZ(0K)	Se la struttura CFS (Coupling Facility Structure) è più del 90% di dati di scaricamento completo per i messaggi che superano 0 KB (tutti i messaggi)

Se una regola offload ha il valore OFFLD *x* SZ di 64K , ciò indica che la regola non è effettiva. In questo caso, i messaggi verranno scaricati solo se è attiva un'altra regola di scaricamento o se il messaggio è maggiore di 63.75 KB e quindi è troppo grande per essere memorizzato nella struttura.

Ogni messaggio scaricato richiede ancora 0.75 KB di memoria nella CF (Coupling Facility).

Le tre regole di offload che possono essere specificate per ogni struttura sono destinate ad essere utilizzate come segue.

- Prestazioni
  - Quando c'è molto spazio nella struttura dell'applicazione, i dati del messaggio devono essere scaricati solo se sono troppo grandi da memorizzare nella struttura o se superano una soglia di dimensione del messaggio inferiore, in modo che il valore delle prestazioni della memorizzazione nella struttura non valga la quantità di spazio della struttura di cui avrebbe bisogno.
  - Se è richiesta una soglia di dimensione del messaggio specifica, viene convenzionalmente specificata utilizzando la prima regola di offload.
- Capacità
  - Quando lo spazio nella struttura dell'applicazione è molto limitato, la quantità massima di dati del messaggio deve essere scaricata in modo da utilizzare al meglio lo spazio rimanente.

  - La terza regola di offload viene utilizzata in modo convenzionale per indicare che quando la struttura è quasi piena, la maggior parte dei messaggi deve essere scaricata, in modo che le voci nella struttura dell'applicazione siano generalmente della dimensione minima (che richiede circa 0.75K byte).
  - Il parametro di soglia di utilizzo deve essere selezionato in base alla dimensione della struttura dell'applicazione e al backlog massimo anticipato. Ad esempio, se il backlog anticipato massimo è 1M messaggi, la quantità di memoria della struttura richiesta per questo numero di messaggi è di circa 0.75G byte. Ciò significa, ad esempio, che se la struttura è di circa 10G byte, la soglia di utilizzo per l'offload di tutti i messaggi deve essere impostata su un valore pari o inferiore al 92%.
  - Lo spazio della struttura è diviso in elementi e voci, e anche se ci può essere spazio sufficiente in generale, uno di questi può esaurire prima dell'altro. Il sistema fornisce le funzionalità AUTOALTER per regolare il rapporto quando necessario, ma questo non è molto sensibile, quindi la quantità di spazio effettivamente disponibile potrebbe essere un po 'meno. Può essere preferibile quindi utilizzare non più del 90% dello spazio massimo della struttura, quindi nell'esempio precedente, la soglia di utilizzo per scaricare tutti i messaggi sarebbe impostata meglio intorno all ' 80%.
- Transizione ammortizzata:
  - Poiché la quantità di spazio rimasto nella struttura della funzione di accoppiamento diminuisce, non sarebbe auspicabile avere un grande cambiamento improvviso nelle caratteristiche delle prestazioni. Non è inoltre auspicabile che la gestione della CF abbia una modifica improvvisa della soglia nel rapporto tipico tra le voci e gli elementi utilizzati.
  - La seconda regola di offload è convenzionalmente utilizzata per fornire un ammortizzamento intermedio tra le regole di offload con distorsione delle prestazioni e della capacità. Può essere impostato in modo da causare un aumento significativo dell'attività di offload quando lo spazio utilizzato nella struttura CFS supera una soglia intermedia. Ciò significa che lo spazio rimanente viene utilizzato più lentamente e fornisce all'elaborazione di modifica automatica della CF più tempo per adattarsi ai livelli di utilizzo più elevati.

Se la struttura CFS non può essere espansa ed è necessario memorizzare almeno un numero predeterminato di messaggi, la terza regola può essere modificata come necessario per garantire che lo scaricamento dei dati per tutti i messaggi inizi ad una soglia appropriata per garantire che lo spazio sia riservato per quel numero predeterminato di messaggi.

Ad esempio, se la dimensione della struttura CF (Coupling Facility) è 4 GB e il numero predeterminato di messaggi è 1 milione, sono necessari 1,000,000 * 0.75 KB, che è 768 MB, 18.75% di 4 GB. In questo caso la soglia per scaricare tutti i messaggi deve essere impostata intorno all ' 80% anziché al 90%. Fornisce i parametri OFFLD3TH(80) e OFFLD3SZ(0K). Anche gli altri parametri di offload devono essere modificati.

Se si rileva che l'offload di messaggi molto piccoli ha un impatto sulle prestazioni significativo, ma l'impatto relativo è minore per i messaggi più grandi, è possibile ridurre le soglie di utilizzo per le altre regole per l'offload di messaggi più grandi in anticipo, lasciando più spazio nella struttura per i messaggi più piccoli prima che debbano essere scaricati.

Ad esempio, se i messaggi che superano i 32KB si verificano frequentemente, ma le prestazioni del tempo trascorso per scaricarli (come determinato dalle statistiche RMF o dalle prestazioni dell'applicazione) sono molto simili a quelle per mantenerli nella CF (Coupling Facility), la soglia per la prima regola potrebbe essere impostata su 0% per scaricare tutti questi messaggi. Fornisce i parametri OFFLD1TH(0) e OFFLD1SZ(32K). Anche in questo caso, gli altri parametri di offload dovranno essere modificati.

Se ci sono molti messaggi intorno a specifiche dimensioni intermedie, come ad esempio 16 KB e 6 KB, potrebbe essere utile modificare l'opzione di dimensione del messaggio per la seconda regola in modo che quelli più grandi vengano scaricati a una soglia di utilizzo abbastanza bassa, risparmiando una quantità significativa di spazio, ma quelli più piccoli vengono ancora memorizzati solo nella CF.

z/OS *Gestione dell'ambiente SMDS (shared message data set)*

Se si selezionano i dataset di messaggi condivisi per scaricare i messaggi di grandi dimensioni, è necessario essere consapevoli delle informazioni che IBM MQ utilizza per gestire tali dataset e dei comandi utilizzati per gestire tali informazioni. Utilizzare questo argomento per comprendere come gestire i dataset di messaggi condivisi.

Oggetti SMDS

Le proprietà e lo stato di ciascun dataset di messaggi condivisi vengono tracciati in un oggetto SMDS condiviso che può essere aggiornato tramite qualsiasi gestore code nel gruppo di condivisione code.

Esiste un dataset di messaggi condivisi per ciascun gestore code che può accedere a ciascuna struttura dell'applicazione Coupling Facility. Il dataset del messaggio condiviso è identificato dal nome del gestore code proprietario, specificato utilizzando la parola chiave SMDS, e dal nome della struttura dell'applicazione, specificato utilizzando la parola chiave CFSTRUCT.

Nota: Quando si definiscono i dataset SMDS per una struttura, è necessario averne uno per ciascun gestore code.

L'oggetto SMDS è memorizzato in un array (con una voce per gestore code nel gruppo) che forma un'estensione dell'oggetto CFSTRUCT corrispondente memorizzato in Db2.

Non è presente alcun comando per DEFINE o DELETE dell'oggetto SMDS perché è stato creato o eliminato come parte dell'oggetto CFSTRUCT, ma è presente un comando per ALTER per modificarne le impostazioni per un singolo gestore code proprietario.

Per ulteriori informazioni sui comandi SMDS, consultare “Comandi correlati a SMDS” a pagina 190

Informazioni SMDSCONN

È possibile che un dataset di messaggi condivisi si trovi in uno stato normale, ma che uno o più gestori code non siano in grado di connettersi ad esso, ad esempio a causa di un problema con una definizione di sicurezza o con la connettività del dispositivo di accesso diretto. Pertanto, è necessario che ciascun gestore code tenga traccia dello stato della connessione e delle informazioni sulla disponibilità per ciascun dataset di messaggi condivisi, indicando ad esempio se è attualmente in grado di connettersi ad esso e, in caso contrario, perché no.

Le informazioni SMDSCONN rappresentano una connessione del gestore code a un dataset di messaggi condivisi. Come per il dataset del messaggio condiviso, viene identificato dal gestore code proprietario del dataset del messaggio condiviso (come specificato nella parola chiave SMDS per l'oggetto condiviso stesso) combinato con il nome CFSTRUCT.

Non esiste alcun parametro per identificare il gestore code di connessione poiché i comandi indirizzati a un determinato gestore code possono fare riferimento solo alle informazioni SMDSCONN per lo stesso gestore code.

Le voci di informazioni SMDSCONN vengono conservate nella memoria principale nel gestore code proprietario e vengono ricreate quando il gestore code viene riavviato. Tuttavia, se una connessione da un singolo gestore code è stata esplicitamente arrestata, queste informazioni vengono memorizzate anche come indicatore in un array di connessione nell'oggetto CFSTRUCT o SMDS corrispondente, in modo che persistano durante il riavvio di un gestore code.

Informazioni su stato e disponibilità

Le informazioni sullo stato indicano lo stato di una risorsa o di una connessione (ad esempio, se non è ancora in uso, se è in uso normale o se è necessario un ripristino). Viene generalmente descritto utilizzando la parola chiave STATUS. I valori possibili dipendono dal tipo di oggetto.

Le informazioni sullo stato vengono di norma aggiornate automaticamente, ad esempio quando viene rilevato un errore durante l'utilizzo della risorsa o della connessione. Tuttavia, in alcuni casi è possibile utilizzare un comando anche per aggiornare lo stato, per consentire i casi in cui non è possibile per un gestore code determinare automaticamente lo stato corretto.

Le informazioni sulla disponibilità indicano se è possibile utilizzare la risorsa o la connessione e di solito sono determinate principalmente dalle informazioni sullo stato. Per i tipi di risorsa o di connessione utilizzati nel supporto dataset di messaggi condivisi, sono implementati tre livelli di disponibilità:

Disponibile
Ciò significa che la risorsa è disponibile per essere utilizzata normalmente. Ciò non significa necessariamente che sia in uso al momento (che può essere determinato invece dal valore STATUS). Per un dataset, se richiede l'elaborazione del riavvio, ciò consente al gestore code proprietario di aprirlo, ma gli altri gestori code devono attendere che il dataset sia di nuovo nello stato ATTIVO.

Non disponibile a causa di un errore
Ciò significa che la risorsa è stata resa non disponibile automaticamente a causa di un errore e non è previsto che sia nuovamente disponibile fino a quando non viene eseguita una qualche forma di riparazione o di elaborazione di ripristino. Tuttavia, i tentativi di renderla nuovamente disponibile sono consentiti senza l'intervento dell'operatore. Tale tentativo può essere attivato anche da un comando per contrassegnare la risorsa come abilitata o da un comando che modifica lo stato in modo da indicare che l'elaborazione del ripristino è stata completata.

Il motivo per cui la risorsa è stata resa non disponibile è di solito ovvio dal valore STATUS correlato, ma in alcuni casi potrebbero esserci altri motivi per rendere la risorsa non disponibile, nel qual caso viene fornito un valore REASON separato per indicare il motivo.

Non disponibile a causa del comando dell'operatore
Ciò significa che l'accesso alla risorsa è stato esplicitamente disabilitato da un comando. Può essere reso disponibile solo utilizzando un comando per abilitarlo nuovamente.

Disponibilità SMDS

Per l'oggetto SMDS condiviso, la disponibilità è descritta dalla parola chiave ACCESS, con i valori possibili ENABLED, SUSPENDED e DISABLED.

La disponibilità può essere aggiornata utilizzando un comando **RESET SMDS** per l'oggetto condiviso pertinente da qualsiasi gestore code nel gruppo per impostare ACCESS (ENABLED) o ACCESS (DISABLED).

Se la disponibilità era in precedenza ACCESS (SUSPENDED), modificarla in ACCESS (ENABLED) attiverà un nuovo tentativo di utilizzare il data set del messaggio condiviso, ma se l'errore precedente è ancora presente, la disponibilità verrà reimpostata su ACCESS (SUSPENDED).

disponibilità SMDSCONN

Per una voce di informazioni SMDSCONN locale, la disponibilità è descritta dalla parola chiave AVAIL, con i valori possibili NORMAL, ERROR o STOPPED. La disponibilità può essere aggiornata utilizzando un comando **START SMDSCONN** o **STOP SMDSCONN** indirizzato a un gestore code specifico per abilitare o disabilitare la connessione.

Se la disponibilità era in precedenza AVAIL (ERROR), modificarla in AVAIL (NORMAL) attiverà un nuovo tentativo di utilizzo del dataset di messaggi condivisi, ma se l'errore precedente è ancora presente, la disponibilità verrà reimpostata su AVAIL (ERROR).

Disponibilità e stato condiviso del dataset del messaggio condiviso

La disponibilità di ciascun dataset di messaggi condivisi viene gestita all'interno del gruppo utilizzando le informazioni sullo stato condiviso, che è possibile visualizzare utilizzando il comando **DISPLAY CFSTATUS** con TYPE (SMDS). Visualizza le informazioni di stato per ogni gestore code che ha attivato un dataset per ciascuna struttura. Ogni dataset può trovarsi in uno dei seguenti stati:

Non trovato
Ciò significa che il dataset corrispondente non è stato ancora attivato. Questo stato viene visualizzato solo quando viene specificato un gestore code specifico, in quanto i dataset che non sono stati attivati vengono ignorati quando vengono selezionati tutti i gestori code.

NUOVO
Il dataset è in fase di apertura e inizializzazione per la prima volta, pronto per essere reso attivo.

ATTIVO
Ciò significa che il dataset è completamente disponibile e deve essere allocato e aperto da tutti i gestori code attivi per la struttura.

NON RIUSCITO
Ciò significa che il dataset non è disponibile (ad eccezione dell'elaborazione di ripristino) e deve essere chiuso e deallocato da tutti i gestori code.

In fase di recupero
Ciò significa che il ripristino del supporto (utilizzando RECOVER CFSTRUCT) è in corso per questo dataset.

Recuperato
Ciò indica che è stato immesso un comando per riportare un dataset non riuscito allo stato attivo, ma è richiesta un'ulteriore elaborazione di riavvio che non è ancora completa, in modo che il dataset possa essere aperto solo dal gestore code proprietario per l'elaborazione di riavvio.

VUOTO
Il dataset non contiene messaggi. Il dataset viene inserito in questo stato se viene chiuso normalmente dal gestore code proprietario, in un momento in cui non contiene alcun messaggio. Può anche essere impostato in stato EMPTY quando il contenuto del dataset precedente deve essere eliminato perché la struttura dell'applicazione è stata svuotata (utilizzando **RECOVER CFSTRUCT** con TYPE PURGE o, solo per una struttura non recuperabile, eliminando l'istanza precedente della struttura). La volta successiva che il dataset viene aperto dal relativo gestore code proprietario, la mappa di spazi viene reimpostata su vuoto e lo stato viene modificato in ATTIVO. Dal momento che il contenuto del dataset precedente non è più richiesto, un dataset in questo stato può essere sostituito con un dataset appena assegnato, ad esempio per modificare l'assegnazione dello spazio o spostarlo in un altro volume.

L'output del comando include la data e l'ora in cui è stata abilitata la registrazione di ripristino, se presente, e la data e l'ora in cui il dataset ha avuto esito negativo, se non è attualmente attivo.

Un dataset di messaggi condivisi può essere messo in uno stato NON RIUSCITO da un comando **RESET SMDS** o automaticamente quando viene rilevato uno dei seguenti tipi di errore:

- Il dataset non può essere assegnato o aperto dal gestore code proprietario.
- La convalida dell'intestazione del dataset ha esito negativo dopo che è stata aperta correttamente da qualsiasi gestore code.
- Un errore I/O permanente si verifica quando il gestore code proprietario sta leggendo o scrivendo i dati.
- Un errore I/O permanente si verifica quando un altro gestore code sta leggendo i dati da un dataset che ha completato correttamente l'elaborazione e la convalida aperte.

Quando un dataset si trova nello stato FAILED o INRECOVER, non è disponibile per l'uso normale, quindi se lo stato di disponibilità è ACCESS (ENABLED) viene modificato in ACCESS (SUSPENDED).

Se un dataset è stato inserito nello stato NON RIUSCITO ma non è richiesto alcun ripristino del supporto, ad esempio perché i dati erano ancora validi ma la periferica di memoria era temporaneamente non in linea, è possibile utilizzare il comando **RESET SMDS** per richiedere la modifica dello stato direttamente nello stato RECUPERATO.

Quando il dataset entra nello stato RECUPERATO, al termine dell'elaborazione del recupero o come risultato del comando **RESET SMDS** , è pronto per essere utilizzato di nuovo una volta completata l'elaborazione del riavvio. Se si trovava nello stato ACCESS (SUSPENDED), viene automaticamente riportato allo stato ACCESS (ENABLED), che consente al gestore code proprietario di eseguire il riavvio dell'elaborazione. Una volta completato il processo di riavvio, lo stato viene modificato in ATTIVO e tutti gli altri gestori code possono connettersi nuovamente al dataset.

Disponibilità e stato della connessione del dataset del messaggio condiviso

Ogni gestore code conserva lo stato locale e le informazioni sulla disponibilità per la connessione a ciascun dataset di messaggi condivisi di proprietà propria e di altri gestori code del gruppo. Queste informazioni possono essere visualizzate utilizzando il comando **DISPLAY SMDSCONN** .

Se non è in grado di accedere a un dataset di messaggi condivisi nello stato ATTIVO che appartiene ad un altro gestore code, contrassegna la connessione come non disponibile dal proprio punto di vista.

Se l'errore indica definitivamente un problema con il dataset stesso, anche il gestore code modifica automaticamente lo stato condiviso per indicare che il dataset è ora in uno stato NON RIUSCITO. Tuttavia, se l'errore può essere causato da un problema ambientale, ad esempio non si è autorizzati ad aprire il dataset, il gestore code emette messaggi di errore e considera il dataset come non disponibile, ma non modifica lo stato del dataset condiviso. Se l'errore ambientale risulta essere comunque un problema con il dataset (ad esempio, è stato assegnato su una periferica a cui non è possibile accedere da parte di alcuni dei gestori code), un operatore può utilizzare il comando RESET SMDS specificando STATUS (FAILED) per consentire il ripristino o la riparazione del dataset in base alle necessità.

Se non è stato possibile stabilire una connessione a un dataset di messaggi condivisi, ma il dataset sembra essere valido, è possibile attivare un nuovo tentativo di utilizzo immettendo un comando **START SMDSCONN** per il gestore code proprietario.

Se è necessario terminare temporaneamente la connessione tra un gestore code specifico e un dataset, ma il dataset non è danneggiato, è possibile chiudere e annullare l'assegnazione del dataset utilizzando il comando **STOP SMDSCONN** . Se il dataset è in uso, il gestore code lo chiuderà normalmente (anche se le richieste di dati in tale dataset verranno rifiutate con un codice di ritorno). Se si tratta del dataset di proprietà, il gestore code salverà la mappa di spazio durante l'elaborazione CLOSE, evitando la necessità di riavviare l'elaborazione.

Se un dataset deve essere portato fuori servizio temporaneamente da tutti i gestori code (ad esempio per spostarlo) ma non è danneggiato, è preferibile utilizzare **STOP SMDSCONN** per il dataset rilevante con l'opzione CMDSCOPE (*) per arrestare prima i gestori code che lo utilizzano, in quanto ciò eviterà la necessità di riavviare l'elaborazione quando il dataset viene ripristinato. Al contrario, se il dataset è contrassegnato come NON RIUSCITO, indica ai gestori code che devono smettere di utilizzarlo immediatamente, il che significa che la mappa di spazio non verrà salvata e dovrà essere rigenerata riavviando l'elaborazione.

L'accesso a tutti i dataset di messaggi condivisi precedentemente in stato ACCESS (SUSPENDED) verrà ritentato se il gestore code viene riavviato.

Registrazione di recupero della serie di dati dei messaggi condivisi

I messaggi condivisi persistenti vengono registrati per il ripristino dei supporti. Ciò significa che i messaggi possono essere recuperati dopo qualsiasi errore delle strutture CFS (coupling facility structure) o dei dataset di messaggi condivisi, purché i log di recupero siano ancora intatti. I messaggi persistenti possono anche essere ricreati dai log di ripristino su un altro sito per scopi di ripristino di emergenza.

Quando i dati del messaggio vengono scritti in un dataset di messaggi condivisi, ogni blocco scritto nel dataset viene registrato separatamente seguito dalla voce del messaggio (inclusa la mappa di dati) come scritto nella CF (Coupling Facility). Il processo di ripristino recupera sempre la struttura CFS (coupling facility structure), ma non è necessario ripristinare i singoli dataset di messaggi condivisi tranne quando lo stato del data set è NON RIUSCITO o quando lo stato è ATTIVO ma il record di intestazione del data set non è più valido, indicando che il data set è stato ricreato. Un dataset non è selezionato per il ripristino se il suo stato è ATTIVO e l'intestazione del dataset è ancora valida, né se il suo stato è VUOTO, che indica che non è stato memorizzato alcun messaggio al momento dell'errore.

Backup della serie di dati dei messaggi condivisi

Quando BACKUP CFSTRUCT viene utilizzato per eseguire un backup dei messaggi condivisi in una struttura dell'applicazione, viene eseguito contemporaneamente il backup di tutti i dati per i messaggi persistenti memorizzati nei dataset di messaggi condivisi, come per i messaggi persistenti condivisi precedentemente memorizzati nel DB.

Ripristino data set di messaggi condivisi

Se un dataset di messaggi condivisi è danneggiato o perso, è necessario inserirlo nello stato NON RIUSCITO per impedire ai gestori code di utilizzarlo fino a quando non viene riparato. Ciò di solito avviene

automaticamente, ma può essere eseguito anche utilizzando il comando **RESET SMDS** specificando STATUS (FAILED).

Se la serie di dati dei messaggi condivisi conteneva messaggi persistenti, è possibile recuperarli utilizzando il comando RECOVER CFSTRUCT. Questo comando ripristina prima tutti i dati dei messaggi persistenti per tale dataset di messaggi condivisi dal comando BACKUP CFSTRUCT più recente, quindi applica tutte le modifiche registrate da quel momento. Se non è stato eseguito alcun comando **BACKUP CFSTRUCT** dal momento in cui il dataset è stato attivato per la prima volta, viene reimpostato su vuoto e vengono applicate tutte le modifiche a partire dall'attivazione.

Se il contenuto CFSTRUCT e tutti i dataset di messaggi condivisi non sono disponibili, ad esempio in una situazione di ripristino di emergenza, è possibile recuperarli tutti in un singolo comando **RECOVER CFSTRUCT** .

Se un dataset di messaggi condivisi è danneggiato ma il ripristino non era attivo per CFSTRUCT o il log che contiene l'ultimo BACKUP CFSTRUCT non è disponibile o non è utilizzabile, i messaggi scaricati su tale dataset non possono essere recuperati. In questo caso, il comando **RECOVER CFSTRUCT** con il parametro TYPE (PURGE) può essere utilizzato per contrassegnare il dataset di messaggi condivisi come vuoto ed eliminare tutti i messaggi dalla struttura che avevano i dati memorizzati in tale dataset.

Quando si immette il comando **RECOVER CFSTRUCT** , lo stato del dataset del messaggio condiviso viene modificato da NON RIUSCITO a INRECOVER. Se il ripristino viene completato correttamente, lo stato viene automaticamente modificato in RECUPERATO, altrimenti ritorna su NON RIUSCITO.

Quando il dataset viene modificato nello stato RECUPERATO, questo indica al gestore code proprietario che può ora provare ad aprire il dataset ed eseguire il riavvio dell'elaborazione.

Punti di sincronizzazione e ripristino del dataset del messaggio condiviso

Il processo di ripristino del dataset del messaggio condiviso applica nuovamente le modifiche per tutti i record di log completi fino alla fine del log, indipendentemente dai punti di sincronizzazione.

Se le modifiche sono state apportate all'interno del punto di sincronizzazione, l'elaborazione di riavvio o ripristino per CFSTRUCT potrebbe causare il backout delle richieste non sottoposte a commit, quindi alcune delle modifiche ripristinate potrebbero non essere effettivamente utilizzate, ma non vi è alcun danno nel recuperarle comunque.

È anche possibile che un messaggio MQPUT senza commit sia stato scritto nella struttura ma i dati corrispondenti potrebbero non essere stati scritti nel dataset o nel log (poiché il completamento I/O viene forzato solo all'inizio dell'elaborazione del punto di sincronizzazione). Ciò è innocuo perché il riavvio dell'elaborazione eseguirà il backout della voce del messaggio nella struttura, quindi il fatto che si riferisca a dati non recuperati non è importante.

Elaborazione del riavvio della serie di dati dei messaggi condivisi

Se una connessione del gestore code a un CFSTRUCT termina normalmente, il gestore code scrive la mappa dello spazio di blocco libero per ciascun dataset di messaggi condivisi in un'area di checkpoint all'interno del dataset, appena prima della chiusura del dataset. La mappa di spazio può essere letta nuovamente al momento del riavvio della connessione, purché né CFSTRUCT né il data set di messaggi condivisi richiedano alcuna elaborazione di ripristino prima del successivo riavvio.

Tuttavia, se un gestore code termina in maniera anomala o la struttura o il dataset richiedono un'elaborazione di recupero, è richiesta un'ulteriore elaborazione per ricreare dinamicamente la mappa di spazio quando la connessione del gestore code alla struttura viene riavviata.

Se non è necessario ripristinare il data set, il riavvio del gestore code esegue semplicemente la scansione del contenuto corrente della struttura per individuare i riferimenti ai dati del messaggio di proprietà del gestore code corrente e contrassegna i blocchi di dati rilevanti come appartenenti alla mappa dello spazio. Gli altri gestori code possono continuare a utilizzare la struttura e leggere i dati di proprietà del gestore code in fase di riavvio mentre l'associazione spazio è in fase di ricostruzione.

Riavvio del dataset del messaggio condiviso dopo il recupero

Se è stato necessario ripristinare un data set di messaggi condivisi da un backup, tutti i messaggi non persistenti memorizzati nel data set andranno persi e se il data set è stato recuperato utilizzando TYPE (PURGE), tutti i messaggi memorizzati nel data set andranno persi. Fino al completamento del ripristino, il dataset verrà contrassegnato come NON RIUSCITO o INRECOVER, pertanto qualsiasi tentativo di leggere uno dei messaggi interessati da un altro gestore code restituisce un codice di errore che indica che il dataset è temporaneamente non disponibile.

Quando il dataset è stato ripristinato, lo stato viene modificato in RECUPERATO, che consente al gestore code proprietario di aprirlo per il riavvio dell'elaborazione, ma il dataset rimane non disponibile per altri gestori code. Il riavvio del gestore code esegue la scansione della struttura per ricreare la mappa di spazio per eventuali messaggi rimanenti. La scansione controlla anche i messaggi per i quali i dati sono stati persi e li elimina dalla struttura (o, se necessario, li contrassegna come persi, da eliminare in un secondo momento).

Lo stato del dataset viene automaticamente modificato da RECUPERATO a ATTIVO al completamento di questa scansione di riavvio, a quel punto altri gestori code possono iniziare a utilizzarlo di nuovo.

Informazioni sull'utilizzo del dataset del messaggio condiviso

Il comando DISPLAY USAGE ora mostra anche le informazioni sullo spazio del dataset di messaggi condivisi e sull'utilizzo del pool di buffer per tutti i dataset di messaggi condivisi attualmente aperti. Queste informazioni vengono visualizzate se viene specificata la nuova opzione TYPE (SMDS) o l'opzione esistente TYPE (ALL).

Considerazioni sulla capacità e sulle prestazioni dei dati dei messaggi condivisi

Monitoraggio dell'utilizzo del dataset

La percentuale di riempimento corrente di ciascun dataset di messaggi condivisi di proprietà può essere visualizzata dal comando **DISPLAY USAGE** con opzione **TYPE(SMDS)**.

Il gestore code espande automaticamente un dataset di messaggi condivisi quando raggiunge il 90% di riempimento, a condizione che l'opzione **DSEXPAND(YES)** sia attiva per la definizione SMDS. Ciò si applica quando l'opzione SMDS è impostata su **DSEXPAND(YES)** o quando l'opzione SMDS è impostata su **DSEXPAND(DEFAULT)** e l'opzione CFSTRUCT predefinita è impostata su **DSEXPAND(YES)**.

Se il tentativo di espansione non riesce perché non è stata specificata alcuna dimensione di allocazione secondaria quando è stato creato il data set (fornendo il messaggio IEC070I con codice di errore 203) il gestore code ripete la richiesta di espansione utilizzando un'assegnazione secondaria di sovrascrittura di circa il 20% della dimensione corrente.

Quando un dataset viene espanso, le nuove estensioni del dataset vengono formattate come parte dell'elaborazione di espansione, che può richiedere decine di secondi o anche minuti per estensioni molto grandi. Il nuovo spazio diventa disponibile una volta completata la formattazione e il catalogo è stato aggiornato per mostrare il nuovo intervallo di controllo utilizzato.

Se i nuovi messaggi vengono creati molto rapidamente, è possibile che il dataset esistente diventi pieno prima del completamento dell'elaborazione dell'espansione. In questo caso, qualsiasi richiesta che non può allocare spazio viene temporaneamente sospesa fino a quando il tentativo di espansione non viene completato e il nuovo spazio diventa disponibile per l'uso. Se l'espansione ha avuto esito positivo, la richiesta viene ritentata automaticamente.

Se un tentativo di espansione ha esito negativo, a causa di una mancanza di spazio disponibile o perché è già stato raggiunto il numero massimo di estensioni, viene emesso un messaggio che indica la causa dell'errore, l'opzione di sovrascrittura per l'SMDS interessato viene automaticamente modificata in **DSEXPAND(NO)** per impedire ulteriori tentativi di espansione. In questo caso, vi è il rischio che il dataset diventi pieno, nel qual caso potrebbe essere necessaria un'ulteriore azione come descritto in Il dataset diventa pieno.

Monitoraggio dell'utilizzo della struttura dell'applicazione

Il livello di uso di una struttura dell'applicazione può essere visualizzato utilizzando il comando MVS **DISPLAY XCF,STRUCTURE** specificando il nome completo della struttura dell'applicazione (incluso il prefisso del gruppo di condivisione code). Il IXC360I messaggio di risposta mostra l'utilizzo corrente di elementi e voci.

Quando l'utilizzo della struttura supera il valore **FULLTHRESHOLD** specificato nella politica CFRM, il sistema emette il messaggio IXC585E e può eseguire azioni **ALTER** automatiche, se specificate, che possono modificare il rapporto voce / elemento o aumentare la dimensione della struttura.

Ottimizzazione delle dimensioni del pool di buffer

Ogni buffer in un pool di buffer condiviso viene utilizzato per leggere o scrivere un intervallo contiguo di pagine per un messaggio fino alla dimensione del blocco logico. Se il messaggio si riversa in ulteriori blocchi, ogni intervallo di pagine in un blocco separato richiede un buffer separato.

I buffer che contengono i dati dei messaggi dopo un'operazione di scrittura o di lettura vengono conservati nella memoria e riutilizzati utilizzando uno schema di cache LRU (least - recently - used) in modo che una richiesta di leggere nuovamente gli stessi dati poco dopo non debba andare su disco. Ciò fornisce un'ottimizzazione significativa quando i messaggi condivisi vengono scritti e riletti subito dopo dalle applicazioni in esecuzione sullo stesso sistema. Se i messaggi di proprietà di un altro gestore code vengono sfogliati per scopi di selezione, quindi richiamati, ciò evita anche la necessità di rileggere il messaggio dal disco.

Ciò significa che il numero di buffer richiesti per ogni struttura dell'applicazione è uno per ogni richiesta API simultanea che legge o scrive messaggi di grandi dimensioni per tale struttura dell'applicazione più un certo numero di buffer aggiuntivi che verranno utilizzati per salvare i dati a cui è stato eseguito l'accesso di recente al fine di ottimizzare gli accessi di lettura successivi.

Per i pool di buffer condivisi, se non ci sono buffer sufficienti, le richieste API semplicemente attenderanno se un buffer non è immediatamente disponibile. Tuttavia, questa situazione deve essere evitata poiché può causare prestazioni significativamente degradate.

Le statistiche del comando **DISPLAY USAGE** per i pool di buffer condivisi mostrano se ci sono state attese di buffer nell'intervallo di statistiche corrente e mostrano anche il numero più basso di buffer liberi (o un valore negativo che indica il numero massimo di thread in attesa di un buffer in qualsiasi momento), il numero di buffer che hanno salvato i dati e la percentuale di volte in cui una richiesta di buffer ha trovato correttamente i dati salvati nella catena LRU ("hit LRU") invece di doverlo leggere ("Mancati riscontri LRU")[1].

- Se ci sono state delle attese, il numero di buffer deve essere aumentato.
- Se ci sono molti buffer inutilizzati, il numero di buffer può essere ridotto per rendere disponibile più memoria nella regione per altri scopi.
- Se ci sono molti buffer che contengono dati salvati, ma la proporzione di letture che erano risultati positivi rispetto a tali dati salvati è molto piccola, il numero di buffer può essere ridotto se la memoria può essere utilizzata meglio per altri scopi. Il numero di buffer, tuttavia, non dovrebbe essere ridotto di più del numero più basso di buffer liberi, poiché ciò potrebbe innescare attese, e dovrebbe essere preferibilmente abbastanza elevato che il conteggio di buffer liberi più basso sia di norma ben superiore a zero.

Eliminazione di dataset di messaggi condivisi

Il comando DELETE CFSTRUCT (che è consentito solo quando tutte le code condivise nella struttura sono vuote e chiuse) non elimina i dataset di messaggi condivisi, ma è possibile eliminarli nel modo consueto dopo il completamento di questo comando. Se la stessa serie di dati deve essere riutilizzata come una serie di dati di messaggi condivisi, è necessario riformattarla prima per reimpostarla allo stato vuoto.

[1] `(Hits / (Hits+Misses))* 100`

Situazioni di errore per dataset di messaggi condivisi

Esistono diverse situazioni di eccezione che possono verificarsi durante il normale utilizzo, anche quando non è presente alcun errore software o hardware.

Il dataset diventa pieno

Se un dataset diventa pieno ma non può essere espanso o se il tentativo di espansione ha esito negativo, le applicazioni che utilizzano il gestore code corrispondente per scrivere messaggi di grandi dimensioni nella struttura dell'applicazione corrispondente riceveranno l'errore 2192, `MQRC_STORAGE_MEDIUM_FULL` (noto anche come `MQRC_PAGET_FULL`).

Un dataset potrebbe diventare pieno a causa di un malfunzionamento nell'applicazione che dovrebbe elaborare i dati, causando l'accumulo di un ampio backlog di messaggi. In caso affermativo, l'ulteriore espansione del dataset sarà solo una soluzione temporanea ed è importante far ripartire l'applicazione di elaborazione il più presto possibile.

Se è possibile rendere disponibile più spazio, il comando **ALTER SMDS** può essere utilizzato per impostare **DSEXPAND(YES)** o **DSEXPAND(DEFAULT)** (supponendo che YES sia stato impostato o assunto come valore predefinito **DSEXPAND** per la definizione CFSTRUCT) per attivare un nuovo tentativo. Se il motivo dell'errore è stato tuttavia che è stato raggiunto il numero massimo di estensioni, il nuovo tentativo di espansione verrà rifiutato con un messaggio e **DSEXPAND(NO)** verrà impostato di nuovo. In questo caso, l'unico modo per espanderlo ulteriormente è riassegnarlo, il che significa renderlo temporaneamente non disponibile, come descritto di seguito.

Il dataset deve essere spostato o riallocato

Se un dataset deve essere spostato o espanso ma è altrimenti in uso normale, può essere temporaneamente fuori uso per consentirne lo spostamento o la riassegnazione. Tutte le richieste API che tentano di utilizzare il dataset mentre non è disponibile riceveranno il codice motivo `MQRC_DATA_SET_NOT_AVAILABLE.`

1. Utilizzare il comando **RESET SMDS** per contrassegnare il dataset come **ACCESS(DISABLED)**. Ciò causerà la chiusura normale e l'annullamento dell'assegnazione da parte di tutti i gestori code attualmente connessi.
2. Spostare o riassegnare il dataset come necessario, copiando il vecchio contenuto nel dataset appena assegnato, ad esempio utilizzando il comando AMS (Access Method Services) **REPRO** .

   Non tentare di preformattare il nuovo dataset prima di copiare i vecchi dati in esso, poiché ciò comporterebbe l'accodamento dei dati copiati alla fine del dataset formattato.
3. Utilizzare il comando **RESET SMDS** per contrassegnare nuovamente il data set come **ACCESS(ENABLED)** e riportarlo in uso.

   Se il vecchio contenuto è inferiore alla dimensione del nuovo dataset, il resto dello spazio verrà preformattato automaticamente quando viene aperto il nuovo dataset.

   Se il contenuto precedente era più grande della dimensione del nuovo dataset, il gestore code deve eseguire la scansione dei messaggi nella struttura CFS (coupling facility structure) e rigenerare la mappa di spazio per garantire che nessuno dei dati attivi sia stato perso. Se viene rilevato un riferimento a un blocco di dati esterno alle nuove estensioni, il dataset viene contrassegnato come **STATUS(FAILED)** e deve essere riparato sostituendo il dataset con una delle dimensioni corrette e copiando nuovamente il vecchio dataset oppure utilizzando **RECOVER CFSTRUCT** per ripristinare i messaggi persistenti.

La struttura CF (Coupling Facility) ha poco spazio

Se la struttura di Coupling Facility sta esaurendo lo spazio, causando il messaggio IXC585E, vale la pena controllare se le regole di offload sono state impostate per garantire che in questo caso venga eseguito l'offload della quantità massima di dati. In caso contrario, le regole di offload possono essere modificate utilizzando il comando **ALTER CFSTRUCT** .

Situazioni di errore per i dataset di messaggi condivisi

Ci sono una serie di problemi di cui essere consapevoli, che possono essere causati solo da errori e non si verificano in normali situazioni operative.

Impossibile aprire il dataset di proprietà

Se il gestore code proprietario di un dataset di messaggi condivisi non può assegnarlo o aprirlo oppure se gli attributi del dataset non sono supportati, il gestore code imposta un valore di stato **SMDSCONN** appropriato di **ALLOCFAIL** o **OPENFAIL** e imposta la disponibilità **SMDSCONN** su **AVAIL(ERROR)**. Imposta anche la disponibilità SMDS su **ACCESS(SUSPENDED)**. Una volta corretto l'errore, utilizzare il comando **RESET SMDS** per impostare **ACCESS(ENABLED)** per attivare un nuovo tentativo oppure immettere il comando **START SMDSCONN** per il gestore code proprietario.

Impossibile aprire il dataset di sola lettura

Se un gestore code non è in grado di allocare o aprire un dataset di messaggi condiviso di proprietà di un altro gestore code e contrassegnato come **STATUS(ACTIVE)**, presume che ciò sia probabilmente dovuto a un problema specifico con la connessione al dataset (rappresentato dall'oggetto **SMDSCONN**) piuttosto che a un problema con il dataset stesso.

Contrassegna **SMDSCONN** come **STATUS(ALLOCFAIL)** o **STATUS(OPENFAIL)** come appropriato e contrassegna la disponibilità **SMDSCONN** come AVAIL (ERROR) per evitare ulteriori tentativi di utilizzo.

Se il problema può essere risolto senza influenzare lo stato del dataset stesso, utilizzare il comando **START SMDSCONN** per attivare un nuovo tentativo.

Se il problema risulta essere un problema con il dataset stesso, il comando **RESET SMDS** può essere utilizzato per contrassegnare il dataset come **STATUS(FAILED)** fino a quando non viene ripristinato. Una volta ripristinato il dataset, l'azione di modifica dello stato in **STATUS(ACTIVE)** causerà la notifica agli altri gestori code. Se **SMDSCONN** è contrassegnato come **AVAIL(ERROR)**, verrà automaticamente modificato in **AVAIL(NORMAL)** per attivare un nuovo tentativo di aprire il dataset.

L'intestazione del dataset è danneggiata

Se il dataset è stato aperto correttamente, ma il formato delle informazioni di intestazione non è corretto, il gestore code chiude e annulla l'allocazione del dataset e imposta lo stato impostato su **STATUS(FAILED)** e la disponibilità su **ACCESS(SUSPENDED)**. Ciò consente di utilizzare **RECOVER CFSTRUCT** per ripristinare il contenuto.

Se l'errore si è verificato perché il dataset conteneva dati residui di un altro utilizzo e non era stato successivamente preformattato, preformattare il data set e utilizzare il comando **RESET SMDS** per modificare lo stato in **STATUS(RECOVERED)**.

Altrimenti, il dataset deve essere recuperato.

Il dataset è inaspettatamente vuoto

Se il gestore code apre un dataset contrassegnato come **STATUS(ACTIVE)** ma rileva che non è inizializzato o è appena preformattato ma altrimenti valido, il gestore code chiude e annulla l'assegnazione del dataset del messaggio condiviso, quindi imposta lo stato su **STATUS(FAILED)** e la disponibilità su **ACCESS(SUSPENDED)**.

Il dataset contiene errori I/O permanenti

Se un dataset presenta errori I/O permanenti dopo la corretta elaborazione di **OPEN** , è probabile che sia necessario eseguire il ripristino. Il gestore code contrassegnerà il data set come **STATUS(FAILED)** in modo che tutti i gestori code attualmente connessi verranno chiusi e deallocati.

Il dataset ha errori I/O recuperabili

Se si verificano problemi hardware con il data set, è possibile che ciò possa causare errori I/O recuperabili che non vengono riportati al gestore code ma che causano una significativa riduzione delle prestazioni e che indicano anche il rischio di errori I/O permanenti nel prossimo futuro.

In questo caso, il dataset potrebbe essere disattivato per il ripristino utilizzando il comando **RESET SMDS** per contrassegnarlo come **STATUS(FAILED)**. Ciò causerà la chiusura e l'annullamento

dell'assegnazione da parte di tutti i gestori code, quindi, ad esempio, potrebbe essere spostato su un nuovo volume prima di essere reso nuovamente disponibile.

Quando un dataset viene reso non disponibile in questo modo, la mappa di spazi non viene salvata, pertanto il processo di riavvio della connessione del gestore code dovrà eseguire la scansione della struttura CF per individuare i messaggi nel dataset e ricreare la mappa di spazi prima che il dataset possa essere reso nuovamente disponibile. In alternativa, se il dataset del messaggio condiviso è ancora utilizzabile, può essere reso non disponibile più delicatamente utilizzando il comando **RESET SMDS** per contrassegnare il dataset **ACCESS(DISABLED)** fino a quando non è pronto per essere reso nuovamente disponibile.

Il contenuto del data set non è corretto

Il gestore code non può rilevare direttamente che un dataset contiene dati non corretti o non è aggiornato, ad esempio perché un volume che include tale dataset ha dovuto essere ripristinato dai backup. Tuttavia, esegue controlli di integrità che rendono molto improbabile che tali errori possano causare la visualizzazione di dati di messaggi non corretti da parte dei programmi di applicazione.

Ai fini del controllo di integrità, ogni blocco di messaggi nel data set è preceduto da una copia dell'ID della voce Coupling Facility corrispondente, inclusa una data / ora univoca, che viene controllata ogni volta che il blocco di messaggi viene letto, prima che i dati del messaggio vengano passati al programma utente. Se il prefisso del blocco di messaggi non corrisponde all'ID voce (e la voce CF non è stata eliminata nel frattempo), si presume che il blocco di messaggi sia danneggiato e inutilizzabile.

Se il messaggio danneggiato era persistente, il dataset è contrassegnato come **STATUS(FAILED)** e il contenuto della struttura deve essere ripristinato utilizzando il comando **RECOVER CFSTRUCT** . Se il messaggio danneggiato era non persistente, non è possibile ripristinarlo, quindi viene emesso un messaggio diagnostico e la voce del messaggio CF corrispondente viene eliminata.

Se non è disponibile alcuna mappa di spazio salvata quando il dataset viene aperto, viene rigenerato eseguendo la scansione della struttura CFS per i riferimenti ai dati nel dataset. Durante questa scansione, il gestore code esegue una serie di azioni:

1. Il gestore code determina l'ubicazione del messaggio più recente (se presente) attualmente rimanente nel data set.
2. Il gestore code legge quindi tale messaggio dal dataset per garantire che il prefisso del blocco corrisponda all'ID della voce del messaggio

Queste azioni assicurano che il gestore code rilevi tutti i casi in cui il dataset è di livello inferiore e contrassegni il dataset come NON RIUSCITO. Questo controllo tollera tuttavia il caso in cui il dataset è stato ripristinato da una copia precedente e da allora non è stato aggiunto alcun nuovo messaggio oppure tutti i messaggi aggiunti da quando tale copia è stata successivamente letta ed eliminata.

Per proteggersi dai dati di livello inferiore nel caso in cui il dataset sia stato chiuso normalmente, il gestore code esegue una serie di azioni:

1. Il gestore code salva una copia della data / ora della mappa di spazi nell'oggetto SMDS in Db2 quando il data set viene chiuso normalmente.
2. Il gestore code verifica quindi che la data / ora della mappa di spazi sia la stessa, quando il dataset viene aperto di nuovo

Se la data / ora non corrisponde, ciò suggerisce che potrebbe essere stata utilizzata una copia di livello inferiore del dataset, in modo che il gestore code ignori la mappa dello spazio esistente e la ricostruisca, il che avrà esito positivo solo se non è stato effettivamente perso alcun dato del messaggio.

Nota: Questi controlli di integrità non garantiscono il rilevamento di un set di dati di livello inferiore o danneggiato in tutti i casi teoricamente possibili. Ad esempio, non rileveranno un caso in cui l'inizio di un blocco di messaggi è valido ma il resto dei dati è stato parzialmente sovrascritto.

Scenari di recupero per i dataset di messaggi condivisi

Questa sezione descrive gli scenari di recupero dei dataset di messaggi condivisi.

Ripristino del dataset in cui non è stato perso alcun dato

In alcuni casi, il contenuto corretto di un dataset non riuscito può essere ripristinato senza la necessità di un ripristino effettivo. Un esempio è quando un dataset contiene dati residui di un utilizzo precedente e non è stato preformattato di nuovo, che può essere corretto preformattandolo. Un altro caso è quando un dataset è stato spostato, ma si è verificato un errore nel processo di copia dei dati, che può essere corretto copiando di nuovo correttamente i dati.

In questi casi, il dataset corretto può essere reso nuovamente disponibile utilizzando il comando **RESET SMDS** per impostare **STATUS(RECOVERED)**. Se la disponibilità è attualmente **ACCESS(SUSPENDED)** , questa verrà automaticamente reimpostata su **ACCESS(ENABLED)**.

Quando il gestore code proprietario riceve una notifica che il dataset è stato ripristinato, esegue la scansione del contenuto della struttura per ricostruire la mappa di spazio, quindi modifica lo stato in **STATUS(ACTIVE)**. Gli altri gestori code possono quindi iniziare nuovamente la lettura del dataset.

Ripristino dataset con TYPE (NORMAL)

Se il contenuto di un dataset è stato perso, ma la struttura dell'applicazione è stata definita con **RECOVER(YES)** e sono disponibili i log di recupero appropriati, è possibile utilizzare il comando **RECOVER CFSTRUCT** per ripristinare i messaggi persistenti memorizzati nella struttura, inclusi i dati dei messaggi persistenti scaricati nei dataset di messaggi condivisi. Questo comando ripristina lo stato attuale utilizzando le informazioni registrate dal comando **BACKUP CFSTRUCT** più tutte le modifiche registrate ai messaggi persistenti dal momento del backup.

Il comando **RECOVER CFSTRUCT** recupera sempre tutti i messaggi persistenti nella struttura CFS (coupling facility structure) insieme ai dati dei messaggi scaricati memorizzati in Db2. Per i dati scaricati archiviati in dataset di messaggi condivisi, ogni dataset viene selezionato per l'elaborazione del ripristino solo se è già contrassegnato come **STATUS(FAILED)** o se risulta inaspettatamente vuoto o non valido quando viene aperto dall'elaborazione del ripristino. Qualsiasi data set di messaggi condiviso contrassegnato come attivo e che supera i controlli di convalida non deve essere ripristinato, poiché i dati del messaggio esistenti sono già corretti, ma l'intestazione viene aggiornata per indicare che qualsiasi mappa di spazio salvata dovrà essere ricostruita dopo il ripristino.

L'elaborazione del recupero è possibile solo quando la struttura è stata contrassegnata come non riuscita, poiché il contenuto completo della struttura deve essere ricostruito dall'elaborazione del recupero. Tuttavia, se almeno un dataset di messaggi condivisi è stato contrassegnato come non riuscito, il comando **RECOVER CFSTRUCT** contrassegnerà automaticamente la struttura come non riuscita, se necessario, per consentire l'elaborazione del ripristino.

Il ripristino può essere eseguito da qualsiasi gestore code nel gruppo di condivisione code, a condizione che gli sia stato concesso l'accesso in scrittura ai dataset pertinenti.

Solo i messaggi persistenti vengono sottoposti a backup e registrati, quindi la normale elaborazione di ripristino ripristinerà tutti i messaggi persistenti, ma causerà la perdita di tutti i messaggi non persistenti nella struttura.

Una volta completato il ripristino, qualsiasi dataset selezionato per il recupero viene automaticamente modificato in **STATUS(RECOVERED)**e, se la disponibilità era **ACCESS(SUSPENDED)** , viene modificato in **ACCESS(ENABLED)**. Il gestore code ricrea la mappatura dello spazio per ogni dataset eseguendo la scansione dei messaggi nella CF (Coupling Facility), quindi contrassegna il dataset come **STATUS(ACTIVE)** in modo che possa essere riutilizzato.

Ripristino dataset con TYPE (PURGE)

Per una struttura recuperabile, se il contenuto del dataset è stato perso, ma il ripristino non è possibile per qualche motivo, ad esempio perché i log di ripristino non sono disponibili o perché il ripristino richiede troppo tempo, il comando **RECOVER CFSTRUCT** può essere utilizzato con **TYPE(PURGE)** per riportare la struttura ad uno stato utilizzabile. Questa operazione reimposta la struttura sullo stato vuoto e contrassegna tutti i dataset associati come **STATUS(EMPTY)**.

Eliminazione della struttura dell'applicazione

Se una struttura dell'applicazione non ripristinabile viene eliminata utilizzando il comando MVS **SETXCF FORCE** , o come risultato di un errore della struttura, la volta successiva che la struttura

viene connessa, viene emesso il messaggio CSQE028I per indicare che la struttura è stata reimpostata e tutti i messaggi esistenti sono stati eliminati e tutti i dataset esistenti vengono reimpostati automaticamente su **STATUS(EMPTY)** . Questa azione rende nuovamente utilizzabile una struttura non recuperabile dopo la perdita di dati nella struttura o in uno dei dataset associati.

Se una struttura applicativa recuperabile viene eliminata, verrà trattata come se la struttura avesse avuto esito negativo.

Ripristino del dataset non riuscito

Se **RECOVER CFSTRUCT** non può essere completato per qualche motivo, ad esempio perché un dataset di log non è più disponibile o perché il gestore code è stato terminato mentre il ripristino era in corso, qualsiasi dataset per cui il ripristino è stato almeno avviato verrà contrassegnato nell'intestazione per mostrare che è stato eseguito un tentativo di ripristino parziale e il dataset verrà lasciato nello stato **STATUS(FAILED)** .

In questo caso, le opzioni sono di ripetere la richiesta di ripristino originale o di recuperare con **TYPE(PURGE)** , eliminando i dati esistenti.

Se viene effettuato un tentativo di contrassegnare il dataset come **STATUS(RECOVERED)** senza recuperarlo, la volta successiva che viene aperto il gestore code vedrà che l'intestazione indica un ripristino incompleto e lo contrassegnerà nuovamente come **STATUS(FAILED)** .

Ripristino di emergenza fuori sede

Per il ripristino di emergenza esterno al sito, i messaggi condivisi persistenti possono essere ricreati utilizzando solo i log e gli oggetti condivisi Db2 contenenti le definizioni CFSTRUCT e le informazioni sullo stato SMDS associate.

Dopo aver impostato le tabelle Db2 contenenti le definizioni, la struttura dell'applicazione e i dataset dei messaggi condivisi possono essere impostati come vuoti. Quando un gestore code si connette ad essi e rileva che sono inaspettatamente vuoti, li contrassegna come non riusciti, dopodiché è possibile utilizzare un singolo comando **RECOVER CFSTRUCT** per ripristinare tutti i messaggi persistenti per tutte le strutture interessate.

z/OS *Comandi correlati a SMDS*

Questo argomento descrive e fornisce l'accesso ai comandi relativi ai dataset di messaggi condivisi.

Visualizzare e modificare le opzioni **CFSTRUCT** relative all'offload di messaggi di grandi dimensioni (regole **OFFLOAD** e offload) e ai dataset di messaggi condivisi (**DSGROUP, DSBLOCK, DSBUFS, DSEXPAND)**:

- VISUALIZZA CFSTRUCT
- DEFINE CFSTRUCT
- MODIFICA CFSTRUCT
- DELETE CFSTRUCT

Visualizzare lo **CFSTRUCT** stato relativo all'offload di messaggi di grandi dimensioni **(OFFLDUSE)**:

- VISUALIZZA CFSTATUS

Visualizzare e modificare le opzioni del dataset di sovrascrittura (**DSEXPAND** e **DSBUFS**) per singoli gestori code:

- Visualizzazione SMDS
- SMDS ALTER

Visualizzare o modificare lo stato e la disponibilità dei dataset all'interno del gruppo di condivisione code:

- SMDS (DISPLAY CFSTATUS TYPE
- Reimposta SMDS

Visualizzare le informazioni sull'utilizzo dello spazio del data set SMDS e del buffer per un gestore code:

- SMDS (DISPLAY USAGE TYPE

Visualizzare o modificare lo stato e la disponibilità delle connessioni (**SMDSCONN**) ai dataset da un singolo gestore code:

- VISUALIZZA SMDSCONN
- START SMDSCONN
- STOP SMDSCONN

Backup e ripristino dei messaggi condivisi, inclusi i dati dei messaggi di grandi dimensioni in SMDS quando necessario:

- CFSTRUCT BACKUP
- RECUPERO CFSTRUCT

z/OS *Vantaggi dell'utilizzo delle code condivise*

La coda condivisa consente alle applicazioni IBM MQ di essere scalabili, altamente disponibili e consente l'implementazione del bilanciamento del carico di lavoro.

I vantaggi delle code condivise

L'architettura della coda condivisa, in cui i server clonati estraggono il lavoro da una singola coda condivisa, ha alcune proprietà utili:

- È scalabile, aggiungendo nuove istanze dell'applicazione server o aggiungendo una nuova immagine z/OS con un gestore code (nel gruppo di condivisione code) e una copia dell'applicazione.
- È altamente disponibile.
- Esegue naturalmente il bilanciamento del carico di lavoro *pull* , in base alla capacità di elaborazione disponibile di ciascun gestore code nel gruppo di condivisione code.

Utilizzo delle code condivise per l'alta disponibilità

I seguenti esempi illustrano come è possibile utilizzare una coda condivisa per aumentare la disponibilità dell'applicazione.

Considerare uno scenario IBM MQ in cui le applicazioni client in esecuzione nella rete desiderano effettuare richieste di applicazioni server in esecuzione su z/OS. L'applicazione client costruisce un messaggio di richiesta e lo inserisce in una coda di richiesta. Il client attende quindi una risposta dal server, inviata alla coda di risposta indicata nel descrittore del messaggio del messaggio di richiesta.

IBM MQ gestisce il trasporto del messaggio di richiesta dalla macchina client alla coda di input del server su z/OS e della risposta dal server al client. Definendo la coda di input del server come una coda condivisa, qualsiasi messaggio inserito nella coda può essere richiamato su qualsiasi gestore code nel gruppo di condivisione code. Ciò significa che è possibile configurare un gestore code su ciascuna immagine z/OS nel sysplex e, collegandoli tutti allo stesso gruppo di condivisione code, ognuno di essi può accedere ai messaggi sulla coda di input del server.

I messaggi sulla coda di input del server sono ancora disponibili, anche se uno dei gestori code termina in modo anomalo o se è necessario arrestarlo per motivi di gestione. È possibile portare un'intera immagine z/OS offline e i messaggi saranno ancora disponibili.

Per usufruire di questa disponibilità di messaggi su una coda condivisa, eseguire un'istanza dell'applicazione server su ciascuna immagine z/OS nel sysplex per fornire una maggiore capacità e disponibilità dell'applicazione server, come mostrato in Figura 60 a pagina 192.

Un'istanza dell'applicazione server richiama un messaggio di richiesta dalla coda condivisa e, in base al contenuto, ne esegue l'elaborazione, producendo un risultato che viene restituito al client come un messaggio IBM MQ . Il messaggio di risposta è destinato al gestore code reply - to e reply - to denominato nel descrittore del messaggio di richiesta.

Esistono diverse opzioni che è possibile utilizzare per configurare il percorso di ritorno. Per ulteriori informazioni su tali opzioni, consultare “Accodamento distribuito e gruppi di condivisione code” a pagina 212.

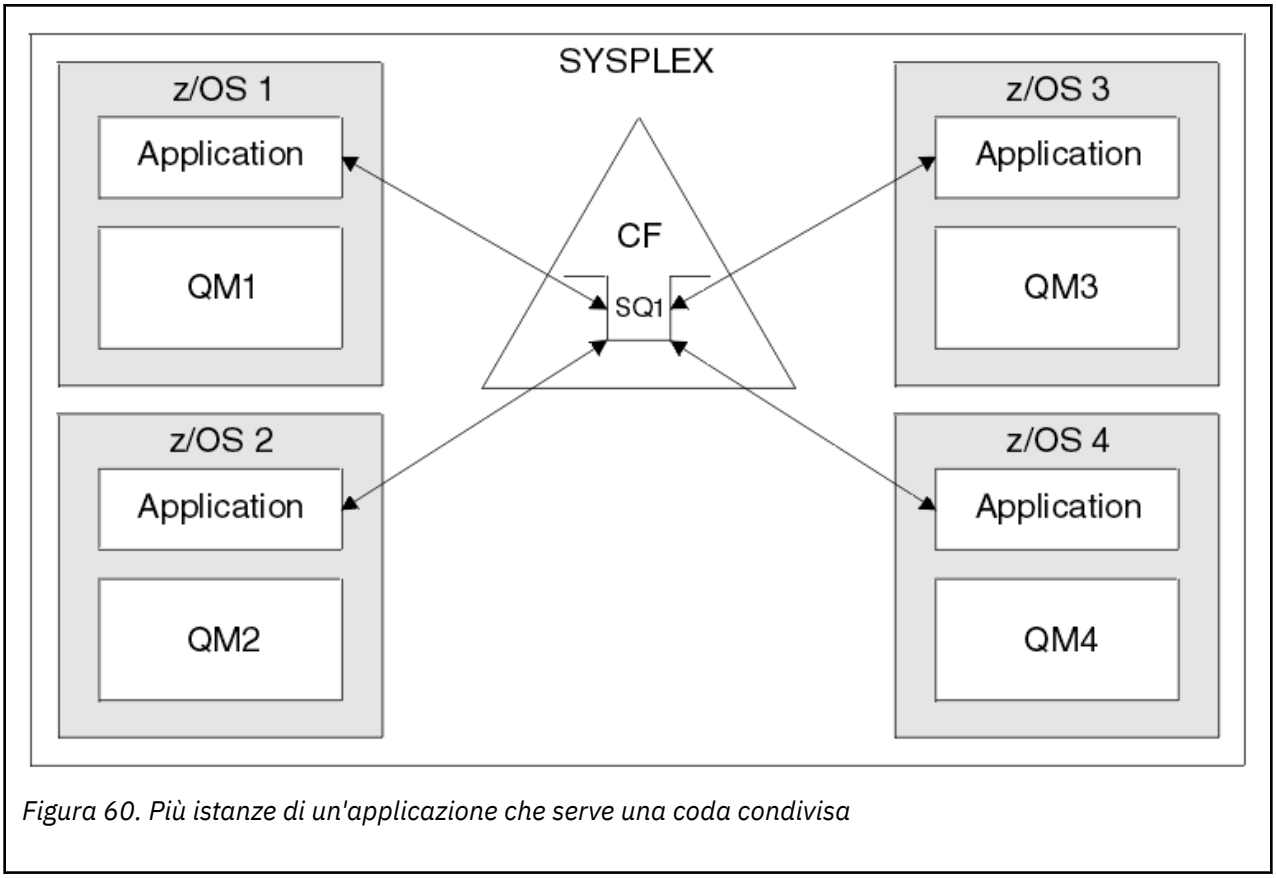


Figura 60. Più istanze di un'applicazione che serve una coda condivisa

Ripristino peer

Per migliorare ulteriormente la disponibilità dei messaggi in un gruppo di condivisione code, IBM MQ rileva se un altro gestore code nel gruppo si disconnette in modo anomalo dalla CF (Coupling Facility) e completa le unità di lavoro per tale gestore code che sono ancora in sospeso, laddove possibile. Questa funzione è nota come *ripristino peer.*

Si supponga che un gestore code termini in modo anomalo in un momento in cui un'applicazione ha richiamato un messaggio di richiesta da una coda nel punto di sincronizzazione, ma non ha ancora inserito il messaggio di risposta o eseguito il commit dell'unità di lavoro. Un altro gestore code nel gruppo di condivisione code rileva l'errore ed esegue il backout delle unità di lavoro in esecuzione sul gestore code in errore. Ciò significa che il messaggio di richiesta viene reinserito nella coda di richiesta ed è disponibile per l'elaborazione da parte di una delle altre istanze del server, senza attendere il riavvio del gestore code non riuscito.

Se IBM MQ non è in grado di risolvere automaticamente un'unità di lavoro, è possibile risolvere manualmente la parte condivisa per abilitare un altro gestore code nel gruppo di condivisione code a continuare l'elaborazione di tale lavoro.

z/OS *Utilizzo della memoria della classe storage con code condivise*

L'utilizzo di SCM (storage class memory) può essere vantaggioso se utilizzato con code condivise IBM MQ for z/OS .

Importante: IBM z16 è pianificato per essere l'ultima generazione di IBM Z ® per supportare l'utilizzo di memoria flash virtuale (nota anche come SCM (Storage Class Memory) o SCM) per le immagini della CF

(Coupling Facility). Per ulteriori informazioni, consultare IBM Z e IBM LinuxONE 4Q 2023 Statement of Direction.

In alternativa, è necessario utilizzare strutture più grandi o scaricare i messaggi su SMDS.

Le macchine z13, zEC12e zBC12 consentono l'installazione di schede Flash Express. Queste schede contengono SSD (solid - state drive) flash. Dopo l'installazione, la memoria flash dalle schede può essere assegnata a una o più LPAR in cui è generalmente nota come SCM.

SCM si colloca tra storage reale e DASD (direct access storage device) in termini di latenza I/O e costi. Poiché SCM non ha parti mobili, presenta latenze I/O molto più basse rispetto a DASD.

SCM è anche molto più economico di stoccaggio reale. Di conseguenza, è possibile installare una grande quantità di memoria per un costo relativamente basso; ad esempio, una coppia di schede Flash Express contiene 1424 GB di memoria utilizzabile.

Queste caratteristiche indicano che SCM è utile quando una grande quantità di dati deve essere presa dalla memoria reale in un breve periodo di tempo, perché tali dati possono essere scritti su SCM molto più velocemente di quanto possa essere scritto su DASD. Questo punto specifico può essere molto utile quando si utilizzano strutture di elenco CF (coupling facility) contenenti code condivise IBM MQ .

Perché le strutture dell'elenco si riempiono

Quando una struttura CF viene definita, viene configurata con un attributo SIZE che descrive la dimensione massima della struttura. Poiché le strutture CF sono sempre residenti in modo permanente nella memoria reale, la somma degli attributi SIZE delle strutture definite in una CF deve essere inferiore alla quantità di memoria reale assegnata alla CF.

Di conseguenza, c'è una pressione costante per mantenere il valore SIZE di una data struttura al valore minimo possibile, in modo che più strutture possano adattarsi alla CF. Tuttavia, garantire che le strutture siano abbastanza grandi per raggiungere il loro obiettivo può provocare una pressione conflittuale, perché rendere una struttura troppo piccola significa che potrebbe riempirsi, interrompendo le applicazioni o i sottosistemi che ne fanno uso.

C'è una forte necessità di dimensionare con precisione una struttura in base al suo utilizzo previsto. Tuttavia, questa attività è difficile da eseguire perché i carichi di lavoro possono cambiare nel tempo e non è facile contarne le fluttuazioni.

Le code condivise IBM MQ utilizzano le strutture elenco CF per memorizzare i messaggi. IBM MQ richiama le strutture CF, che contengono messaggi e strutture dell'applicazione.

Alle strutture dell'applicazione si fa riferimento utilizzando le informazioni memorizzate negli oggetti CFSTRUCT IBM MQ . Quando un messaggio inferiore a 63 KB viene inserito in una coda condivisa, il messaggio viene memorizzato interamente in una struttura dell'applicazione come una singola voce di elenco e zero o più elementi di elenco.

Poiché le code condivise IBM MQ utilizzano strutture di elenco, le pressioni descritte influiscono anche sulle code condivise. In questo caso, il numero massimo di messaggi che è possibile memorizzare in una coda condivisa è una funzione di:

- Dimensione dei messaggi sulla coda
- Dimensione massima della struttura
- Numero di voci ed elementi disponibili nella struttura

Poiché fino a 512 code condivise possono utilizzare la stessa struttura e competere in modo efficace per voci ed elementi, ciò complica ulteriormente le cose

Le code IBM MQ sono utilizzate per il trasferimento dei dati tra le applicazioni, quindi una situazione comune è un'applicazione che inserisce i messaggi in una coda, quando l'applicazione partner, che dovrebbe ricevere tali messaggi, non è in esecuzione.

Quando ciò si verifica, il numero di messaggi sulla coda aumenta nel corso del tempo fino a quando si verificano una o più delle seguenti situazioni:

- L'applicazione di inserimento smette di inserire messaggi.
- L'applicazione di richiamo inizia a ricevere messaggi.
- I messaggi esistenti sulla coda iniziano a scadere e vengono rimossi dalla coda.
- La coda raggiunge la profondità massima nel qual caso viene restituito un codice motivo MQRC_Q_FULL all'applicazione di inserimento.
- La struttura che contiene la coda condivisa raggiunge la sua dimensione massima oppure la CF che contiene la struttura esaurisce la memoria disponibile. In entrambi i casi, un codice motivo MQRC_STORAGE_MEDIUM_FULL viene restituito all'applicazione di inserimento.

Negli ultimi tre casi la coda è piena. A questo punto l'applicazione di inserimento ha un problema perché non c'è nessun posto dove i suoi messaggi possono andare. L'applicazione di inserimento in genere risolve questo problema utilizzando una o più delle seguenti soluzioni:

- Ripetere ripetutamente l'immissione del messaggio, facoltativamente con un ritardo tra i tentativi.
- Inserire i messaggi in un altro punto, ad esempio un database o un file. È possibile accedere ai messaggi in un secondo momento e inserirlo nella coda come di consueto.
- Eliminare il messaggio se non è persistente.

Tuttavia, per alcune classi di applicazioni, ad esempio quelle con un grande volume di messaggi in entrata o senza accesso a un file system, queste soluzioni non sono pratiche. C'è una reale necessità di garantire che le code non si riempiano mai, o siano estremamente improbabili, in primo luogo, e questo è particolarmente pertinente per le code condivise.

Regole SMDS e offload

Le regole di offload introdotte in IBM WebSphere MQ 7.1 forniscono un modo per ridurre la probabilità che una struttura dell'applicazione si riempa.

Ciascuna struttura dell'applicazione ha tre regole associate, specificate utilizzando tre coppie di parole chiave:

- OFFLD1SZ e OFFLD1TH
- OFFLD2SZ e OFFLD2TH
- OFFLD3SZ e OFFLD3TH

Ogni regola specifica le condizioni che devono essere soddisfatte per l'offload dei dati del messaggio nel meccanismo di archiviazione associato alla struttura dell'applicazione. Attualmente sono disponibili due tipi di meccanismi di archiviazione:

- Db2
- Dataset lineari VSAM (Group of Virtual Storage Access Method), che IBM MQ richiama un SMDS (shared message data set).

Il seguente esempio mostra il comando MQSC per creare una struttura dell'applicazione denominata LIST1, utilizzando il comando DEFINE CFSTRUCT .

Questa struttura ha le regole di offload predefinite e utilizza SMDS come meccanismo di offload. Ciò significa che quando la struttura è piena al 70% (OFFLD1TH), tutti i messaggi di dimensioni pari o superiori a 32 KB (OFFLD1SZ) vengono scaricati su SMDS.

Allo stesso modo, quando la struttura è piena all ' 80% (OFFLD2TH), vengono scaricati tutti i messaggi di dimensioni pari o superiori a 4 KB (OFFLD2SZ). Quando la struttura è piena al 90% (OFFLD3TH) tutti i messaggi (OFFLD3SZ) vengono scaricati.

```
DEFINE CFSTRUCT(LIST1)
CFLEVEL(5)
OFFLOAD(SMDS)
OFFLD1SZ(32K) OFFLD1TH(70)
OFFLD2SZ(4K) OFFLD2TH(80)
OFFLD3SZ(0K) OFFLD3TH(90)
```

Un messaggio scaricato viene memorizzato nel supporto di scaricamento e un puntatore al messaggio viene memorizzato nella struttura. Mentre le regole di offload riducono la probabilità che la struttura si riempia, inserendo meno dati di messaggio nella struttura quando esaurisce la memoria, alcuni dati vengono ancora scritti nella struttura per ogni messaggio. Vale a dire, il puntatore al messaggio scaricato.

Inoltre, le regole di offload vengono con un costo delle prestazioni. La scrittura di un messaggio in una struttura è relativamente rapida ed è in gran parte dominata dal tempo impiegato per inviare la richiesta di scrittura alla CF. La scrittura effettiva nella struttura è veloce, e avviene a velocità di archiviazione reali.

La scrittura di un messaggio in SMDS è molto più lenta perché include la scrittura nella struttura per il puntatore del messaggio e la scrittura dei dati del messaggio in SMDS. Questa seconda operazione di scrittura viene eseguita alla velocità DASD e ha il potenziale per aggiungere latenza. Se Db2 viene utilizzato come meccanismo di offload, il costo delle prestazioni è molto maggiore.

z/OS *Come funziona la memoria della classe di archiviazione con IBM MQ for z/OS*

Una panoramica dell'utilizzo di SCM (storage class memory) con code condivise IBM MQ for z/OS .

Importante: IBM z16 è pianificato per essere l'ultima generazione di IBM Z ® per supportare l'utilizzo di memoria flash virtuale (nota anche come SCM (Storage Class Memory) o SCM) per le immagini della CF (Coupling Facility). Per ulteriori informazioni, consultare IBM Z e IBM LinuxONE 4Q 2023 Statement of Direction.

In alternativa, è necessario utilizzare strutture più grandi o scaricare i messaggi su SMDS.

Una CF (coupling facility) che si trova a CFLEVEL 19, o superiore, può avere SCM assegnato ad essa. Le strutture definite in tale CF possono quindi essere configurate in modo da utilizzare SCM per ridurre le possibilità di riempimento delle strutture (noto come condizione completa della struttura). Quando una struttura configurata per utilizzare SCM si riempie oltre un punto determinato dal sistema, la CF inizia a spostare i dati dalla struttura in SCM, liberando spazio nella struttura per nuovi dati.

Nota: Poiché SCM stesso può riempirsi, l'assegnazione di SCM a una struttura riduce solo la probabilità di una condizione completa della struttura, ma non elimina completamente la probabilità che si verifichi una condizione.

Una struttura viene configurata per utilizzare SCM specificando sia le parole chiave **SCMALGORITHM** che **SCMMAXSIZE** nella politica CFRM (coupling facility resource manager), contenente la definizione di tale struttura.

Notare che una volta specificate queste parole chiave e applicata la politica CFRM, la struttura deve essere ricreata o deallocata in modo che possano diventare effettive.

SCMALGORITHM Parola chiave

Poiché la velocità di input / output di SCM è più lenta di quella della memoria reale, la CF utilizza un algoritmo adattato all'uso previsto della struttura per ridurre l'impatto della scrittura o della lettura da SCM.

L'algoritmo viene configurato dalla parola chiave **SCMALGORITHM** nella politica CFRM per la struttura, utilizzando il valore *KEYPRIORITY1* . Tenere presente che è necessario utilizzare il valore *KEYPRIORITY1* solo con le strutture elenco utilizzate dalle code condivise IBM MQ .

L'algoritmo *KEYPRIORITY1* funziona supponendo che la maggior parte delle applicazioni riceva i messaggi da una coda condivisa in ordine di priorità; ossia, quando un'applicazione riceve un messaggio, riceve il messaggio più vecchio con la priorità più alta.

Quando una struttura inizia a riempirsi oltre la soglia definita dal sistema del 90%, la CF avvia la migrazione asincrona dei messaggi che hanno meno probabilità di essere ricevuti successivamente. Si tratta di messaggi con priorità più basse che sono stati inseriti più recentemente nella coda.

Questa migrazione asincrona dei messaggi dalla struttura a SCM è nota come "pre - staging".

Il pre - staging riduce il costo delle prestazioni dell'utilizzo di SCM in quanto riduce la probabilità che un'applicazione venga bloccata durante il verificarsi di input / output sincrono su SCM.

Oltre alla pre - preparazione, l'algoritmo *KEYPRIORITY1* riporta in modo asincrono i messaggi da SCM e nella struttura quando è disponibile spazio libero sufficiente. Per l'algoritmo *KEYPRIORITY1* , ciò significa che quando la struttura è inferiore o uguale al 70% di riempimento.

L'atto di portare i messaggi da SCM nella struttura è noto come "precaricamento".

Il precaricamento riduce la possibilità che un'applicazione tenti di ottenere un messaggio che è stato pre - preparato su SCM e debba attendere mentre la CF riporta in modo sincrono il messaggio nella struttura.

SCMMAXSIZE Parola chiave

La parola chiave **SCMMAXSIZE** definisce la quantità massima di SCM che può essere utilizzata da una struttura. Poiché SCM viene assegnato alla struttura dalla CF quando è richiesto, è possibile specificare un **SCMMAXSIZE** maggiore della quantità totale di SCM liberi disponibili. Questo è noto come "over-commit".

Importante: Non eseguire mai il commit eccessivo di SCM. Se lo si fa, le applicazioni che si basano su di esso non otterranno il comportamento che si aspettano. Ad esempio, le applicazioni IBM MQ che utilizzano code condivise potrebbero ricevere codici di errore MQRC_STORAGE_MEDIUM_FULL non previsti.

La CF utilizza varie strutture di dati per tracciare il relativo utilizzo di SCM. Queste strutture di dati risiedono nella memoria reale assegnata alla CF e, di conseguenza, riducono la quantità di memoria reale che può essere utilizzata dalle strutture. La memoria utilizzata da queste strutture dati è nota come "spazio aumentato".

Quando una struttura è configurata con SCM, una piccola quantità di memoria reale viene assegnata dalla CF alla struttura nota come spazio aumentato fisso. Questo viene assegnato anche se la struttura non utilizza mai alcun SCM. Poiché i dati della struttura vengono memorizzati in SCM, lo spazio incrementato dinamico supplementare verrà allocato dalla memoria reale di riserva nella CF.

Quando i dati vengono rimossi da SCM, lo spazio convertito dinamico viene restituito alla CF. Lo spazio incrementato, fisso o dinamico, non viene mai preso dalla memoria reale assegnata a una struttura.

Oltre alla memoria aumentata, quando una struttura è configurata per utilizzare SCM, la quantità di memoria di controllo utilizzata da tale struttura aumenta. Ciò significa che una struttura di elenco configurata con SCM può contenere meno voci ed elementi di una struttura della stessa dimensione senza che SCM sia configurato.

Per comprendere l'impatto di SCM sulle strutture nuove o esistenti, utilizzare lo strumento CFSizer .

Un ultimo punto importante da notare è che dopo che i dati sono stati spostati dalla struttura in SCM, e lo spazio incrementato dinamico è stato utilizzato, la struttura non può essere modificata manualmente o automaticamente.

Ovvero, la quantità di memoria assegnata alla struttura non può essere aumentata o diminuita, il rapporto voce - elemento utilizzato dalla struttura non può essere modificato e così via. Per rendere la struttura nuovamente modificabile, la struttura non deve avere alcun dato memorizzato in SCM e non deve utilizzare la memoria aumentata dinamica.

z/OS *Perché utilizzare SCM*

L'archiviazione di emergenza e le prestazioni migliorate sono due casi di utilizzo per l'utilizzo di SCM con IBM MQ for z/OS.

Importante: IBM z16 è pianificato per essere l'ultima generazione di IBM Z ® per supportare l'utilizzo di memoria flash virtuale (nota anche come SCM (Storage Class Memory) o SCM) per le immagini della CF (Coupling Facility). Per ulteriori informazioni, consultare IBM Z e IBM LinuxONE 4Q 2023 Statement of Direction.

In alternativa, è necessario utilizzare strutture più grandi o scaricare i messaggi su SMDS.

Questa sezione introduce la teoria dietro i due possibili scenari. Per ulteriori dettagli su come impostare gli scenari, consultare:

- “Memoria di emergenza - configurazione di base” a pagina 200

- “Prestazioni migliorate - configurazione di base” a pagina 206

Importante: L'utilizzo di SCM con strutture CF non dipende da alcuna versione specifica di IBM MQ. Tuttavia, lo scenario di archiviazione di emergenza funziona solo con IBM WebSphere MQ 7.1 e versioni successive, poiché richiede SMDS e regole di offload.

Stoccaggio di emergenza

L'SMDS e lo scaricamento dei messaggi possono essere utilizzati insieme a SCM per ridurre la probabilità che un codice motivo MQRC_STORAGE_MEDIUM_FULL venga restituito a un'applicazione IBM MQ durante un'interruzione estesa.

Panoramica

Una singola coda condivisa è configurata su una struttura dell'applicazione. L'applicazione di inserimento inserisce i messaggi nella coda condivisa; l'applicazione di richiamo richiama i messaggi dalla coda condivisa.

Durante l'esecuzione normale, la profondità della coda è prevista prossima a zero, ma un requisito di business indica che il sistema deve essere in grado di tollerare un'interruzione di due ore dell'applicazione di richiamo. Ciò significa che la coda condivisa deve essere in grado di contenere due ore di messaggi dall'applicazione di inserimento.

Attualmente, questo processo si ottiene utilizzando le regole di offload predefinite e SMDS, in modo che la dimensione della struttura sia ridotta al minimo, riducendo al contempo il costo delle prestazioni associato all'offload.

La frequenza dei messaggi inviati alla coda condivisa dovrebbe raddoppiare nel breve e medio termine. Sebbene il requisito che il sistema sia in grado di tollerare un'interruzione di due ore esista ancora, nella CF non è disponibile sufficiente memoria reale per raddoppiare la dimensione della struttura.

Poiché la CF, che contiene la struttura dell'applicazione, risiede su una macchina zEC12 , esiste la possibilità di associare un SCM sufficiente alla struttura per memorizzare un numero sufficiente di messaggi in modo da poter tollerare un'interruzione di due ore.

Considerare ciò che accade in un periodo di tempo:

1. Inizialmente, il sistema è nello stato stazionario. Sia l'applicazione di inserimento che quella di richiamo sono in esecuzione normalmente e la profondità della coda è vicina o uguale a zero. Il risultato è che la struttura dell'applicazione è in gran parte vuota.
2. In un determinato momento, l'applicazione di richiamo subisce un malfunzionamento imprevisto e si arresta. L'applicazione di inserimento continua a inserire i messaggi nella coda e la struttura dell'applicazione inizia a riempirsi.
3. Una volta che la struttura raggiunge il 70% di riempimento, le condizioni della prima regola offload vengono soddisfatte e tutti i messaggi con una dimensione maggiore o uguale a 32 KB vengono scaricati su SMDS.

   Consultare “Regole SMDS e offload” a pagina 194 per una panoramica delle regole di offload.
4. Man mano che i messaggi continuano ad essere inseriti nella coda condivisa, la struttura continua a riempirsi (a causa dei dati del messaggio memorizzati nella struttura o come risultato dei puntatori ai messaggi scaricati memorizzati nella struttura).

   Quando la struttura raggiunge l'80% di riempimento, la seconda regola di offload inizia ad essere applicata e i messaggi che sono 4 KB o superiori vengono scaricati su SMDS.
5. Quando la struttura è piena al 90%, tutti i messaggi vengono scaricati in SMDS e solo i puntatori dei messaggi vengono inseriti nella struttura.

   In questo periodo, l'algoritmo di pre - staging inizia ad essere eseguito e inizia a spostare i dati dalla struttura in SCM. Supponendo che tutti i messaggi nella coda abbiano la stessa priorità, i messaggi più recenti vengono pre - inseriti.

   Poiché tutti i messaggi vengono scaricati in SMDS, i dati spostati in SCM non sono dati di messaggi effettivi, ma i puntatori ai messaggi in SMDS.

Di conseguenza, il numero di messaggi che possono essere memorizzati sulla combinazione della struttura e l'SCM e l'SMDS associati alla struttura, è molto grande.

Prestazioni: Durante questa fase dell'interruzione, l'applicazione di inserimento può subire un grado di riduzione delle prestazioni a causa della necessità di scrivere in SMDS. In questo caso, l'uso di SCM non dovrebbe essere un fattore limitante per l'applicazione in termini di prestazioni. SCM fornisce ulteriore spazio per evitare che la struttura si riempa.

6. Alla fine l'applicazione di acquisizione è di nuovo disponibile e l'interruzione è terminata.

   Tuttavia, SCM è ancora utilizzato dalla struttura. L'applicazione di richiamo inizia a leggere i messaggi dalla coda, ottenendo prima i messaggi con priorità più alta e meno recenti.

   Poiché questi messaggi sono stati scritti prima che la struttura iniziasse a riempirsi, essi escono interamente dalla parte di memoria reale della struttura.
7. Quando la struttura inizia a svuotarsi, scende al di sotto della soglia in cui è attivo il pre - staging e quindi si arresta.
8. L'utilizzo della struttura si riduce al di sotto del punto in cui le regole di offload hanno effetto, quindi i messaggi non vengono più scaricati su SMDS a meno che non siano più di 63 KB.

   In questo momento, l'algoritmo di pre - fetch inizia a spostare i dati da SCM nella struttura. Poiché l'applicazione di richiamo riceve i messaggi dalla coda nell'ordine previsto dagli algoritmi SCM, i messaggi vengono introdotti prima che l'applicazione di richiamo ne abbia bisogno.

   Il risultato è che l'applicazione di richiamo non deve mai aspettare che i messaggi vengano portati in modo sincrono da SCM.
9. Man mano che l'applicazione di richiamo continua a spostare verso il basso la coda, inizia a richiamare i messaggi scaricati su SMDS.
10. Infine, il sistema è di nuovo in stato stazionario. Nessun messaggio viene memorizzato in SCM o SMDS e la profondità della coda è prossima a zero.

Migliorate performance

Questo scenario descrive l'utilizzo di SCM per aumentare il numero di messaggi che possono essere archiviati su una coda condivisa senza incorrere nel costo delle prestazioni dell'utilizzo di SMDS.

Descrizione

Per questo scenario, un'applicazione di inserimento e richiamo comunica attraverso una coda condivisa memorizzata nella struttura dell'applicazione.

L'applicazione di inserimento tende ad essere eseguita in burst, quando inserisce un numero elevato di messaggi in un breve periodo di tempo. Quindi, in un periodo di tempo prolungato, non produce alcun messaggio.

L'applicazione di richiamo elabora sequenzialmente ogni messaggio ed esegue elaborazioni complesse su ciascuno di essi. Di conseguenza, la maggior parte delle volte la profondità della coda è zero, ad eccezione di quando l'applicazione di inserimento inizia l'esecuzione, dove la profondità della coda inizia ad aumentare man mano che i messaggi vengono inseriti più velocemente di quanto non vengano ricevuti.

La profondità della coda aumenta finché l'applicazione di inserimento non si arresta e l'applicazione di richiamo ha tempo sufficiente per elaborare tutti i messaggi sulla coda.

Note:

1. In questo scenario, il fattore chiave è la prestazione. I messaggi inviati alla coda sono sempre inferiori a 63 KB e quindi non devono mai essere scaricati su SMDS.
2. La struttura dell'applicazione è stata ridimensionata in modo che sia abbastanza grande da contenere tutti i messaggi che verranno posizionati su di essa dall'applicazione di inserimento in un singolo "burst".

3. Le regole di scaricamento devono essere tutte disabilitate in modo che, anche quando la struttura inizia a riempirsi, i messaggi non vengano scaricati su SMDS. Ciò è dovuto al fatto che i costi delle prestazioni associati alla scrittura e alla lettura di messaggi da SMDS sono considerati inaccettabili.

Nel tempo, il numero di messaggi che l'applicazione di inserimento invia in un burst deve aumentare di diversi ordini di grandezza. Poiché l'applicazione di richiamo deve elaborare ogni messaggio in modo sequenziale, il numero di messaggi sulla coda aumenta fino al punto in cui la struttura si riempie.

A questo punto, l'applicazione di inserimento riceve un codice motivo (MQRC_STORAGE_MEDIUM_FULL) durante l'inserimento di un messaggio e l'operazione di inserimento non riesce. L'applicazione di inserimento può tollerare solo brevemente i periodi in cui non è in grado di inserire i messaggi nella coda. Se il periodo è troppo lungo, l'applicazione termina.

Supponendo che non si disponga del tempo, o delle competenze disponibili, per riscrivere l'applicazione di inserimento o ottenere l'applicazione, questo problema ha tre possibili soluzioni:

1. Aumentare la dimensione della struttura dell'applicazione.
2. Aggiungere le regole di offload alla struttura dell'applicazione in modo che i messaggi vengano scaricati su SMDS quando la coda inizia a riempirsi.
3. Associare SCM alla struttura.

La prima soluzione è rapida da implementare, ma non è disponibile memoria reale sufficiente sul CF.

Anche la seconda soluzione potrebbe essere rapida da implementare, ma l'impatto sulle prestazioni dello scaricamento su SMDS è considerato troppo significativo per utilizzare questa opzione.

La terza soluzione, associando SCM alla struttura, fornisce un equilibrio accettabile di costi e prestazioni.

L'associazione di SCM a una struttura comporta un maggiore utilizzo della memoria reale nella CF a causa della memoria aumentata che ottiene le operazioni utilizzate. Tuttavia, la quantità effettiva di memoria reale sarà inferiore alla quantità utilizzata nella prima opzione.

Un'altra considerazione è il costo del SCM. Tuttavia, questo costo è molto più economico rispetto allo stoccaggio reale. Questi fattori si combinano per rendere la terza opzione più economica rispetto alla prima opzione.

Anche se la terza opzione, potenzialmente, potrebbe non funzionare così come la prima opzione, gli algoritmi di pre - fetch e pre - staging utilizzati dalla CF possono combinarsi per rendere le differenze nelle prestazioni accettabili, o in alcuni casi trascurabili.

Certamente le prestazioni possono essere molto migliori rispetto all'utilizzo di SMDS per scaricare i messaggi.

Considerare ciò che accade in un periodo di tempo:

1. Inizialmente, l'applicazione di richiamo è attiva e attende che i messaggi vengano consegnati alla coda condivisa. L'applicazione di inserimento non è attiva e la coda condivisa è vuota.
2. In un certo momento, l'applicazione di inserimento diventa attiva e inizia a inserire un gran numero di messaggi nella coda condivisa. L'applicazione di richiamo inizia a ricevere i messaggi, ma la profondità della coda aumenta rapidamente perché l'applicazione di richiamo è più lenta dell'applicazione di inserimento.

   Di conseguenza, la struttura dell'applicazione inizia a riempirsi.
3. Con l'aumentare del tempo, l'applicazione di inserimento è ancora attiva. La struttura di applicazione si riempie fino a circa il 90%.

   Questo avviene quando l'algoritmo di pre - staging SCM inizia a spostare i messaggi dalla struttura in SCM, liberando spazio nella struttura.

   Poiché l'applicazione di richiamo riceve prima i messaggi con priorità più alta e più vecchi dalla coda, riceve sempre i messaggi dalla struttura e non deve attendere che i messaggi vengano portati in modo sincrono da SCM nella struttura.

4. L'applicazione di inserimento è ancora attiva e i messaggi vengono inseriti nella coda condivisa. Tuttavia, l'applicazione non riceve mai un codice motivo MQRC_STORAGE_MEDIUM_FULL, poiché in SCM esiste spazio sufficiente per memorizzare tutti i messaggi che non rientrano nella struttura.
5. Alla fine, l'applicazione di inserimento si arresta perché non ha più messaggi da inserire.

   L'algoritmo di pre - staging si arresta perché la struttura è inferiore al 90% in uso e l'applicazione di richiamo continua l'elaborazione dei messaggi nella coda.
6. Quando l'applicazione di richiamo inizia a liberare spazio nella struttura, l'algoritmo di precaricamento inizia a riportare i messaggi da SCM nella struttura.

   Poiché l'applicazione di richiamo elabora i messaggi nell'ordine previsto dall'algoritmo di precaricamento, l'applicazione di richiamo non viene mai bloccata in attesa che i dati del messaggio vengano portati in modo sincrono da SCM nella struttura.
7. Infine, l'applicazione di richiamo elabora tutti i messaggi sulla coda condivisa e attende che il messaggio successivo sia disponibile. La struttura e SCM sono vuoti di messaggi.

z/OS *Memoria di emergenza - configurazione di base*

Come configurare uno scenario di base per l'archiviazione di emergenza su IBM MQ.

Informazioni su questa attività

Importante: IBM z16 è pianificato per essere l'ultima generazione di IBM Z ® per supportare l'utilizzo di memoria flash virtuale (nota anche come SCM (Storage Class Memory) o SCM) per le immagini della CF (Coupling Facility). Per ulteriori informazioni, consultare IBM Z e IBM LinuxONE 4Q 2023 Statement of Direction.

In alternativa, è necessario utilizzare strutture più grandi o scaricare i messaggi su SMDS.

L'SMDS e lo scaricamento dei messaggi possono essere utilizzati insieme a SCM per ridurre la probabilità che un codice motivo MQRC_STORAGE_MEDIUM_FULL venga restituito a un'applicazione IBM MQ durante un'interruzione estesa.

Ad esempio, l'azienda dispone di un'applicazione che inserisce i messaggi nella coda e di un'applicazione che richiama i messaggi dalla coda. Durante l'esecuzione normale, si prevede che la profondità della coda sia prossima a zero, ma un requisito aziendale indica che il sistema è in grado di tollerare un'interruzione di due ore dell'applicazione che riceve i messaggi.

Ciò significa che la coda condivisa utilizzata deve essere in grado di contenere due ore di messaggi dall'applicazione di inserimento. Attualmente si ottiene questo risultato utilizzando le regole di offload predefinite e SMDS.

Si prevede che la frequenza dei messaggi inviati alla coda condivisa raddoppi nel breve - medio termine. Sebbene il tuo requisito che il sistema sia in grado di tollerare un'interruzione di due ore esista ancora, nella CF non è disponibile sufficiente memoria reale per raddoppiare la dimensione della struttura. Poiché la CF contenente la struttura dell'applicazione risiede su una macchina zEC12 , è possibile associare un SCM sufficiente alla struttura per memorizzare un numero sufficiente di messaggi, in modo che sia possibile tollerare un'interruzione di due ore

Questo scenario iniziale utilizza:

- Gruppo di condivisione code, `IBM1`, che contiene un singolo gestore code, `CSQ3`. Oltre alla struttura di gestione, il gruppo di condivisione code ha definito una singola struttura dell'applicazione, `SCEN1`.
- CF (Coupling Facility) `CF01`, in cui la struttura dell'applicazione `SCEN1` è memorizzata come struttura `IBM1SCEN1` . Questa struttura ha una dimensione massima di 1 GB.
- Coda condivisa singola, `SCEN1.Q` utilizzata dalla struttura dell'applicazione.

Questa configurazione è illustrata in Figura 61 a pagina 201.

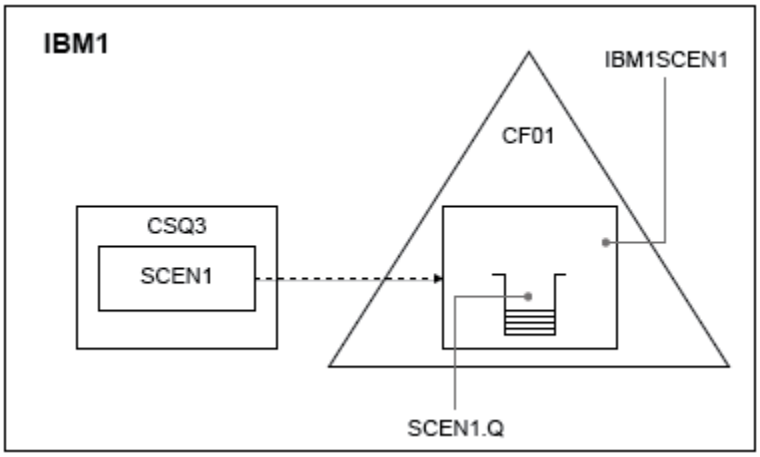


Figura 61. Configurazione base

Inoltre, si supponga che il gestore code CSQ3 sia già l'unico membro del gruppo di condivisione code IBM1.

È necessario aggiungere la definizione per la struttura IBM1SCEN1 alla politica CFRM (coupling facility resource manager). Per semplicità, la struttura è definita in modo che possa essere creata solo in una singola CF, CF01, specificando PREFLIST(CF01).

Attenzione: Per consentire l'alta disponibilità nel sistema di produzione, è necessario includere almeno due CF in PREFLIST per tutte le strutture utilizzate da IBM MQ.

Procedura

1. Aggiornare la politica CFRM utilizzando il seguente comando:

   ```
   SETXCF START,POLICY,TYPE=CFRM,POLNAME=IBM1SCEN1
   ```

   Politica CFRM di esempio per struttura IBM1SCEN1:

   ```
   STRUCTURE
   NAME(IBM1SCEN1)
   SIZE(1024M)
   INITSIZE(512M)
   ALLOWAUTOALT(YES)
   FULLTHRESHOLD(85)
   PREFLIST(CF01)
   ALLOWREALLOCATE(YES)
   DUPLEX(DISABLED)
   ENFORCEORDER(NO)
   ```

2. Verificare che la struttura sia stata creata correttamente, utilizzando il seguente comando:

   ```
   D XCF,STR,STRNAME=IBM1SCEN1
   ```

   A questo punto, la struttura non è stata assegnata, mostrata dalla riga STATUS , al gruppo di condivisione code.

3. Configurare IBM MQ per utilizzare la struttura definita nella politica CFRM.

   a. Utilizzare il comando DEFINE CFSTRUCT , con il nome struttura SCEN1 per creare un oggetto IBM MQ CFSTRUCT:

   ```
   DEFINE CFSTRUCT(SCEN1)
   CFCONLOS(TOLERATE)
   CFLEVEL(5)
   DESCR('Structure for SCM scenario 1')
   ```

```
RECOVER(NO)
RECAUTO(YES)
OFFLOAD(DB2)
OFFLD1SZ(64K) OFFLD1TH(70)
OFFLD2SZ(64K) OFFLD2TH(80)
OFFLD3SZ(64K) OFFLD3TH(90)
```

.

b. Convalidare la struttura, utilizzando il comando DISPLAY CFSTRUCT .

c. Definire la coda condivisa SCEN1.Q , per utilizzare la struttura SCEN1 , utilizzando il seguente comando MQSC:

```
DEFINE QLOCAL(SCEN1.Q) QSGDISP(SHARED) CFSTRUCT(SCEN1) MAXDEPTH(999999999)
```

4. Utilizzare IBM MQ Explorer per inserire un singolo messaggio nella coda SCEN1.Q e disattivare nuovamente il messaggio.
5. Immettere il seguente comando per controllare che la struttura sia ora assegnata:

```
D XCF,STR,STRNAME=IBM1SCEN1
```

Verificare l'output del comando, che la riga STATUS mostri ALLOCATED.

Risultati

È stata creata la configurazione di base. È ora possibile ottenere un'idea delle prestazioni della baseline della configurazione utilizzando qualsiasi metodo selezionato.

Operazioni successive

Aggiungere SMDS e SCM alla struttura iniziale

Concetti correlati

“Utilizzo della memoria della classe storage con code condivise” a pagina 192

L'utilizzo di SCM (storage class memory) può essere vantaggioso se utilizzato con code condivise IBM MQ for z/OS .

z/OS *Aggiunta di SMDS e SCM alla struttura iniziale*

Come aggiungere SMDS e SCM per l'archiviazione di emergenza su IBM MQ.

Informazioni su questa attività

Importante: IBM z16 è pianificato per essere l'ultima generazione di IBM Z ® per supportare l'utilizzo di memoria flash virtuale (nota anche come SCM (Storage Class Memory) o SCM) per le immagini della CF (Coupling Facility). Per ulteriori informazioni, consultare IBM Z e IBM LinuxONE 4Q 2023 Statement of Direction.

In alternativa, è necessario utilizzare strutture più grandi o scaricare i messaggi su SMDS.

Questa parte dell'attività utilizza la configurazione di base descritta in “Memoria di emergenza - configurazione di base” a pagina 200. Lo scenario descrive l'aggiunta di SMDS (shared message data sets) e quindi di SCM alla struttura iniziale.

Questa configurazione finale è illustrata in Figura 62 a pagina 203.

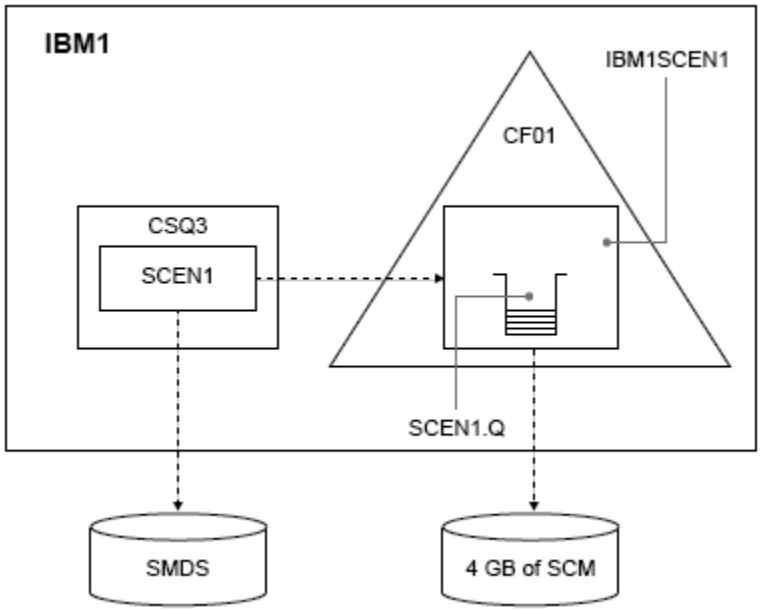


Figura 62. Configurazione dell'aggiunta di SMDS e SCM per lo storage di emergenza

Procedura

1. Creare il dataset SMDS utilizzato dalla struttura dell'applicazione SCEN1 , modificando il JCL di esempio **CSQ4SMDS** , come mostrato:

```
//CSQ4SMDS JOB NOTIFY=&SYSUID
//*
//* Allocate SMDS
//*
//DEFINE EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER           -
(NAME(CSQSMDS.SCEN1.CSQ3.SMDS) -
MEGABYTES(5000 3000)     -
LINEAR            -
SHAREOPTIONS(2 3) )      -
DATA                -
(NAME(CSQSMDS.SCEN1.CSQ3.SMDS.DATA) )
/*
//*
//* Format the SMDS
//*
//FORM   EXEC PGM=CSQJUFMT,COND=(0,NE),REGION=0M
//STEPLIB DD DSN=MQ800.SCSQANLE,DISP=SHR
//     DD DSN=MQ800.SCSQAUTH,DISP=SHR
//SYSUT1  DD DISP=OLD,DSN=CSQSMDS.SCEN1.CSQ3.SMDS
//SYSPRINT DD SYSOUT=*
```

2. Immettere il comando ALTER CFSTRUCT per modificare la struttura dell'applicazione SCEN1 , per utilizzare SMDS per l'offload, implementare le regole di offload predefinite:

```
ALTER CFSTRUCT(SCEN1) OFFLOAD(SMDS) OFFLD1SZ(32K) OFFLD2SZ(4K) OFFLD3SZ(0K)
DSGROUP('CSQSMDS.SCEN1.*.SMDS') DSBLOCK(1M)
```

   Tieni presente quanto segue:

   - Poiché SCEN1.Q è l'unica coda condivisa nella struttura dell'applicazione SCEN1 , il valore **DSBLOCK** è stato impostato su 1M, il valore più grande possibile. Questa dovrebbe essere l'impostazione più efficiente per il nostro scenario.

- Poiché i messaggi inviati dall'applicazione di inserimento sono 30 KB, lo scaricamento su SMDS non viene avviato fino a quando non viene soddisfatta la seconda regola di scaricamento, quando la struttura è piena all ' 80%.

3. Eseguire nuovamente l'applicazione di test.

   Notare l'aumento della memoria dei messaggi sulla coda.
4. Aggiungere 4 GB di SCM alla struttura IBM1SCEN1 effettuando la seguente procedura:
   a) Controllare la quantità di SCM installata e assegnata a CF01, immettendo il seguente comando:

   ```
   D CF,CFNAME=CF01
   ```

   b) Controllare le figure STORAGE-CLASS MEMORY nella sezione STORAGE CONFIGURATION dell'emissione visualizzata per visualizzare la memoria disponibile.
   c) Aggiornare la politica CFRM con le parole chiave SCMMAXSIZE e SCMALGORITHM come mostrato:

   ```
   STRUCTURE
   NAME(IBM1SCEN1)
   SIZE(1024M)
   INITSIZE(512M)
   ALLOWAUTOALT(YES)
   FULLTHRESHOLD(85)
   PREFLIST(CF01)
   ALLOWREALLOCATE(YES)
   DUPLEX(DISABLED)
   ENFORCEORDER(NO)
   SCMMAXSIZE(4G)
   SCMALGORITHM(KEYPRIORITY1)
   ```
5. Attivare la politica CFRM immettendo il seguente comando:

   ```
   SETXCF START,POLICY,TYPE=CFRM,POLNAME=polname
   ```
6. Rigenerare la struttura IBM1SCEN1 .

   È necessario eseguire questa procedura perché la struttura è stata assegnata quando sono state apportate le precedenti modifiche.

   Immettere il seguente comando per ricreare la struttura:

   ```
   SETXCF START,REBUILD,STRNM=IBM1SCEN1
   ```

Risultati

SCM è stato aggiunto correttamente alla propria configurazione.

Operazioni successive

Ottimizzare le prestazioni del sistema. Per ulteriori informazioni, consultare “Ottimizzazione dell'utilizzo della memoria della classe di archiviazione” a pagina 204.

z/OS *Ottimizzazione dell'utilizzo della memoria della classe di archiviazione*

Come migliorare l'utilizzo di SCM (storage class memory).

Importante: IBM z16 è pianificato per essere l'ultima generazione di IBM Z ® per supportare l'utilizzo di memoria flash virtuale (nota anche come SCM (Storage Class Memory) o SCM) per le immagini della CF (Coupling Facility). Per ulteriori informazioni, consultare IBM Z e IBM LinuxONE 4Q 2023 Statement of Direction.

In alternativa, è necessario utilizzare strutture più grandi o scaricare i messaggi su SMDS.

Esegui il seguente comando:

```
D XCF,STR,STRNAME=IBM1SCEN1
```

Poiché la struttura era già piena di dati del messaggio, a causa dei precedenti test, parte della ricostruzione implicava la pre - preparazione di alcuni dei messaggi dalla struttura in SCM. Questo processo è stato avviato utilizzando il comando precedente.

L'output di questo comando produce, ad esempio:

```
ACTIVE STRUCTURE
----------------
ALLOCATION TIME: 06/17/2014 09:28:50
CFNAME : CF01
COUPLING FACILITY: 002827.IBM.02.00000000B8D7
PARTITION: 3B CPCID: 00
STORAGE CONFIGURATION ALLOCATED MAXIMUM %
ACTUAL SIZE: 1024 M 1024 M 100
AUGMENTED SPACE: 3 M 142 M 2
STORAGE-CLASS MEMORY: 88 M 4096 M 2
ENTRIES: 120120 1089536 11
ELEMENTS: 240240 15664556 1
SPACE USAGE IN-USE TOTAL %
ENTRIES: 84921 219439 38
ELEMENTS: 2707678 3149050 85
EMCS: 2 282044 0
LOCKS: 1024
SCMHIGHTHRESHOLD : 90
SCMLOWTHRESHOLD : 70
ACTUAL SUBNOTIFYDELAY: 5000
PHYSICAL VERSION: CD5186A0 2BD8B85C
LOGICAL VERSION: CD515C50 CE2ED258
SYSTEM-MANAGED PROCESS LEVEL: 9
XCF GRPNAME : IXCLO053
DISPOSITION : KEEP
ACCESS TIME : NOLIMIT
MAX CONNECTIONS: 32
# CONNECTIONS : 1
CONNECTION NAME ID VERSION  SYSNAME JOBNAME  ASID STATE
---------------- -- --------  ------- --------  ---- ----------------
CSQEIBM1CSQ301  01 00010059 SC61    CSQ3MSTR 0091 ACTIVE
```

Notare quanto segue dall'emissione del comando:

- STORAGE_CLASS MEMORY fornisce la conferma che un **MAXIMUM** di 4096 MB di SCM è stato aggiunto alla struttura.
- La figura ALLOCATED per la quantità di STORAGE-CLASS MEMORY utilizzata per la preparazione. Ora c'è spazio libero nella struttura dove non c'era nessuno prima che SCM fosse aggiunto.
- La quantità di AUGMENTED SPACE utilizzata per tracciare l'utilizzo di SCM.
- Il punto in cui l'algoritmo di pre - staging inizia a spostare i dati dalla struttura in SCM è quando la struttura è piena al 90%. Ciò è indicato dalla proprietà **SCMHIGHTHRESHOLD** non configurabile.
- Il punto al di sotto del quale l'algoritmo di precaricamento inizia a spostare i dati da SCM nella struttura è quando la struttura è piena al 70%. Ciò è indicato dalla proprietà **SCMLOWTHRESHOLD** non configurabile.

È ora possibile testare diversi metodi per ottimizzare l'utilizzo di SCM. Tieni presente quanto segue:

- Una volta utilizzato SCM per memorizzare i messaggi, non è possibile modificare la struttura fino a quando non vengono rimossi tutti i dati da SCM.

  In questo caso, ciò significa che il rapporto voce - elemento è bloccato al valore che era presente quando SCM è stato utilizzato per la prima volta. È necessario assicurarsi con attenzione che la struttura sia nello stato desiderato, prima che l'algoritmo di pre - staging inizi a spostare i dati in SCM.
- La dimensione della struttura corrente è corretta prima di utilizzare SCM?

  Ad esempio, è stato aumentato **INITSIZE** da 512 MB a una DIMENSIONE di 1 GB?

  Se non si esegue questa operazione, è possibile che, sebbene sia stata abilitata la struttura per l'alterazione automatica, l'algoritmo di pre - staging inizierà a spostare i dati in SCM prima che l'alterazione abbia la possibilità di iniziare. Di conseguenza, la struttura viene congelata utilizzando 512 MB di memoria reale.
- Il rapporto voce - elemento è corretto prima di utilizzare SCM?

L'obiettivo di questo scenario è aumentare il numero di puntatori di messaggi scaricati che possono essere memorizzati nella struttura e SCM nel suo insieme, oltre a mantenere il maggior numero possibile di messaggi interamente nella memoria della struttura. L'accesso a questi messaggi è più rapido rispetto all'accesso ai messaggi su SMDS.

Di conseguenza, è necessario disporre di una struttura che inizi con un rapporto voce - elemento che sia buono per la memorizzazione dei messaggi e quindi transizioni a un rapporto che sia buono per la memorizzazione dei puntatori dei messaggi prima dell'avvio dell'algoritmo di prestage. Questa transizione può essere ottenuta, in parte, utilizzando le regole di offload IBM MQ .

Modificare le regole di offload immettendo il seguente comando:

```
ALTER CFSTRUCT(SCEN1) OFFLD1SZ(0K)
```

Potrebbe essere necessario eseguire diverse esecuzioni per ottimizzare i rapporti voce - elemento.

La seguente tabella mostra i possibili miglioramenti nel numero di messaggi inseriti nella coda durante le diverse fasi dello scenario di memoria di emergenza.

Tabella 18. Confronto dei risultati per lo scenario di emergenza

Descrizione test	**Numero di messaggi**	**Tempo per riempire la coda (secondi)**
Configurazione base	27,850	3.2
SMDS con regole di offload predefinite	205,000	158
SCM con regole di offload predefinite	828,610	469
SCM con regole di offload modificate	1,135,775	679

L'ultima riga nella tabella mostra che la modifica delle regole di offload ha avuto l'effetto richiesto.

È necessario esaminare il sistema per vedere se è possibile migliorare su queste cifre in qualche modo. Ad esempio, si potrebbe esaurire l'archiviazione SMDS disponibile. Se è possibile assegnare più memoria SMDS, è necessario aumentare il numero di messaggi sulla coda in modo significativo.

z/OS *Prestazioni migliorate - configurazione di base*

Come impostare uno scenario di base per migliorare le prestazioni utilizzando le code condivise su IBM MQ.

Informazioni su questa attività

Importante: IBM z16 è pianificato per essere l'ultima generazione di IBM Z ® per supportare l'utilizzo di memoria flash virtuale (nota anche come SCM (Storage Class Memory) o SCM) per le immagini della CF (Coupling Facility). Per ulteriori informazioni, consultare IBM Z e IBM LinuxONE 4Q 2023 Statement of Direction.

In alternativa, è necessario utilizzare strutture più grandi o scaricare i messaggi su SMDS.

Questo scenario descrive l'utilizzo di SCM per aumentare il numero di messaggi che è possibile memorizzare su una coda condivisa senza incorrere nel costo delle prestazioni dell'utilizzo di SMDS.

Questo scenario iniziale è molto simile a quello utilizzato per lo stoccaggio di emergenza e utilizza:

- Gruppo di condivisione code, `IBM1`, che contiene un singolo gestore code, `CSQ3`. Oltre alla struttura di gestione, il gruppo di condivisione code ha definito una singola struttura dell'applicazione, `SCEN2`.
- CF (Coupling Facility) `CF01`, in cui la struttura dell'applicazione `SCEN2` è memorizzata come struttura `IBM1SCEN2` . Questa struttura ha una dimensione massima di 2 GB.
- Una singola coda condivisa, `SCEN2.Q`, configurata per utilizzare la struttura dell'applicazione.

Questa configurazione è illustrata in Figura 63 a pagina 207.

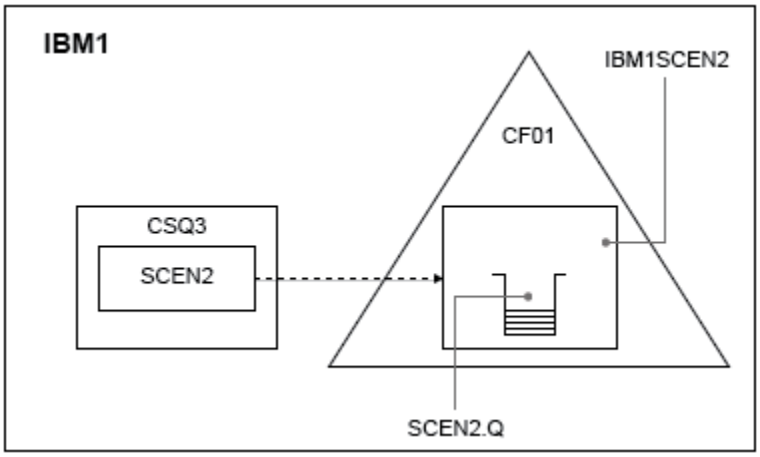


Figura 63. Configurazione base

Inoltre, si supponga che il gestore code CSQ3 sia già l'unico membro del gruppo di condivisione code IBM1.

È necessario aggiungere la definizione per la struttura IBM1SCEN2 alla politica CFRM (coupling facility resource manager). Per semplicità, la struttura è definita in modo che possa essere creata solo in una singola CF, CF01, specificando PREFLIST(CF01).

Politica CFRM di esempio per struttura IBM1SCEN2:

```
STRUCTURE
NAME(IBM1SCEN2)
SIZE(2048M)
INITSIZE(2048M)
ALLOWAUTOALT(YES)
FULLTHRESHOLD(85)
PREFLIST(CF01)
ALLOWREALLOCATE(YES)
DUPLEX(DISABLED)
ENFORCEORDER(NO)
```

Entrambe le parole chiave **INITSIZE** e **SIZE** hanno il valore 2048M in modo che la struttura non possa ridimensionare.

Procedura

1. Aggiornare la politica CFRM utilizzando il seguente comando:

   ```
   SETXCF START,POLICY,TYPE=CFRM,POLNAME=IBM1SCEN2
   ```

2. Verificare che la struttura sia stata creata correttamente, utilizzando il seguente comando:

   ```
   D XCF,STR,STRNAME=IBM1SCEN2
   ```

   L'immissione del precedente comando fornisce il seguente output:

   ```
   RESPONSE=SC61
   IXC360I 07.58.51 DISPLAY XCF 581
   STRNAME: IBM1SCEN2
   STATUS: NOT ALLOCATED
   POLICY INFORMATION:
   POLICY SIZE : 2048 M
   POLICY INITSIZE: 2048 M
   POLICY MINSIZE : 1536 M
   FULLTHRESHOLD : 85
   ALLOWAUTOALT : YES
   REBUILD PERCENT: N/A
   ```

```
DUPLEX : DISABLED
ALLOWREALLOCATE: YES
PREFERENCE LIST: CF01
ENFORCEORDER : NO
EXCLUSION LIST IS EMPTY

EVENT MANAGEMENT: MESSAGE-BASED   MANAGER SYSTEM NAME: SC53
MANAGEMENT LEVEL : 01050107
```

A questo punto, la struttura non è stata assegnata, mostrata dalla riga STATUS , al gruppo di condivisione code.

3. Configurare IBM MQ per utilizzare la struttura definita nella politica CFRM.
   a. Utilizzare il comando DEFINE CFSTRUCT , con il nome struttura SCEN2 per creare un oggetto IBM MQ CFSTRUCT.

```
DEFINE CFSTRUCT(SCEN2)
CFCONLOS(TOLERATE)
CFLEVEL(5)
DESCR('Structure for SCM scenario 2')
RECOVER(NO)
RECAUTO(YES)
OFFLOAD(DB2)
OFFLD1SZ(64K) OFFLD1TH(70)
OFFLD2SZ(64K) OFFLD2TH(80)
OFFLD3SZ(64K) OFFLD3TH(90)
```

   b. Controllare la struttura utilizzando il comando DISPLAY CFSTRUCT .
   c. Definire la coda condivisa SCEN2.Q , per utilizzare la struttura SCEN2 , utilizzando il seguente comando MQSC:

```
DEFINE QLOCAL(SCEN2.Q) QSGDISP(SHARED) CFSTRUCT(SCEN2) MAXDEPTH(999999999)
```

4. Utilizzare IBM MQ Explorer per inserire un singolo messaggio nella coda SCEN2.Q e disattivare nuovamente il messaggio.
5. Immettere il seguente comando per controllare che la struttura sia ora assegnata:

```
D XCF,STR,STRNAME=IBM1SCEN2
```

Esaminare l'output del comando, una parte dei quali viene visualizzata, e assicurarsi che la riga STATUS mostri ALLOCATED.

```
RESPONSE=SC61
IXC360I 08.31.27 DISPLAY XCF 703
STRNAME: IBM1SCEN2
STATUS: ALLOCATED
EVENT MANAGEMENT: MESSAGE-BASED
TYPE: SERIALIZED LIST
POLICY INFORMATION:
POLICY SIZE : 2048 M
POLICY INITSIZE: 2048 M
POLICY MINSIZE : 1536 M
FULLTHRESHOLD : 85
ALLOWAUTOALT : YES
REBUILD PERCENT: N/A
DUPLEX : DISABLED
ALLOWREALLOCATE: YES
PREFERENCE LIST: CF01
ENFORCEORDER : NO
EXCLUSION LIST IS EMPTY
```

Inoltre, annotare i valori dei campi nella sezione SPACE USAGE:

- Voci
- ELEMENTI
- EMCS

- BLOCCHI

Di seguito è riportato un esempio dei valori:

```
SPACE USAGE   IN-USE      TOTAL    %
ENTRIES:      344686     345242   99
ELEMENTS:    6548455    6548467   99
EMCS:              2     780318    0
LOCKS:          1024
```

Risultati

È stata creata la configurazione di base. È ora possibile ottenere un'idea delle prestazioni della baseline della configurazione utilizzando qualsiasi metodo selezionato.

Operazioni successive

È necessario verificare lo scenario di base. Come esempio, è possibile utilizzare le seguenti tre applicazioni, avviando le applicazioni nell'ordine mostrato e, eseguendole contemporaneamente.

1. Utilizzare un'applicazione PCF per richiedere la profondità corrente (**CURDEPTH**) per SCEN2.Q ogni 5 secondi. L'output può essere utilizzato per tracciare la profondità della coda nel tempo.
2. Un'applicazione di acquisizione a thread singolo riceve ripetutamente i messaggi da SCEN2.Q, utilizzando una ricezione con un'attesa infinita. Per simulare l'elaborazione dei messaggi che sono stati rimossi, l'applicazione di richiamo si interrompe per quattro millisecondi per ogni dieci messaggi che ha rimosso.
3. Un'applicazione di inserimento a thread singolo inserisce un totale di un milione di messaggi non persistenti da 4 KB in SCEN2.Q. Questa applicazione non si interrompe tra l'inserimento di ogni messaggio, quindi i messaggi vengono inseriti SCEN2.Q più velocemente di quanto l'applicazione di richiamo possa ottenere.

   Di conseguenza, quando l'applicazione di inserimento è in esecuzione, la profondità di SCEN2.Q aumenta.

   Quando la struttura IBM1SCEN2 è riempita e l'applicazione di inserimento riceve un codice motivo MQRC_STORAGE_MEDIUM_FULL, l'applicazione di inserimento rimane inattiva per cinque secondi prima di tentare di inserire il messaggio successivo nella coda.

È possibile tracciare i risultati dell'applicazione CURDEPTH in un periodo di tempo. Si ottiene una qualche forma di uscita dell'onda del dente di sega mentre l'applicazione di inserimento si interrompe per consentire alla coda di svuotarsi parzialmente.

Andare a “Aggiunta di SCM alla struttura iniziale” a pagina 209.

Concetti correlati

“Utilizzo della memoria della classe storage con code condivise” a pagina 192

L'utilizzo di SCM (storage class memory) può essere vantaggioso se utilizzato con code condivise IBM MQ for z/OS .

z/OS *Aggiunta di SCM alla struttura iniziale*

Come si aggiunge SCM per migliorare le prestazioni su IBM MQ.

Informazioni su questa attività

Importante: IBM z16 è pianificato per essere l'ultima generazione di IBM Z ® per supportare l'utilizzo di memoria flash virtuale (nota anche come SCM (Storage Class Memory) o SCM) per le immagini della CF (Coupling Facility). Per ulteriori informazioni, consultare IBM Z e IBM LinuxONE 4Q 2023 Statement of Direction.

In alternativa, è necessario utilizzare strutture più grandi o scaricare i messaggi su SMDS.

Questa parte dell'attività utilizza la configurazione di base descritta in “Prestazioni migliorate - configurazione di base” a pagina 206. Lo scenario descrive l'aggiunta di SCM alla struttura iniziale.

Questa configurazione finale è illustrata in Figura 64 a pagina 210.

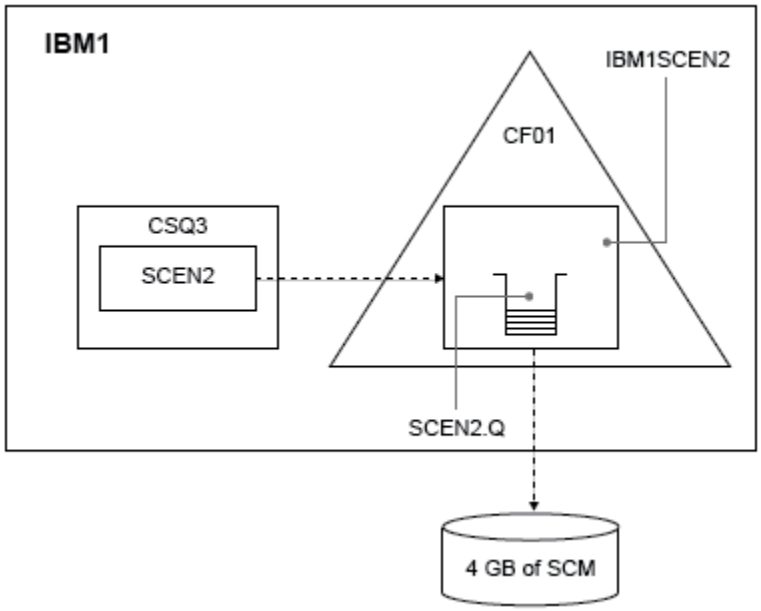


Figura 64. Configurazione dell'aggiunta di SCM per prestazioni migliori

Procedura

1. Aggiungere 4 GB di SCM alla struttura IBM1SCEN2 effettuando la seguente procedura:
   a) Controllare la quantità di SCM installata e assegnata a CF01, immettendo il seguente comando:

```
D CF,CFNAME=CF01
```

   b) Controllare le figure STORAGE-CLASS MEMORY nella sezione STORAGE CONFIGURATION dell'emissione visualizzata per visualizzare la memoria disponibile.
   c) Aggiornare la politica CFRM con le parole chiave SCMMAXSIZE e SCMALGORITHM come mostrato:

```
STRUCTURE
NAME(IBM1SCEN2)
SIZE(2048M)
INITSIZE(2048M)
ALLOWAUTOALT(YES)
FULLTHRESHOLD(85)
PREFLIST(CF01)
ALLOWREALLOCATE(YES)
DUPLEX(DISABLED)
ENFORCEORDER(NO)
SCMMAXSIZE(4G)
SCMALGORITHM(KEYPRIORITY1)
```

2. Attivare la politica CFRM immettendo il seguente comando:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=IBM1SCEN2
```

3. Rigenerare la struttura IBM1SCEN2 .
   È necessario eseguire questa procedura perché la struttura è stata assegnata quando sono state apportate le precedenti modifiche.

   Immettere il seguente comando per ricreare la struttura:

```
SETXCF START,REBUILD,STRNM=IBM1SCEN2
```

4. Immettere il seguente comando per confermare la nuova configurazione della struttura:

```
D XCF,STR,STRNAME=IBM1SCEN2
```

Esaminare l'emissione del comando, di cui una parte è riportata di seguito:

```
SPACE USAGE     IN-USE       TOTAL      %
ENTRIES:          33        342684      0
ELEMENTS:         48       6503697      0
EMCS:              2        575600      0
LOCKS:                        1024
```

Risultati

Calcolare la modifica nell'utilizzo della memoria reale mediante l'aumento della memoria di controllo richiesta per utilizzare SCM.

- Prima di aggiungere SCM alla struttura, la struttura ha questi totali come mostrato in “Prestazioni migliorate - configurazione di base” a pagina 206:
  - 345.242 voci
  - 6.548.467 elementi
  - 780 318 EMCS
- Dopo l'aggiunta di SCM alla struttura, la struttura ha i seguenti totali:
  - 342.684 voci
  - 6.503.697 elementi
  - 575.600 EMCS

Utilizzando queste figure, dopo l'aggiunta del SCM, la struttura è ridotta in dimensioni da:

- 2558 voci
- 44.770 elementi
- 204.718 EMCS

La quantità di memoria della struttura utilizzata per gestire SCM è la seguente per una struttura da 2 GB con 4 GB di SCM assegnati:

```
(2558 + 44,770 + 204,718) * 256 = 61.5 MB
```

Si noti che l'aggiunta di più SCM è probabile che raggiunga solo una riduzione marginale della dimensione della struttura, perché la quantità di memoria di controllo utilizzata per tenere traccia di SCM aumenta, sia come dimensione della struttura, sia come la quantità di SCM assegnata aumenta.

Operazioni successive

Ripetere i test descritti nella sezione finale di “Prestazioni migliorate - configurazione di base” a pagina 206.

È possibile tracciare i risultati dell'applicazione revisionata in un periodo di tempo. Confrontando il grafico con quello ottenuto in precedenza, ora si ottiene un output senza un'onda a dente di sega, poiché l'applicazione di inserimento non deve più attendere che la coda sia parzialmente vuota.

Per ulteriori informazioni, fare riferimento a MP16: WebSphere MQ for z/OS - Capacity planning & tuning.

z/OS Accodamento distribuito e gruppi di condivisione code

L'accodamento distribuito e i gruppi di condivisione code sono due tecniche che possono essere utilizzate per aumentare la disponibilità dei propri sistemi dell'applicazione. Utilizzare questo argomento per trovare ulteriori informazioni su queste tecniche.

Per completare l'alta disponibilità dei messaggi sulle code condivise, il componente di accodamento distribuito di IBM MQ ha funzioni aggiuntive per fornire quanto segue:

- Maggiore disponibilità per la rete.
- Capacità aumentata per le connessioni di rete in entrata al gruppo di condivisione code.

La Figura 65 a pagina 212 illustra l'accodamento distribuito e i gruppi di condivisione code. Mostra due gestori code all'interno di un sysplex, entrambi appartenenti allo stesso gruppo di condivisione code. Entrambi possono accedere alla coda condivisa SQ1. I gestori code nella rete (su AIX e Windows ad esempio) possono inserire messaggi in questa coda tramite l'iniziatore di canali di uno dei gestori code. Le applicazioni clonate su entrambi i gestori code gestiscono la coda.

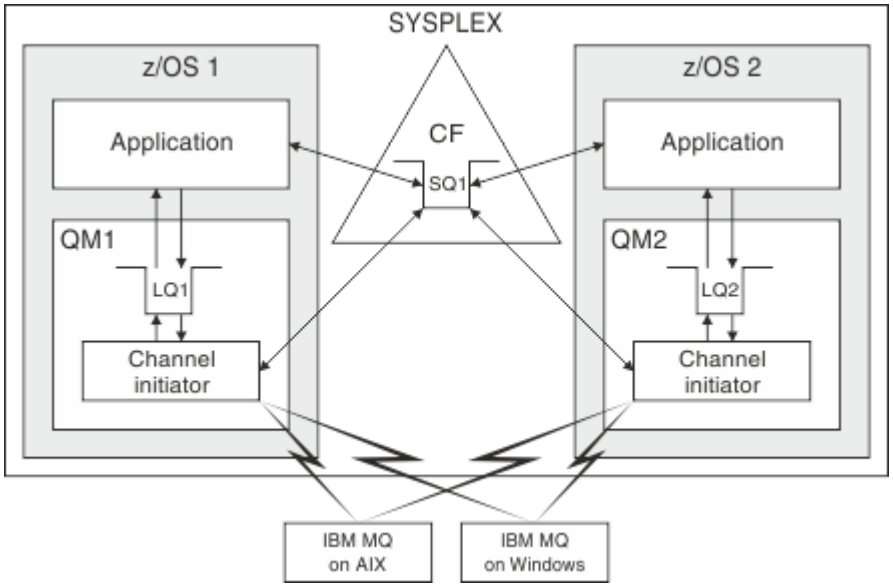


Figura 65. Accodamento distribuito e gruppi di condivisione code

Concetti correlati

"Canali condivisi" a pagina 212
Utilizzare questo argomento per comprendere i concetti di canali condivisi e il loro utilizzo con IBM MQ for z/OS.

"Accodamento all'interno del gruppo" a pagina 217
Questa sezione descrive l'accodamento all'interno del gruppo, una funzione IBM MQ for z/OS univoca per la piattaforma z/OS . Questa funzione è disponibile solo per i gestori code definiti per un gruppo di condivisione code.

"Cluster e gruppi di condivisione code" a pagina 214
Utilizzare questo argomento per comprendere come utilizzare i gruppi di condivisione code con i cluster.

z/OS *Canali condivisi*

Utilizzare questo argomento per comprendere i concetti di canali condivisi e il loro utilizzo con IBM MQ for z/OS.

Alcuni prodotti di rete forniscono un meccanismo per nascondere gli errori del server dalla rete o per bilanciare le richieste di rete in entrata su una serie di server idonei. I prodotti di rete rendono disponibile una *porta generica* per le richieste di connessione di rete in entrata e la richiesta in entrata può essere soddisfatta collegandosi a uno dei server idonei.

Questi prodotti di rete includono:

- Risorse generiche VTAM
- Distributore SYSPLEX

L'iniziatore di canali si avvale di questi prodotti per utilizzare le capacità delle code condivise

Ci sono due tipi di canali condivisi, *canale in entrata condivisoe canale in uscita condiviso*.

- Canali in entrata condivisi
- Canali in uscita condivisi

Per ulteriori informazioni sui canali, consultare

- Riepilogo canale condiviso
- Stato canale condiviso

Canali in entrata condivisi

Ciascun iniziatore di canali nel gruppo di condivisione code avvia un'attività listener aggiuntiva per l'ascolto su una *porta generica*. Questa porta generica viene resa disponibile alla rete da una delle tecnologie di supporto (VTAM, TCP/IP). Le richieste di collegamento di rete in entrata alla porta generica vengono inviate dalla tecnologia di rete a uno qualsiasi dei listener nel QSG (queue sharing group) in ascolto sulla porta generica.

È possibile avviare un canale sull'iniziatore di canali a cui è indirizzato il collegamento in entrata se l'iniziatore di canali ha accesso a una definizione di canale per un canale con tale nome. È possibile definire una definizione di canale come privata per un gestore code o memorizzata nel repository condiviso e quindi disponibile ovunque (una definizione globale). Ciò significa che è possibile rendere disponibile una definizione di canale su qualsiasi iniziatore di canale nel gruppo di condivisione code definendolo come una definizione globale.

C'è un'ulteriore differenza quando si avvia un canale attraverso la porta generica; la sincronizzazione del canale è con il gruppo di condivisione code e non con un singolo gestore code. Ad esempio, si consideri un gestore code remoto che avvia un canale tramite la porta generica. Quando il canale viene avviato per la prima volta, potrebbe avviarsi sul gestore code QM1 e sul flusso di messaggi. Se il canale viene arrestato e riavviato sul gestore code QM2, le informazioni sul numero di messaggi trasmessi sono ancora corrette perché la sincronizzazione è con il gruppo di condivisione code.

È possibile utilizzare un canale in entrata avviato tramite la porta generica per inserire i messaggi in qualsiasi coda. Il gestore code remoto non sa se la coda di destinazione è condivisa o meno. Se la coda di destinazione è una coda condivisa, il gestore code remoto si connette tramite qualsiasi iniziatore di canale disponibile in modo bilanciato dal carico e i messaggi vengono inseriti nella coda condivisa.

Se la coda di destinazione è una coda privata, i messaggi vengono inseriti nella coda privata di proprietà del gestore code a cui è connessa l'istanza corrente del canale. In questo ambiente, noto come *code locali replicate*, ogni gestore code deve avere la stessa serie di code private definite.

Configurazione dei canali SVRCONN per un gruppo di condivisione code

La configurazione ottimale per i canali SVRCONN in un gruppo di condivisione code è quella di impostare i listener privati in ogni CHINIT che utilizzano un numero di porta diverso dai canali punto a punto. Queste porte listener vengono quindi utilizzate come risorse 'back-end' per un nuovo meccanismo di distribuzione del carico di lavoro come il distributore Sysplex che utilizza gli indirizzi IP virtuali (VIPA). L'indirizzo VIPA esterno viene quindi utilizzato come indirizzo di destinazione per le definizioni CLNTCONN nella rete. Il canale SVRCONN può essere definito con QSGDISP (GROUP) in modo che la stessa definizione sia disponibile per tutti i gestori code in QSG. Questa configurazione evita l'utilizzo di un listener condiviso e quindi riduce l'effetto delle prestazioni del gruppo di condivisione code che mantiene lo stato del canale condiviso, che non è necessario per i canali client/server.

Canali in uscita condivisi

Un canale in uscita viene considerato come un canale condiviso se sta prendendo messaggi da una coda di trasmissione condivisa. Se è condiviso, conserva le informazioni di sincronizzazione a livello di gruppo di condivisione code. Ciò significa che il canale può essere riavviato su un gestore code e su un'istanza dell'iniziatore di canali diversi all'interno del gruppo di condivisione code se il sottosistema di comunicazione, l'iniziatore di canali o il gestore code hanno esito negativo. Il riavvio dei canali non riusciti in questo modo è una funzione dei canali condivisi denominata *ripristino del canale peer*.

Bilanciamento del carico di lavoro per i canali in uscita condivisi

Un canale condiviso in uscita è idoneo per l'avvio su qualsiasi iniziatore di canali all'interno del gruppo di condivisione code, se non è stato specificato che si desidera che venga avviato su un particolare iniziatore di canali. L'iniziatore di canali selezionato da IBM MQ è determinato utilizzando i seguenti criteri:

- Il sottosistema di comunicazioni richiesto è attualmente disponibile per l'iniziatore di canali?
- Una connessione Db2 è disponibile per l'iniziatore di canali?
- Quale iniziatore di canali ha il carico di lavoro corrente più basso? Il carico di lavoro include i canali attivi e in fase di nuovo tentativo.

Riepilogo canale condiviso

I canali condivisi differiscono dai canali privati nei seguenti modi:

Canale privato
Collegato a un singolo iniziatore di canale.

- Il canale in uscita utilizza una coda di trasmissione locale.
- Canale in entrata avviato tramite una porta locale.
- Informazioni di sincronizzazione conservate in SYSTEM.CHANNEL.SYNCQ .

Canale condiviso
Carico di lavoro bilanciato con alta disponibilità.

- Il canale in entrata utilizza una coda di trasmissione condivisa.
- Canale in entrata avviato tramite una porta generica.
- Informazioni di sincronizzazione conservate in SYSTEM.QSG.CHANNEL.SYNCQ .

Specificare se un canale è privato o condiviso quando si avvia il canale utilizzando le opzioni CHLDISP con il comando START CHANNEL . Un canale condiviso può essere avviato attivando nello stesso modo di un canale privato. Tuttavia, quando un canale condiviso viene avviato, IBM MQ esegue il bilanciamento del workload e avvia il canale sull'iniziatore di canale più appropriato all'interno del gruppo di condivisione code. (Se necessario, è possibile specificare che un canale condiviso deve essere avviato su un particolare iniziatore di canali.)

Stato canale condiviso

Gli iniziatori di canale in un gruppo di condivisione code mantengono una tabella di stato del canale condiviso in Db2. Registra quali canali sono attivi e su quali iniziatori di canali. La tabella di stato del canale condiviso viene utilizzata se si verifica un errore dell'iniziatore di canale o del sistema di comunicazioni. Indica quali canali devono essere riavviati su un iniziatore di canali differente nel gruppo di condivisione code.

z/OS *Cluster e gruppi di condivisione code*

Utilizzare questo argomento per comprendere come utilizzare i gruppi di condivisione code con i cluster.

È possibile rendere le code condivise disponibili per un cluster in una singola definizione. A tale scopo, specificare il nome del cluster quando si definisce la coda condivisa.

Gli utenti nella rete vedono la coda condivisa come ospitata da ciascun gestore code all'interno del gruppo di condivisione code (la coda condivisa non è pubblicizzata come ospitata dal gruppo di condivisione code). I client possono avviare le sessioni con qualsiasi membro del gruppo di condivisione code per inserire i messaggi nella stessa coda condivisa.

Figura 66 a pagina 215 mostra in che modo i membri di un cluster possono accedere a una coda condivisa tramite qualsiasi membro del gruppo di condivisione code.

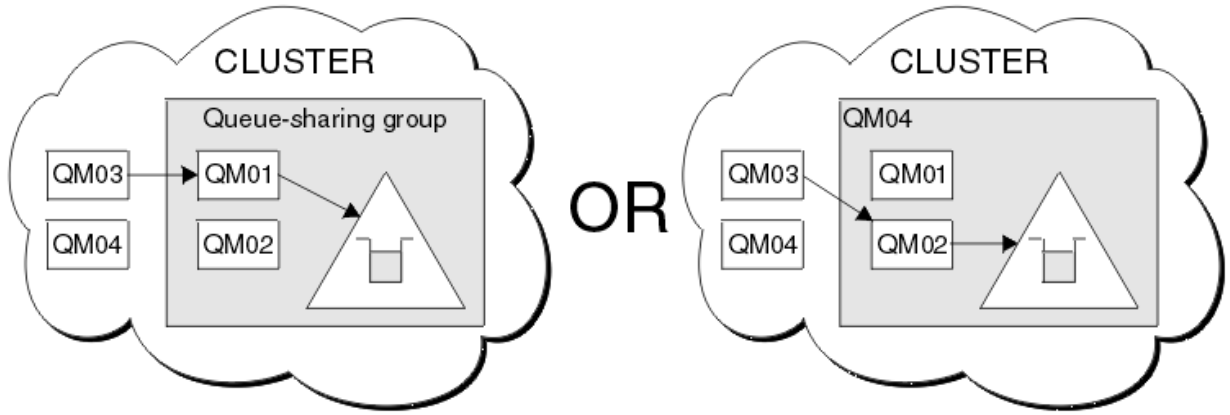


Figura 66. Un gruppo di condivisione code come parte di un cluster

z/OS Influenza della distribuzione del carico di lavoro con le code condivise

Utilizzare questo argomento per comprendere i fattori che influenzano la distribuzione del carico di lavoro con code condivise in un gruppo di condivisione code.

IBM MQ non fornisce il bilanciamento del carico di lavoro per code condivise. Tuttavia, la distribuzione del carico di lavoro in un gruppo di condivisione code (QSG) può essere influenzata in *una modalità basata sul pull*. La scelta del gestore code che serve una coda (riceve un messaggio scritto in una coda condivisa) è influenzata dalla capacità di elaborazione disponibile di ciascun gestore code nel gruppo di condivisione code e dagli obiettivi di gestione del carico di lavoro definiti nel sysplex.

Tuttavia, è importante apprezzare che il gestore code che esegue MQPUT di un messaggio può anche avere una grande influenza nel decidere quale gestore code riceve il messaggio.

È più probabile che un gestore code locale esegua MQGET

Per un'applicazione che esegue un MQPUT, il gestore code locale viene definito il gestore code a cui è connessa l'applicazione.

Le seguenti considerazioni influenzano esattamente il gestore code che esegue un MQPUT di un messaggio eseguendo un MQGET per conto di un'applicazione di richiamo.

Quando un messaggio viene inserito in una coda condivisa vuota, il gestore code locale viene generalmente pubblicato prima che venga notificato uno qualsiasi degli altri gestori code nel gruppo di condivisione code. Se il gestore code locale è in grado di elaborare il messaggio, riceve una notifica di transizione di elenco dalla CF (Coupling Facility) prima di qualsiasi altro gestore code in QSG. (Una notifica di transizione di elenco è una notifica che la coda condivisa ha modificato lo stato da vuoto a non vuoto.)

Gli scenari possibili, in questo caso, sono i seguenti:

1. MQPUT del messaggio non persistente fuori dal punto di sincronizzazione e *inserimento rapido nel getter in attesa*.

Se è presente un'applicazione con un *MQGET con attesa* sul gestore code locale per la coda, l'MQPUT del messaggio viene passato direttamente al buffer dell'applicazione ricevente e non viene scritto nella coda. Ciò è vero per le code condivise e non condivise. Questa funzione è spesso chiamata meccanismo *fast put to a waiting getter* . Nel caso di code condivise, nessun altro gestore code nel QSG viene notificato in quanto non vi è alcuna transizione da vuota a non vuota della coda. Ciò significa, ad esempio, che se questo gestore code è in grado di eseguire tutti gli inserimenti da questa applicazione e supponendo che nessuna altra applicazione stia inserendo messaggi nella coda, nessun altro gestore code nel gruppo di condivisione code è in grado di svuotarla. Se tuttavia non è presente alcun MQGET con attesa sul gestore code locale e viene inserito un messaggio nella coda condivisa, la CF notificherà gli altri gestori code nel gruppo di condivisione code in base alle relative regole per le notifiche delle transizioni di elenco.

2. MQPUT di un messaggio persistente o in - syncpoint.

   In questo caso, se è presente un'applicazione con un *MQGET con attesa* sul gestore code locale, il messaggio viene inserito nella coda condivisa e la CF notifica gli altri gestori code nel gruppo di condivisione code in base alle relative regole per le notifiche delle transizioni di elenco. Tuttavia, il gestore code locale non attende una notifica di transizione dalla CF ma rispetta prima qualsiasi *MQGET con attesa* locale e generalmente esegue il richiamo di questo messaggio per conto dell'applicazione prima che qualsiasi altro gestore code nel gruppo di condivisione code possa rispondere a una notifica CF. Ciò dipende da quanto è occupato il gestore code locale. In caso contrario, qualsiasi gestore code notificato dalla CF a causa dell'arrivo del messaggio sulla coda vuota tenterà di eseguire prima il richiamo. Il primo gestore code a rispondere elabora il nuovo messaggio.

3. Infine, se la coda non è svuotata di messaggi, dove la CF ha inviato una notifica di una modifica dello stato da vuoto a non vuoto per la coda, tutti i gestori code connessi avranno l'opportunità di assistere nell'elaborazione della coda. In questo evento, il carico di lavoro viene definito *basato sull'estrazione*.

Questo design consente di migliorare le prestazioni su una distribuzione del carico di lavoro basata esclusivamente sul pull. L'obiettivo è sfruttare l'alta disponibilità offerta dalle code contenute nella CF consentendo al gestore code, laddove possibile, di eseguire MQGET senza dover fare riferimento alla CF e quindi di elaborare il carico di lavoro del messaggio nel modo più efficiente possibile.

Gli approcci alternativi possono essere adottati quando l'enfasi sul bilanciamento del carico di lavoro è più importante dei miglioramenti delle prestazioni descritti in precedenza. Ad esempio, verificare che nessuna delle applicazioni di richiamo sia connessa al gestore code a cui è connessa l'applicazione di inserimento. Utilizzando questa struttura, tutti i messaggi vengono inseriti nella coda e tutti i gestori code in QSG vengono avvisati quando la coda passa da vuota a non vuota, in accordo all'algoritmo CF per la gestione di tali transizioni. Inoltre, il meccanismo *fast put to waiting getter* non è applicabile.

z/OS Dove trovare ulteriori informazioni sulle code condivise e sui gruppi di condivisione code

Utilizzare la tabella in questo argomento per trovare ulteriori informazioni su come IBM MQ for z/OS utilizza le code condivise e i gruppi di condivisione code.

Tabella 19. Dove trovare ulteriori informazioni sulle code condivise e sui gruppi di condivisione code

Argomento	**Dove cercare**
Ripristino del gruppo di condivisione code	“Ripristino e riavvio su z/OS” a pagina 258
Sicurezza del gruppo di condivisione code	“Concetti di sicurezza in IBM MQ for z/OS” a pagina 275
Definizioni di oggetti privati e globali Indirizzamento dei comandi a una coda diversa IT	Origini da cui è possibile immettere comandi su z/OS

Tabella 19. Dove trovare ulteriori informazioni sulle code condivise e sui gruppi di condivisione code (Continua)

Argomento	Dove cercare
Pianificazione della CF (coupling facility) on demand	Definizione delle risorse CF
Pianificazione dell'ambiente SMDS	Pianificazione dell'ambiente SMDS (shared message data set)
Pianificazione Db2 ambiente	Pianificazione dell'ambiente Db2
Impostazione delle code condivise Parametri di sistema	“Code condivise e gruppi di condivisione code” a pagina 171
Programmi di utilità Migrazione delle code	IBM MQ utilities on z/OS riferimento
Messaggi della console	Messaggi per IBM MQ for z/OS
Comandi MQSC	Comandi MQSC
IBM MQCluster	Configurazione di un cluster di gestori code
IBM MQ accodamento distribuito Nomi canale	Introduzione alla gestione delle code distribuite
Scrittura di applicazioni	Panoramica sulla progettazione dell'applicazione
chiamata MQCONNX	MQCONNX

z/OS Accodamento all'interno del gruppo

Questa sezione descrive l'accodamento all'interno del gruppo, una funzione IBM MQ for z/OS univoca per la piattaforma z/OS . Questa funzione è disponibile solo per i gestori code definiti per un gruppo di condivisione code.

Per informazioni sui gruppi di condivisione code, consultare “Code condivise e gruppi di condivisione code” a pagina 171.

Concetti di accodamento all'interno del gruppo

È possibile eseguire il trasferimento rapido di messaggi tra gestori code in un gruppo di condivisione code senza definire canali. Utilizza una coda di sistema denominata SYSTEM.QSG.TRANSMIT.QUEUE, che è una coda di trasmissione condivisa. Ogni gestore code nel gruppo di condivisione code avvia un'attività denominata agent di accodamento all'interno del gruppo, che attende l'arrivo di messaggi su questa coda destinati al relativo gestore code. Quando viene rilevato un messaggio di questo tipo, viene rimosso dalla coda e posizionato sulla corretta coda di destinazione.

Le regole di risoluzione dei nomi standard vengono utilizzate ma, se IGQ (intra - group queuing) è abilitata e il gestore code di destinazione è all'interno del gruppo di condivisione code, il SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE viene utilizzato per trasferire il messaggio al gestore code di destinazione corretto invece di utilizzare una coda di trasmissione e un canale.

Abilitare l'accodamento all'interno del gruppo tramite un attributo del gestore code. L'accodamento all'interno del gruppo sposta i messaggi non persistenti all'esterno del punto di sincronizzazione e i messaggi persistenti all'interno del punto di sincronizzazione. Se rileva un problema durante la consegna dei messaggi alla coda di destinazione, l'accodamento all'interno del gruppo tenta di inserirli nella coda di messaggi non recapitabili. Se la coda di messaggi non recapitabili è piena o non definita, i messaggi non persistenti vengono eliminati, ma i messaggi persistenti vengono ripristinati e restituiti al SISTEMA SYSTEM.QSG.TRANSMIT.QUEUEe l'agent IGQ tenta di consegnare i messaggi fino a quando non viene eseguito correttamente.

Un canale condiviso in entrata che riceve un messaggio destinato a una coda su un gestore code differente nel gruppo di condivisione code può utilizzare l'accodamento all'interno del gruppo per *passare* il messaggio alla destinazione corretta.

Ci potrebbero essere momenti in cui si desidera che il gestore code locale inserisci un messaggio direttamente nella coda di destinazione, se la coda di destinazione è una coda condivisa, piuttosto che il messaggio viene prima trasferito al gestore code di destinazione. È possibile utilizzare l'attributo del gestore code SQQMNAME per controllarlo. Se si imposta il valore SQQMNAME su USE, il comando MQOPEN viene eseguito sul gestore code specificato da **ObjectQMgrName**.

Tuttavia, se la coda di destinazione è una coda condivisa e si imposta il valore di SQQMNAME su IGNORE e **ObjectQMgrName** è quello di un altro gestore code nel gruppo di condivisione code, la coda condivisa viene aperta sul gestore code locale. Se il gestore code locale non può aprire la coda di destinazione o inserire un messaggio nella coda, il messaggio viene trasferito al **ObjectQMgrName** specificato tramite IGQ o un canale IBM MQ .

L'accodamento all'interno di un gruppo può essere utilizzato per consegnare, in modo più efficiente, piccoli messaggi alle code che risiedono su gestori code remoti all'interno di un gruppo di condivisione code. L'accodamento all'interno del gruppo supporta anche messaggi di grandi dimensioni, il più grande è 100 MB *meno* la lunghezza dell'intestazione della coda di trasmissione.

Nota: Se si utilizza questa funzione, gli utenti devono avere lo stesso accesso alle code su ciascun gestore code nel gruppo di condivisione code.

Il seguente diagramma mostra un tipico esempio di accodamento all'interno del gruppo.

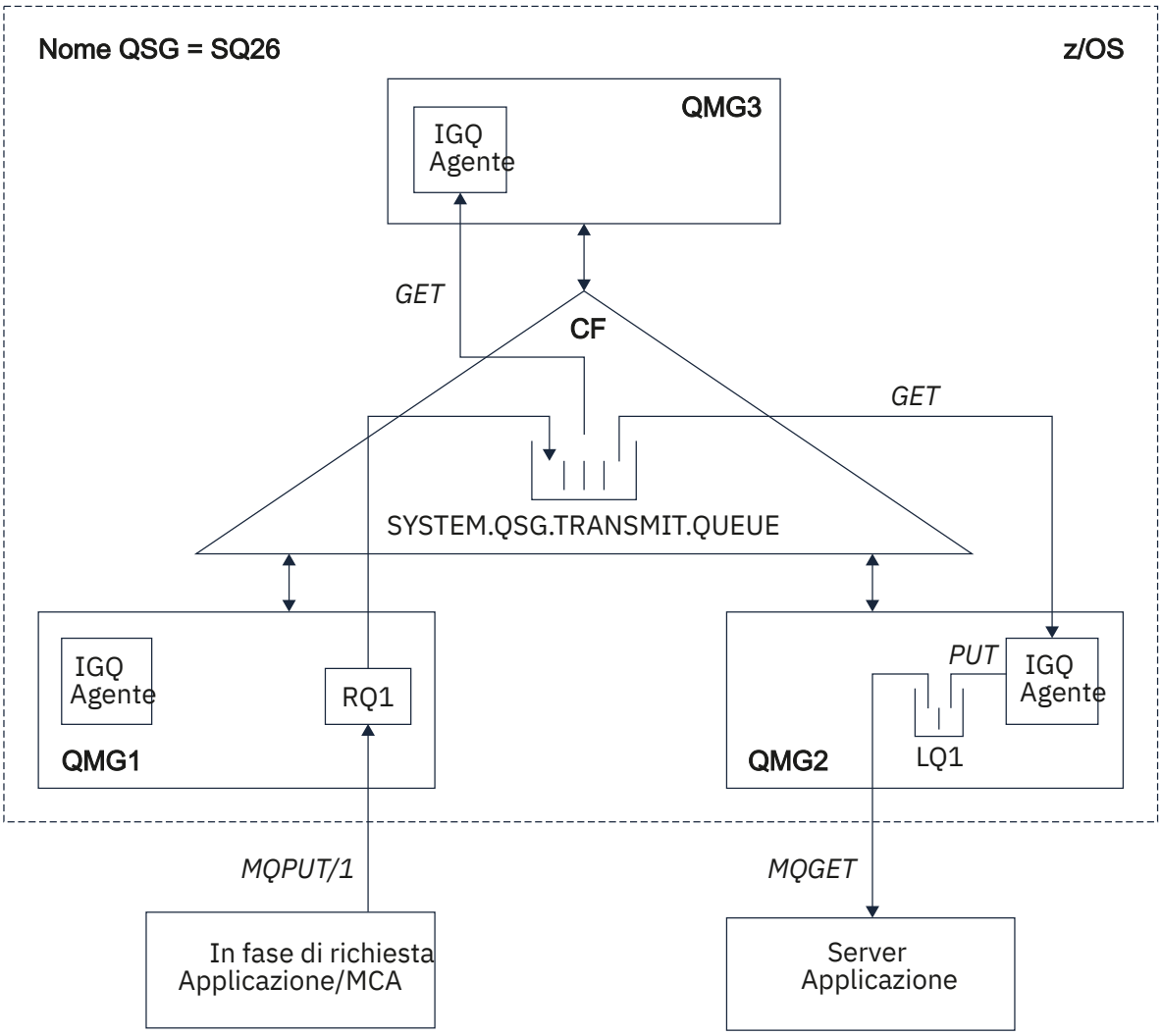


Figura 67. Un esempio di accodamento all'interno del gruppo

Il diagramma mostra:

- agent IGQ in esecuzione su tre gestori code (QMG1, QMG2e QMG3) definiti in un gruppo di condivisione code denominato SQ26.
- Coda di trasmissione condivisa SYSTEM.QSG.TRANSMIT.QUEUE definito nella CF (coupling facility).
- Una definizione di coda remota definita nel gestore code QMG1.
- Una coda locale definita nel gestore code QMG2.
- Un'applicazione richiedente (questa applicazione potrebbe essere un MCA (Message Channel Agent) connesso al gestore code QMG1.
- Un'applicazione server connessa al gestore code QMG2.
- Un messaggio di richiesta inserito nel SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE.

Accodamento all'interno del gruppo e agent di accodamento all'interno del gruppo

Un agent IGQ viene avviato durante l'inizializzazione del gestore code. Quando le applicazioni aprono e inseriscono i messaggi nelle code remote, il gestore code locale determina se l'accodamento all'interno del gruppo viene utilizzato per il trasferimento dei messaggi. Se è necessario utilizzare l'accodamento all'interno del gruppo, il gestore code locale inserisce il messaggio nel SISTEMA

SYSTEM.QSG.TRANSMIT.QUEUE. L'agent IGQ sul gestore code remoto di destinazione richiama i messaggi e li inserisce nella coda di destinazione.

Terminologia accodamento all'interno del gruppo

Spiegazioni della terminologia: accodamento all'interno del gruppo, coda di trasmissione condivisa da utilizzare nell'accodamento all'interno del gruppo e agent di accodamento all'interno del gruppo.

Accodamento all'interno del gruppo

L'accodamento all'interno di un gruppo può determinare un trasferimento di messaggi potenzialmente rapido e meno costoso tra i gestori code in un gruppo di condivisione code, senza la necessità di definire i canali.

Coda di trasmissione condivisa da utilizzare per l'accodamento all'interno del gruppo

Ogni gruppo di condivisione code ha una coda di trasmissione condivisa denominata SYSTEM.QSG.TRANSMIT.QUEUE per l'utilizzo da parte dell'accodamento all'interno del gruppo. Se l'accodamento all'interno del gruppo è abilitato, SYSTEM.QSG.TRANSMIT.QUEUE viene visualizzato nel percorso di risoluzione del nome durante l'apertura di code remote. Quando le applicazioni (inclusi gli MCA (Message Channel Agent)) inseriscono i messaggi in una coda remota, il gestore code locale determina l'idoneità dei messaggi per il trasferimento rapido e li colloca in SYSTEM.QSG.TRANSMIT.QUEUE.

Agent di accodamento all'interno del gruppo

L'agent IGQ è l'attività, avviata durante l'inizializzazione del gestore code, che attende l'arrivo di messaggi adeguati sul sistema SYSTEM.QSG.TRANSMIT.QUEUE. L'agente IGQ richiama i messaggi adatti da questa coda e li consegna alle code di destinazione.

L'agent IGQ per ciascun gestore code è sempre avviato perché l'accodamento all'interno del gruppo viene utilizzato dal gestore code stesso per la propria elaborazione interna.

z/OS Vantaggi dell'accodamento intragruppo

I vantaggi dell'accodamento all'interno del gruppo sono: definizioni di sistema ridotte, gestione del sistema ridotta, prestazioni migliorate, supporto della migrazione e distribuzione dei messaggi quando si passa da un gestore code all'altro in un gruppo di condivisione code.

I vantaggi dell'accodamento intragruppo sono:

Definizioni di sistema ridotte
L'accodamento all'interno del gruppo elimina la necessità di definire i canali tra i gestori code in un gruppo di condivisione code.

Amministrazione del sistema ridotta
Poiché non vi sono canali definiti tra i gestori code in un gruppo di condivisione code, non vi è alcun requisito per la gestione dei canali.

Migliorate performance
Poiché esiste solo un agente IGQ necessario per la consegna di un messaggio a una coda di destinazione (invece di due agenti mittenti e riceventi intermedi), la consegna dei messaggi utilizzando l'accodamento all'interno del gruppo può essere meno costosa della consegna dei messaggi utilizzando i canali. Nell'accodamento all'interno del gruppo c'è solo un componente di ricezione, perché la necessità del componente di invio è stata rimossa. Questo salvataggio è dovuto al fatto che il messaggio è disponibile per l'agent IGQ sul gestore code di destinazione per la consegna alla coda di destinazione una volta che l'operazione di inserimento sul gestore code locale è stata completata e, nel caso di messaggi inseriti nell'ambito del punto di sincronizzazione, è stato eseguito il commit.

Supporta la migrazione

Le applicazioni esterne a un gruppo di condivisione code possono consegnare messaggi a una coda che si trova su qualsiasi gestore code nel gruppo di condivisione code, mentre sono connesse solo a un particolare gestore code nel gruppo di condivisione code. Questo perché i messaggi che arrivano su un canale ricevente, destinati a una coda su un gestore code remoto, possono essere inviati in maniera trasparente alla coda di destinazione utilizzando l'accodamento all'interno del gruppo. Questa funzione consente alle applicazioni di essere distribuite tra il gruppo di condivisione code senza dover modificare i sistemi esterni al gruppo di condivisione code.

Una configurazione tipica è illustrata dal diagramma seguente, in cui:

- Un'applicazione richiedente connessa al gestore code QMG1 deve inviare un messaggio a una coda locale sul gestore code QMG3.
- Il gestore code QMG1 è connesso solo al gestore code QMG2.
- I gestori code QMG2 e QMG3, precedentemente connessi utilizzando i canali, sono ora membri del gruppo di condivisione code SQ26.

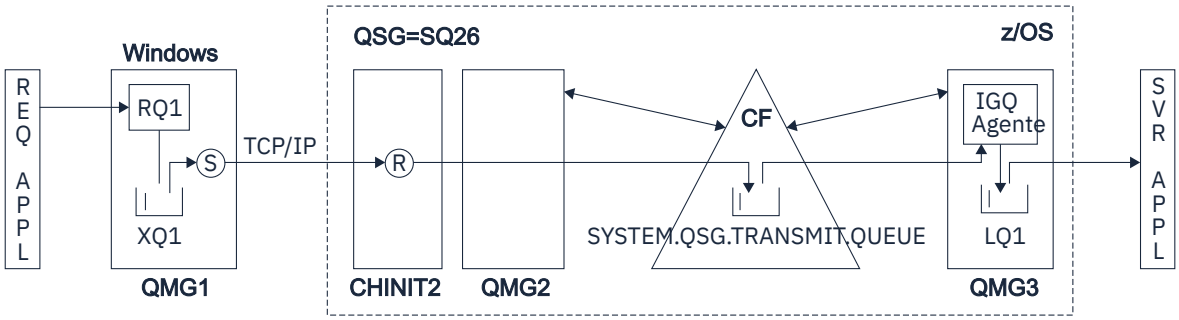


Figura 68. Un esempio di supporto di migrazione

Il flusso delle operazioni è il seguente:

1. L'applicazione richiedente inserisce un messaggio, destinato alla coda locale LQ1 sul gestore code remoto QMG3, nella definizione della coda remota RQ1.
2. Il gestore code QMG1, in esecuzione su una workstation Windows NT , inserisce il messaggio nella coda di trasmissione XQ1.
3. Il mittente MCA (S) su QM1 trasmette il messaggio, utilizzando TCP/IP, al destinatario MCA (R) sull'iniziatore del canale CHINIT2.
4. Il ricevitore MCA (R) sull'iniziatore del canale CHINIT2 inserisce il messaggio nella coda di trasmissione condivisa SYSTEM.QSG.TRANSMIT.QUEUE.
5. L'agent IGQ sul gestore code QMG3 richiama il messaggio da SYSTEM.QSG.TRANSMIT.QUEUE e la inserisce nella coda locale di destinazione LQ1.
6. L'applicazione server richiama il messaggio dalla coda locale di destinazione e lo elabora.

Recapito di messaggi quando si passa tra i gestori code in un gruppo di condivisione code

Il diagramma precedente in Supporta la migrazione illustra anche la consegna dei messaggi quando si passa da un gestore code all'altro in un gruppo di condivisione code. I messaggi in arrivo su un gestore code all'interno del gruppo di condivisione code, ma destinati a una coda su un altro gestore code nel gruppo di condivisione code, possono essere facilmente trasmessi alla coda di destinazione sul gestore code di destinazione, utilizzando l'accodamento all'interno del gruppo.

z/OS Limitazioni dell'accodamento all'interno del gruppo

Le limitazioni dell'accodamento all'interno del gruppo sono: i messaggi idonei per il trasferimento utilizzando l'accodamento all'interno del gruppo, il numero di agent di accodamento all'interno del gruppo per gestore code e l'avvio e l'arresto dell'agent di accodamento all'interno del gruppo.

Questo argomento descrive le limitazioni dell'accodamento all'interno del gruppo.

Messaggi idonei per il trasferimento utilizzando l'accodamento all'interno del gruppo
Poiché l'accodamento all'interno del gruppo utilizza una coda di trasmissione condivisa definita nella CF (Coupling Facility), l'accodamento all'interno del gruppo è limitato alla consegna di messaggi della lunghezza massima supportata per le code condivise meno la lunghezza di un'intestazione della coda di trasmissione (MQXQH).

Numero di agent di accodamento all'interno del gruppo per gestore code
Viene avviato un solo agent IGQ per gestore code in un gruppo di condivisione code.

Avvio e arresto dell'agent di accodamento all'interno del gruppo
L'agent IGQ viene avviato durante l'inizializzazione del gestore code e terminato durante l'arresto del gestore code. È progettato per essere un compito di lunga durata, auto - recupero (in caso di fine anomala),. Se si verifica un errore con la definizione di SYSTEM.QSG.TRANSMIT.QUEUE (ad esempio, se questa coda è Get inibito) l'agent IGQ continua a ripetere l'operazione. Se l'agent IGQ rileva un errore che risulta in una normale chiusura dell'agent mentre il gestore code è ancora attivo, è possibile riavviarlo emettendo un comando ALTER QMGR IGQ (ENABLED). Questo comando evita la necessità di riciclare il gestore code.

Impostazione dell'attributo del gestore code IGQ su ENABLED o DISABLED
Se l'attributo del gestore code IGQ è impostato su ENABLED o DISABLED, gli handle dell'oggetto esistenti potrebbero essere invalidati con il codice motivo MQRC_OBJECT_CHANGED. Per ulteriori informazioni, vedi Introduzione all'accodamento all'interno del gruppo .

z/OS Introduzione all'accodamento all'interno del gruppo

È possibile abilitare, disabilitare e utilizzare l'accodamento all'interno del gruppo come descritto in questo argomento.

Abilitazione dell'accodamento all'interno del gruppo
Per abilitare l'accodamento all'interno dei gruppi sui gestori code, è necessario effettuare le seguenti operazioni:

- Definire una coda di trasmissione condivisa denominata SYSTEM.QSG.TRANSMIT.QUEUE. La definizione di questa coda si trova in thlqual.SCSQPROC(CSQ4INSS), l'esempio CSQINP2 per oggetti SYSTEM per i gruppi di condivisione code. Questa coda deve essere definita con gli attributi corretti, come indicato in thlqual.SCSQPROC(CSQ4INSS), per il corretto funzionamento dell'accodamento all'interno del gruppo.
- Poiché l'agent IGQ viene sempre avviato durante l'inizializzazione del gestore code, l'accodamento all'interno del gruppo è sempre disponibile per l'elaborazione dei messaggi in entrata. L'agent IGQ elabora tutti i messaggi collocati nel SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE. Tuttavia, per abilitare l'accodamento all'interno del gruppo per l'elaborazione in uscita, l'attributo del gestore code IGQ deve essere impostato su ENABLED.

  Importante: Se l'attributo del gestore code IGQ è impostato su ENABLED, gli handle di oggetto esistenti potrebbero essere invalidati con il codice motivo MQRC_OBJECT_CHANGED. Per ulteriori informazioni, consultare “Proprietà specifiche dell'accodamento all'interno del gruppo” a pagina 230. Come descritto nella sezione 'Risposta del programmatore' per questo codice motivo, le applicazioni devono essere codificate per gestire questa situazione (consultare 2041 (07F9) (RC2041): MQRC_OBJECT_CHANGED per ulteriori dettagli).

Inoltre, poiché IGQ è progettato come un'attività di lunga esecuzione e di ripristino automatico, che inizia durante l'inizializzazione e termina con l'arresto, consultare “Limitazioni dell'accodamento all'interno del gruppo” a pagina 221 per ulteriori informazioni.

Disabilitazione dell'accodamento all'interno del gruppo

Per disabilitare l'accodamento all'interno del gruppo per il trasferimento di messaggi in uscita, impostare l'attributo del gestore code IGQ su DISABLED. Se l'accodamento all'interno del gruppo è disabilitato per un particolare gestore code, l'agent IGQ su tale gestore code può comunque elaborare i messaggi in entrata che sono stati inseriti nel SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE

da un gestore code che non ha l'accodamento all'interno del gruppo abilitato per il trasferimento in uscita.

Importante: Se l'attributo del gestore code IGQ è impostato su ENABLED, gli handle di oggetto esistenti potrebbero essere invalidati con il codice motivo MQRC_OBJECT_CHANGED. Per ulteriori informazioni, consultare “Proprietà specifiche dell'accodamento all'interno del gruppo” a pagina 230. Come descritto nella sezione 'Risposta del programmatore' per questo codice motivo, le applicazioni devono essere codificate per gestire questa situazione (consultare 2041 (07F9) (RC2041): MQRC_OBJECT_CHANGED per ulteriori dettagli).

Inoltre, poiché IGQ è progettato come un'attività di lunga esecuzione e di ripristino automatico, che inizia durante l'inizializzazione e termina con l'arresto, consultare “Limitazioni dell'accodamento all'interno del gruppo” a pagina 221 per ulteriori informazioni.

Utilizzo dell'accodamento all'interno del gruppo

Una volta abilitata, l'accodamento all'interno del gruppo è disponibile per l'uso e un gestore code lo utilizza quando possibile. Vale a dire, quando un'applicazione inserisce un messaggio in una definizione di coda remota, in una coda remota completa o in una coda cluster, il gestore code determina se il messaggio è idoneo per essere consegnato utilizzando l'accodamento all'interno del gruppo e, se lo è, lo inserisce in SYSTEM.QSG.TRANSMIT.QUEUE. Non è necessario modificare le applicazioni utente o le code dell'applicazione, poiché per i messaggi idonei il gestore code utilizza il SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE, al posto di qualsiasi altra coda di trasmissione.

z/OS Configurazioni di accodamento all'interno del gruppo

Oltre alla tipica configurazione di accodamento all'interno del gruppo, sono possibili altre configurazioni.

“Concetti di accodamento all'interno del gruppo” a pagina 217 descrive la tipica configurazione.

Concetti correlati

“Accodamento distribuito con accodamento all'interno del gruppo (percorsi di consegna multipli)” a pagina 223
Per le applicazioni che elaborano messaggi brevi, potrebbe essere possibile configurare l'accodamento all'interno del gruppo solo per la consegna di messaggi tra gestori code in un gruppo di condivisione code.

“Clustering con accodamento all'interno del gruppo (più percorsi di consegna)” a pagina 225
È possibile configurare i gestori code in modo che si trovino in un cluster e in un gruppo di condivisione code.

“Clustering, accodamento all'interno del gruppo e accodamento distribuito” a pagina 228
È possibile configurare un gestore code che sia un membro di un cluster e un gruppo di condivisione code e che sia connesso a un gestore code distribuito utilizzando una coppia di canali mittente / destinatario.

z/OS *Accodamento distribuito con accodamento all'interno del gruppo (percorsi di consegna multipli)*

Per le applicazioni che elaborano messaggi brevi, potrebbe essere possibile configurare l'accodamento all'interno del gruppo solo per la consegna di messaggi tra gestori code in un gruppo di condivisione code.

La scelta dell'accodamento all'interno del gruppo sulle comunicazioni del canale può essere controllata dal livello tipo CFSTRUCT. (3 invece di 4 o 5). La lunghezza massima del messaggio impostata sul SISTEMA SYSTEM.QSQ.TRANSMIT.QUEUE.

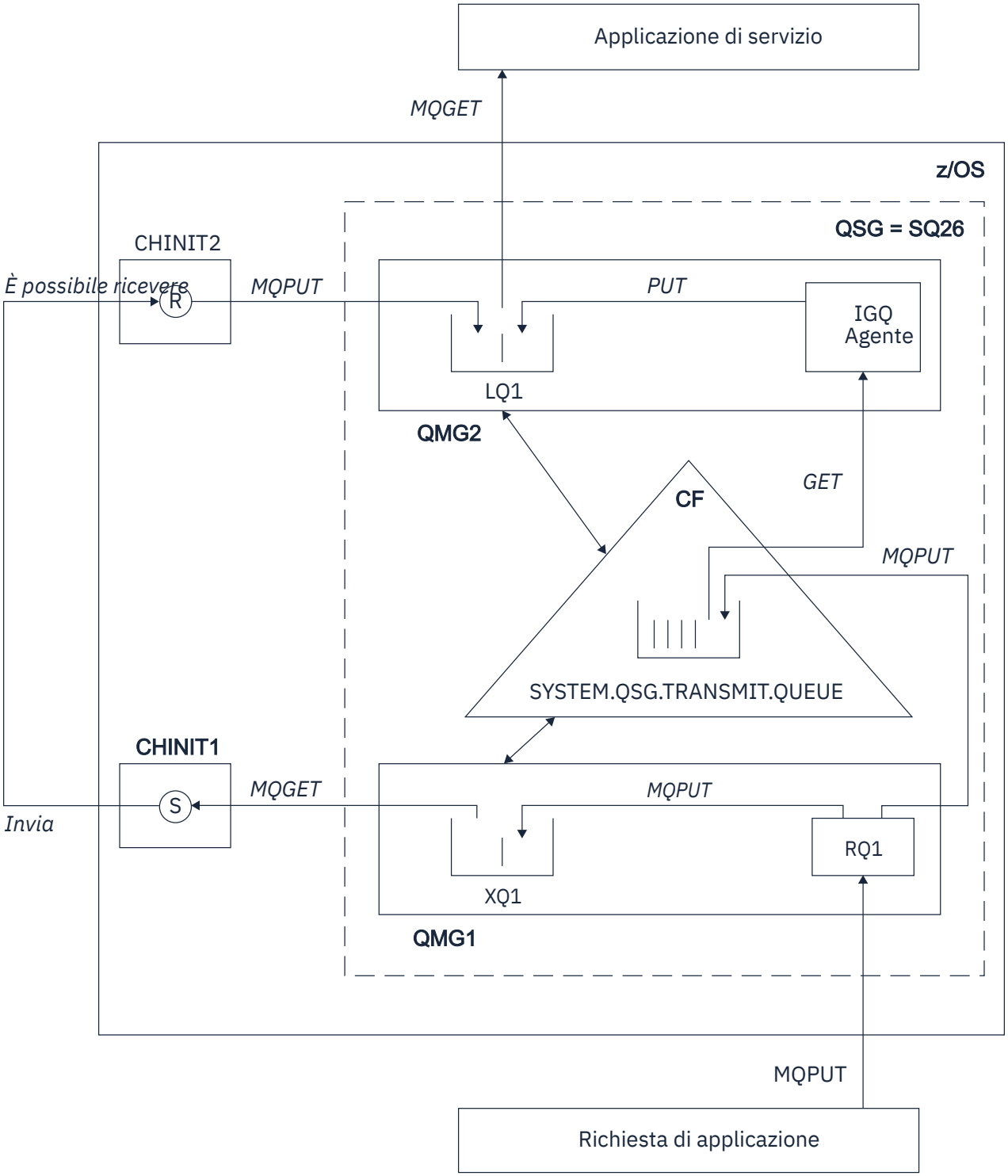


Figura 69. Una configurazione di esempio

Apri / Inserisci elaborazione

1. È importante notare che quando l'applicazione richiedente apre la coda remota RQ1, la risoluzione del nome si verifica sia per la coda di trasmissione non condivisa XQ1 che per la coda di trasmissione condivisa SYSTEM.QSG.TRANSMIT.QUEUE.
2. Quando l'applicazione richiedente inserisce un messaggio nella coda remota, in base al fatto che l'accodamento all'interno del gruppo sia abilitato per il trasferimento in uscita sul gestore code e sulle caratteristiche del messaggio, il messaggio viene inserito nella coda di trasmissione XQ1o nella

coda di trasmissione SYSTEM.QSG.TRANSMIT.QUEUE. Il gestore code inserisce tutti i messaggi di grandi dimensioni nella coda di trasmissione XQ1e tutti i messaggi di piccole dimensioni nella coda di trasmissione SYSTEM.QSG.TRANSMIT.QUEUE.

3. Se la coda di trasmissione XQ1 è piena o non è disponibile, le richieste di inserimento di messaggi di grandi dimensioni non riescono in modo sincrono con un codice di ritorno e di errore appropriato. Tuttavia, le richieste di inserimento di piccoli messaggi continuano ad avere esito positivo e vengono inserite nella coda di trasmissione SYSTEM.QSG.TRANSMIT.QUEUE.
4. Se la coda di trasmissione SYSTEM.QSG.TRANSMIT.QUEUE è pieno, o non può essere inserito, le richieste di inserimento di messaggi di piccole dimensioni non riescono in modo sincrono con un codice di ritorno e di errore adatto. Tuttavia, le richieste di inserimento di messaggi di grandi dimensioni continuano ad avere esito positivo e vengono inserite nella coda di trasmissione XQ1. In tal caso, non viene eseguito alcun tentativo di inserire i messaggi di piccole dimensioni in una coda di trasmissione.

Flusso per messaggi di grandi dimensioni

1. L'applicazione richiedente inserisce messaggi di grandi dimensioni nella coda remota RQ1.
2. Il gestore code QMG1 inserisce i messaggi nella coda di trasmissione XQ1.
3. Il mittente MCA (S) sul gestore code QMG1 richiama i messaggi dalla coda di trasmissione XQ1 e li invia al gestore code QMG2.
4. Il ricevitore MCA (R) sul gestore code QMG2 riceve i messaggi e li inserisce nella coda di destinazione LQ1.
5. L'applicazione utilizzata richiama ed elabora i messaggi dalla coda LQ1.

Flusso per messaggi di piccole dimensioni

1. L'applicazione richiedente inserisce piccoli messaggi nella coda remota RQ1.
2. Il gestore code QMG1 inserisce i messaggi nella coda di trasmissione SYSTEM.QSG.TRANSMIT.QUEUE.
3. IGQ sul gestore code QMG2 richiama i messaggi e li inserisce nella coda di destinazione LQ1.
4. L'applicazione utilizzata richiama i messaggi dalla coda LQ1.

Punti da annotare

1. L'applicazione richiedente non deve essere a conoscenza del meccanismo sottostante utilizzato per la consegna dei messaggi.
2. È possibile ottenere un meccanismo di consegna dei messaggi potenzialmente più rapido per i messaggi di piccole dimensioni.
3. Sono disponibili più percorsi per il recapito dei messaggi (ovvero, l'instradamento del canale normale e l'instradamento di accodamento all'interno del gruppo).
4. L'instradamento di accodamento all'interno del gruppo, essendo potenzialmente più veloce, viene selezionato in preferenza al normale instradamento del canale. A seconda delle caratteristiche del messaggio, la distribuzione del messaggio potrebbe essere divisa tra i due percorsi. Quindi, i messaggi potrebbero essere consegnati fuori sequenza (anche se questa consegna è possibile anche se i messaggi vengono consegnati utilizzando solo il normale instradamento del canale).
5. Quando un instradamento è stato selezionato e i messaggi sono stati inseriti nelle code di trasmissione, solo l'instradamento selezionato viene utilizzato per la consegna del messaggio. Tutti i messaggi non elaborati sul SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE non vengono deviate nella coda di trasmissione XQ1.

z/OS *Clustering con accodamento all'interno del gruppo (più percorsi di consegna)*

È possibile configurare i gestori code in modo che si trovino in un cluster e in un gruppo di condivisione code.

Quando i messaggi vengono inviati a una coda cluster e i gestori code di destinazione locali e remoti si trovano nello stesso gruppo di condivisione code, l'accodamento all'interno del gruppo viene utilizzato per la consegna di piccoli messaggi (utilizzando il SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE) e la consegna di messaggi di grandi dimensioni se l'accodamento all'interno del gruppo supporta la dimensione del messaggio. Inoltre, il SISTEMA SYSTEM.CLUSTER.TRANSMIT.QUEUE viene utilizzato per il recapito dei messaggi a qualsiasi gestore code presente nel cluster, ma esterno al gruppo di condivisione code. Il seguente diagramma illustra questa configurazione (gli iniziatori di canali non sono visualizzati).

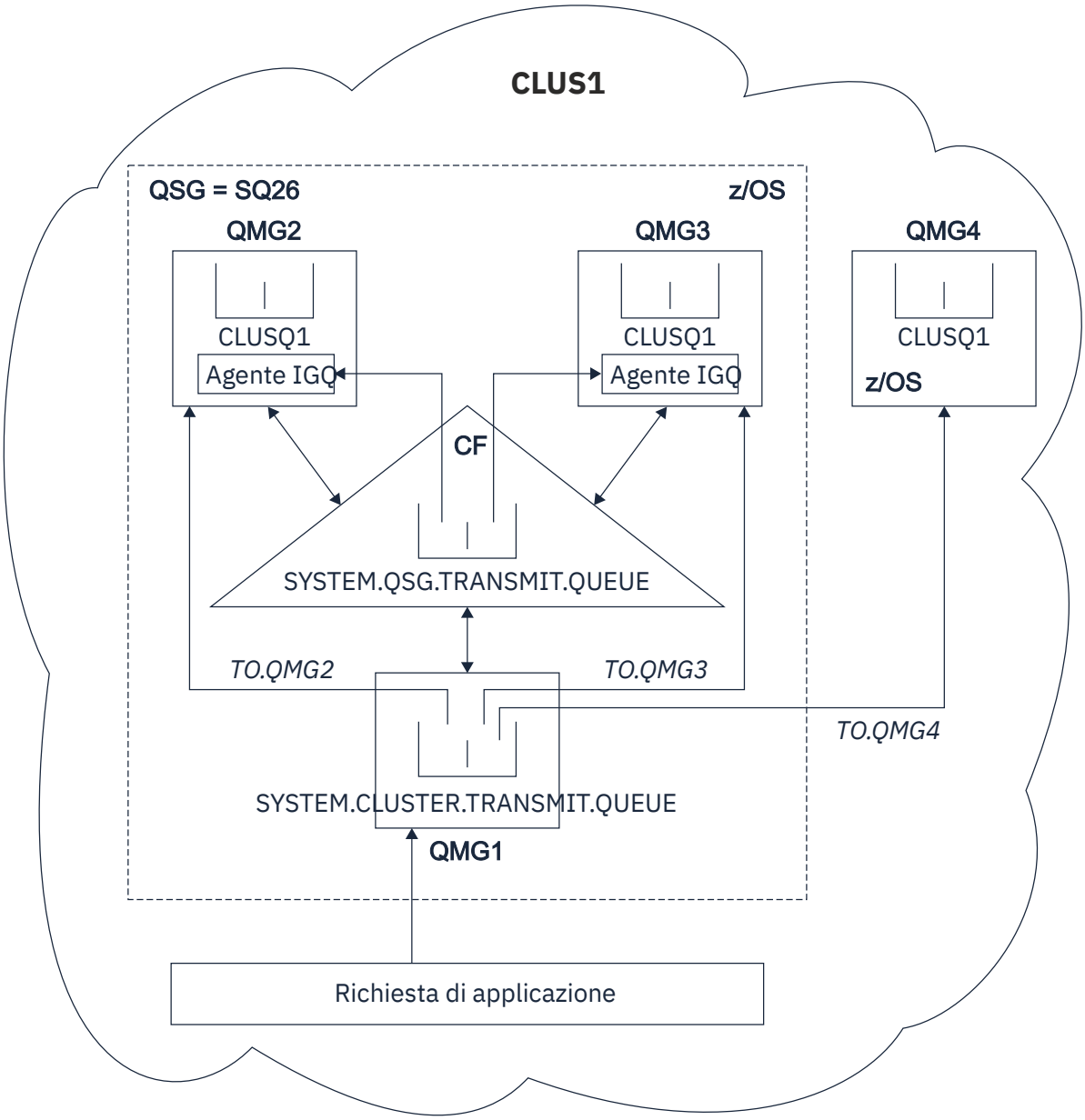


Figura 70. Un esempio di cluster con accodamento all'interno del gruppo

Il diagramma mostra:

- Quattro gestori code z/OS QMG1, QMG2, QMG3e QMG4 configurati in un cluster CLUS1.
- Gestori code QMG1, QMG2e QMG3 configurati in un gruppo di condivisione code SQ26.
- Agent IGQ in esecuzione sui gestori code QMG2 e QMG3.
- Il SYSTEM.CLUSTER.TRANSMIT.QUEUE definito in QMG1.

Nota: Per chiarezza, il SISTEMA SYSTEM.CLUSTER.TRANSMIT.QUEUE sugli altri gestori code non visualizzati.

- Il sistema SYSTEM.QSG.TRANSMIT.QUEUE definita nella CF, che si trova in una struttura IBM MQ configurata con l'attributo CFLEVEL (3) RECOVER (YES).
- Canali cluster TO.QMG2 (collegamento di QMG1 a QMG2), TO.QMG3 (connessione di QMG1 a QMG3) e TO.QMG4 (connessione di QMG1 a QMG4).
- La coda cluster CLUSQ1 risiede sui gestori code QMG2, QMG3e QMG4.

Si supponga che l'applicazione richiedente apra la coda cluster con l'opzione MQOO_BIND_NOT_FIXED, in modo che il gestore code di destinazione per la coda cluster sia selezionato al momento dell'inserimento.

Se il gestore code di destinazione è QMG2:

- Tutti i messaggi di grandi dimensioni immessi dall'applicazione richiedente sono:
  - Inserire in SYSTEM.CLUSTER.TRANSMIT.QUEUE su QMG1, perché SYSTEM.QSG.TRANSMIT.QUEUE è in una struttura CFLEVEL (3); pertanto supporta solo messaggi di dimensione fino a 63 KB.
  - Trasferito alla coda cluster CLUSQ1 su QMG2 utilizzando il canale cluster TO.QMG2
- Tutti i piccoli messaggi immessi dall'applicazione richiedente sono
  - Inserire nella coda di trasmissione condivisa SYSTEM.QSG.TRANSMIT.QUEUE. Questa coda si trova in una struttura configurata con l'attributo RECOVER (YES), quindi viene utilizzata per piccoli messaggi persistenti e non persistenti.
  - Richiamato dall'agent IGQ su QMG2
  - Inserire nella coda cluster CLUSQ1 su QMG2

Se il gestore code di destinazione è QMG4:

- Poiché QMG4 non è un membro del gruppo di condivisione code SQ26, tutti i messaggi immessi dall'applicazione richiedente sono
  - Inserire in SYSTEM.CLUSTER.TRANSMIT.QUEUE su QMG1
  - Trasferito alla coda cluster CLUSQ1 su QMG4 utilizzando il canale cluster TO.QMG4

Punti da annotare

- L'applicazione richiedente non deve essere a conoscenza del meccanismo sottostante utilizzato per la consegna dei messaggi.
- Un meccanismo di consegna potenzialmente più rapido si ottiene per il trasferimento di piccoli messaggi non persistenti tra gestori code in un gruppo di condivisione code (anche se gli stessi gestori code si trovano in un cluster).
- Sono disponibili più percorsi per il recapito del messaggio (ovvero, sia l'instradamento cluster che l'instradamento di accodamento all'interno del gruppo).
- L'instradamento di accodamento all'interno del gruppo, essendo potenzialmente più veloce, viene selezionato in preferenza all'instradamento del cluster. A seconda delle caratteristiche del messaggio, la distribuzione del messaggio potrebbe essere divisa tra i due percorsi. Pertanto, i messaggi potrebbero essere consegnati fuori sequenza. È importante notare che questa consegna è possibile senza considerare l'opzione MQOO_BIND_ * specificata dall'applicazione. L'accodamento all'interno del gruppo distribuisce i messaggi nello stesso modo del clustering, a seconda che MQOO_BIND_NOT_FIXED, MQOO_BIND_ON_OPEN, MQOO_BIND_ON_GROUP o MQOOO_BIND_AS_Q_DEF sia specificato su open.
- Quando un instradamento è stato selezionato e i messaggi sono stati inseriti nelle code di trasmissione, solo l'instradamento selezionato viene utilizzato per la consegna del messaggio. Tutti i messaggi non elaborati sul SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE non vengono deviati nel SYSTEM.CLUSTER.TRANSMIT.QUEUE.

z/OS *Clustering, accodamento all'interno del gruppo e accodamento distribuito*

È possibile configurare un gestore code che sia un membro di un cluster e un gruppo di condivisione code e che sia connesso a un gestore code distribuito utilizzando una coppia di canali mittente / destinatario.

Questa configurazione è una combinazione di accodamento distribuito con accodamento all'interno del gruppo e clustering con accodamento all'interno del gruppo.

L'accodamento all'interno del gruppo è descritto in "Accodamento distribuito con accodamento all'interno del gruppo (percorsi di consegna multipli)" a pagina 223.

Il clustering con accodamento all'interno del gruppo è descritto in "Clustering con accodamento all'interno del gruppo (più percorsi di consegna)" a pagina 225.

z/OS Messaggi di accodamento all'interno del gruppo

Questa sezione descrive i messaggi inseriti nel SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE.

Struttura dei messaggi
Come tutti gli altri messaggi inseriti nelle code di trasmissione, i messaggi inseriti nel SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE ha come prefisso l'intestazione della coda di trasmissione (MQXQH).

Persistenza messaggio

In IBM WebSphere MQ 5.3 e versioni successive, le code condivise supportano sia i messaggi persistenti che quelli non persistenti.

Se il gestore code termina mentre l'agent IGQ sta elaborando messaggi non persistenti o se l'agent IGQ termina in modo anomalo durante l'elaborazione dei messaggi, i messaggi non persistenti elaborati potrebbero andare persi. Le applicazioni devono prendere accordi per il recupero dei messaggi non persistenti, se il loro recupero è necessario.

Se una richiesta di inserimento per un messaggio non persistente, emessa dall'agent IGQ, ha esito negativo in modo imprevisto, il messaggio elaborato viene perso.

Consegna dei messaggi
L'agent IGQ richiama e consegna tutti i messaggi non persistenti all'esterno dell'ambito del punto di sincronizzazione e tutti i messaggi persistenti all'interno dell'ambito del punto di sincronizzazione. In questo caso, l'agent IGQ agisce come coordinatore del punto di sincronizzazione. L'agent IGQ quindi elabora i messaggi non persistenti come il modo in cui i messaggi veloci e non persistenti vengono elaborati su un canale di messaggi. Consultare Messaggi veloci e non persistenti.

Batch dei messaggi
L'agent IGQ utilizza una dimensione batch fissa di 50 messaggi. I messaggi persistenti richiamati all'interno di un batch vengono sottoposti a commit ad intervalli di 50 messaggi. L'agent esegue il commit di un batch composto da messaggi persistenti quando non ci sono più messaggi disponibili per il recupero sul sistema SYSTEM.QSG.TRANSMIT.QUEUE.

Dimensione messaggio
La dimensione massima del messaggio che può essere inserito nel SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE è la lunghezza massima dei messaggi supportati per le code condivise meno la lunghezza di un'intestazione di coda di trasmissione (MQXQH).

Persistenza del messaggio predefinita e priorità del messaggio predefinita
Se SYSTEM.QSG.TRANSMIT.QUEUE si trova nel percorso di risoluzione del nome coda stabilito al momento dell'apertura, quindi per i messaggi inseriti con la persistenza predefinita e la priorità predefinita (o con la persistenza predefinita o la priorità predefinita), le regole normali vengono applicate nella selezione della coda che ha la priorità predefinita e i valori di persistenza utilizzati. (Consultare la sezione IBM MQ messaggi per ulteriori informazioni sulle regole di selezione della coda).

Concetti correlati

"Messaggi non recapitati / non elaborati" a pagina 229
Questo argomento descrive cosa accade ai messaggi non consegnati e non elaborati sul SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE.

“Messaggi di report - Accodamento all'interno del gruppo” a pagina 229
Questo argomento descrive i messaggi di report: conferma di arrivo, conferma di consegna, report di scadenza e report di eccezioni.

z/OS *Messaggi non recapitati / non elaborati*

Questo argomento descrive cosa accade ai messaggi non consegnati e non elaborati sul SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE.

Se un agent IGQ non può consegnare un messaggio alla coda di destinazione, l'agent IGQ:

- Rispetta l'opzione di report MQRO_DISCARD_MSG (se il campo Opzioni di report di MQMD per il messaggio non consegnato indica che deve) ed elimina il messaggio non consegnato.
- Tenta di inserire il messaggio non consegnato nella coda dei messaggi non recapitati per il gestore code di destinazione, se il messaggio non è stato già eliminato. L'agent IGQ prefissa il messaggio con un'intestazione MQDLH (dead letter queue).

Se una coda di messaggi non recapitabili non è definita o se un messaggio non recapitato non può essere inserito nella coda di messaggi non recapitabili e se il messaggio non recapitato è:

- persistenti, l'agent IGQ esegue il backout del batch corrente di messaggi persistenti che sta elaborando ed entra in uno stato di nuovo tentativo. Per ulteriori informazioni, consultare “Proprietà specifiche dell'accodamento all'interno del gruppo” a pagina 230.
- non persistente, l'agent IGQ elimina il messaggio e continua l'elaborazione del messaggio successivo.

Se un gestore code in un gruppo di condivisione code viene terminato prima che il relativo agent IGQ associato abbia avuto il tempo di elaborare tutti i propri messaggi, i messaggi non elaborati rimangono sul SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE fino al successivo avvio del gestore code. L'agente IGQ quindi richiama e consegna i messaggi alle code di destinazione.

Se la CF ha esito negativo prima di tutti i messaggi sul SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE è stata elaborata, tutti i messaggi non persistenti non elaborati vengono persi.

IBM consiglia alle applicazioni di non inserire i messaggi direttamente nelle code di trasmissione. Se un'applicazione inserisce i messaggi direttamente nel SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE, l'agent IGQ potrebbe non essere in grado di elaborare questi messaggi e rimangono sul SYSTEM.QSG.TRANSMIT.QUEUE. Gli utenti devono quindi utilizzare i propri metodi per gestire questi messaggi non elaborati.

z/OS *Messaggi di report - Accodamento all'interno del gruppo*

Questo argomento descrive i messaggi di report: conferma di arrivo, conferma di consegna, report di scadenza e report di eccezioni.

Messaggi di report di conferma di arrivo (COA) /conferma di consegna (COD)
I messaggi COA e COD vengono generati dal gestore code, quando viene utilizzata l'accodamento all'interno del gruppo.

Messaggi del report di scadenza
I messaggi del report di scadenza vengono generati dal gestore code.

Messaggi di report di eccezione

A seconda dell'opzione di report MQRO_ESCEPTION_ * specificata nel campo *Opzioni di report* del descrittore del messaggio per il messaggio non consegnato, l'agent IGQ genera il report di eccezioni richiesto e lo inserisce nella coda di risposta specificata. L'accodamento all'interno del gruppo può essere utilizzato per consegnare il report di eccezioni alla coda di risposta di destinazione.

La persistenza del messaggio di report è uguale alla persistenza del messaggio non consegnato. Se l'agent IGQ non riesce a risolvere il nome della coda di risposta di destinazione o se non riesce a inserire il messaggio di risposta in una coda di trasmissione (per un successivo trasferimento alla coda di risposta di destinazione), tenta di inserire il report di eccezione nella coda di messaggi non recapitabili del gestore code su cui viene generato il messaggio di report. Se non è possibile, se il messaggio non consegnato è:

- persistenti, l'agent IGQ elimina il report di eccezioni, esegue il backout del batch corrente di messaggi ed entra in uno stato di nuovo tentativo. Per ulteriori informazioni, consultare “Proprietà specifiche dell'accodamento all'interno del gruppo” a pagina 230.
- Non persistente, l'agent IGQ elimina il report di eccezioni e continua l'elaborazione del successivo messaggio sul SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE.

z/OS Sicurezza per accodamento all'interno del gruppo

Questo argomento descrive le disposizioni di sicurezza per l'accodamento all'interno del gruppo.

È possibile impostare gli attributi del gestore code IGQAUT (autorizzazione IGQ) e IGQUSER (ID utente agent IGQ) per controllare il livello di controllo di sicurezza eseguito quando l'agent IGQ apre le code di destinazione.

Autorizzazione di accodamento all'interno del gruppo (IGQAUT)

L'attributo IGQAUT può essere impostato per indicare il tipo di controlli di sicurezza da eseguire e quindi per stabilire gli ID utente che devono essere utilizzati dall'agente IGQ quando stabilisce l'autorità per inserire i messaggi sulla coda di destinazione.

L'attributo IGQAUT è analogo all'attributo PUTAUT disponibile nelle definizioni di canali.

Identificativo utente di accodamento all'interno del gruppo (IGQUSER)

L'attributo IGQUSER può essere utilizzato per designare un ID utente che deve essere utilizzato dall'agent IGQ quando stabilisce l'autorità per inserire i messaggi su una coda di destinazione.

L'attributo IGQUSER è analogo all'attributo MCAUSER disponibile sulle definizioni di canale.

z/OS Proprietà specifiche dell'accodamento all'interno del gruppo

Questa sezione descrive le proprietà specifiche dell'accodamento all'interno del gruppo, inclusa l'invalidamento degli handle di oggetti, la funzione di ripristino automatico e di nuovo tentativo dell'agent di accodamento all'interno del gruppo e l'agent di accodamento all'interno del gruppo e la serializzazione.

Invalidazione degli handle di oggetto (MQRC_OBJECT_CHANGED)

Se si rileva che gli attributi di un oggetto sono stati modificati dopo l'apertura dell'oggetto, il gestore code invalida l'handle dell'oggetto con MQRC_OBJECT_CHANGED al successivo utilizzo.

L'accodamento all'interno del gruppo introduce le seguenti regole per l'invalidazione della gestione oggetti:

- Se SYSTEM.QSG.TRANSMIT.QUEUE è stato incluso nel percorso di risoluzione del nome durante l'elaborazione di apertura poiché l'accodamento all'interno del gruppo era ABILITATO al momento dell'apertura, ma l'accodamento all'interno del gruppo è DISABILITATO al momento dell'inserimento, quindi il gestore code invalida l'handle dell'oggetto e non riesce la richiesta di inserimento con MQRC_OBJECT_CHANGED.
- Se SYSTEM.QSG.TRANSMIT.QUEUE non è stato incluso nel percorso di risoluzione del nome durante l'elaborazione di apertura poiché l'accodamento all'interno del gruppo era DISABLED in fase di apertura, ma l'accodamento all'interno del gruppo è ENABLED in fase di inserimento, quindi il gestore code invalida l'handle dell'oggetto e non supera la richiesta di inserimento con MQRC_OBJECT_CHANGED.
- Se SYSTEM.QSG.TRANSMIT.QUEUE è stato incluso nel percorso di risoluzione del nome durante l'elaborazione di apertura poiché l'accodamento all'interno del gruppo era abilitato in fase di apertura, ma il SISTEMA SYSTEM.QSG.TRANSMIT.QUEUE è stata modificata in base all'ora di inserimento, quindi il gestore code invalida l'handle dell'oggetto e non riesce a eseguire la richiesta di inserimento con MQRC_OBJECT_CHANGED.

Ripristino automatico dell'agente di accodamento all'interno del gruppo

Se l'agent IGQ termina in modo anomalo, viene emesso il messaggio CSQM067E e l'agent IGQ viene avviato di nuovo.

Capacità di nuovo tentativo dell'agente di accodamento all'interno del gruppo

Se l'agent IGQ rileva un problema durante l'accesso a SYSTEM.QSG.TRANSMIT.QUEUE (poiché non è definito, ad esempio, o è definito con attributi non corretti, o è inibito per Gets, o per qualche altro motivo), l'agent IGQ passa allo stato di nuovo tentativo.

L'agent IGQ osserva intervalli e conteggi di tentativi brevi e lunghi. I valori per questi conteggi e intervalli, che non possono essere modificati, sono i seguenti:

Tabella 20. Valori di intervalli e conteggi tentativi brevi e lunghi

Costante	**Valore**
Numero di tentativi brevi	10
Intervallo nuovo tentativo breve	60 secondi = 1 min
Numero di tentativi lunghi	999,999,999
Intervallo nuovo tentativo lungo	1200 secondi = 20 min

L'agent di accodamento all'interno del gruppo e la serializzazione

Un tentativo dell'agent IGQ di serializzare l'accesso alle code condivise mentre il ripristino peer è ancora in corso potrebbe non riuscire.

Se si verifica un errore di un gestore code in un gruppo di condivisione code mentre l'agent IGQ sta gestendo i messaggi non sottoposti a commit su una o più code condivise, l'agent IGQ termina e il ripristino peer della coda condivisa viene eseguito per il gestore code in errore. Poiché il ripristino peer della coda condivisa è un'attività asincrona, lascia la possibilità per il gestore code in errore e anche per l'agent IGQ per tale gestore code, di riavviare prima del completamento del ripristino peer della coda condivisa. Ciò a sua volta lascia la possibilità che i messaggi di cui è stato eseguito il commit vengano elaborati prima e fuori sequenza, con i messaggi ancora in fase di recupero. Per garantire che i messaggi non vengano elaborati fuori sequenza, l'agent IGQ serializza l'accesso alle code condivise emettendo la chiamata API MQCONNX.

Un tentativo dell'agent IGQ di serializzare l'accesso alle code condivise mentre il ripristino peer è ancora in corso potrebbe non riuscire. Viene emesso un messaggio di errore e l'agent IGQ viene impostato sullo stato Riprova. Quando il ripristino peer del gestore code è completo, ad esempio al momento del successivo tentativo, l'agente IGQ può essere avviato.

z/OS Gestione della memoria su z/OS

IBM MQ for z/OS richiede strutture di dati permanenti e temporanee e utilizza serie di pagine e buffer di memoria per memorizzare questi dati. Questi argomenti forniscono ulteriori dettagli su come IBM MQ utilizza queste serie di pagine e buffer.

Concetti correlati

“Serie di pagine per IBM MQ for z/OS” a pagina 232
Utilizzare questo argomento per comprendere come IBM MQ for z/OS utilizza le serie di pagine per memorizzare i messaggi.

“Classi di memoria per IBM MQ for z/OS” a pagina 233
Una classe di memorizzazione è un concetto IBM MQ for z/OS che consente al gestore code di associare le code alle serie di pagine. È possibile utilizzare le classi di memoria per controllare quali dataset vengono utilizzati dalle code.

“Buffer e pool di buffer per IBM MQ for z/OS” a pagina 234

IBM MQ for z/OS utilizza i buffer e i pool di buffer per memorizzare temporaneamente nella cache i dati. Utilizzare questo argomento per comprendere ulteriormente come vengono organizzati e utilizzati i buffer.

Riferimenti correlati

“Dove trovare ulteriori informazioni sulla gestione dell'archiviazione per IBM MQ for z/OS” a pagina 236
Utilizzare questo argomento come riferimento per trovare ulteriori informazioni sulla gestione della memoria per IBM MQ for z/OS.

z/OS Serie di pagine per IBM MQ for z/OS

Utilizzare questo argomento per comprendere come IBM MQ for z/OS utilizza le serie di pagine per memorizzare i messaggi.

Una *serie di pagine* è un dataset lineare VSAM che è stato formattato appositamente per essere utilizzato da IBM MQ. Le serie di pagine vengono utilizzate per memorizzare la maggior parte dei messaggi e delle definizioni degli oggetti.

Le eccezioni sono le definizioni globali, memorizzate in un repository condiviso su Db2, e i messaggi sulle code condivise. Questi non vengono memorizzati nei set di pagine del gestore code. Per ulteriori informazioni sulle code condivise, consultare “Code condivise e gruppi di condivisione code” a pagina 171e per ulteriori informazioni sulle definizioni globali, consultare Definizioni private e globali.

Le serie di pagine IBM MQ possono avere una dimensione massima di 64 GB. Ogni serie di pagine è identificata da un PSID (page set identifier), un numero intero compreso tra 00 e 99. Ogni gestore code deve avere le proprie serie di pagine.

IBM MQ utilizza la serie di pagine zero (PSID=00) per memorizzare definizioni di oggetti e altre importanti informazioni relative al gestore code. Per il normale funzionamento di IBM MQ è essenziale che la serie di pagine zero non diventi piena, quindi non utilizzarla per memorizzare i messaggi.

Per migliorare le prestazioni del sistema, è necessario anche separare i messaggi di breve durata dai messaggi di lunga durata inserendoli in serie di pagine differenti.

È necessario formattare le serie di pagine e IBM MQ fornisce un programma di utilità FORMAT per questo; consultare Formattazione delle serie di pagine (FORMAT). Le serie di pagina devono essere definite anche nel sottosistema IBM MQ .

IBM MQ for z/OS può essere configurato per espandere dinamicamente una serie di pagine se diventa piena. IBM MQ continua ad espandere la serie di pagine se richiesto fino a quando non esistono 123 estensioni logiche, se è disponibile spazio di memoria su disco sufficiente. Le estensioni possono estendersi ai volumi se il dataset lineare è definito in questo modo, tuttavia, IBM MQ non può espandere le serie di pagine oltre 64 GB.

Non è possibile utilizzare i set di pagine da un gestore code IBM MQ su un gestore code IBM MQ differente o modificare il nome del gestore code. Se si desidera trasferire i dati da un gestore code a un altro, è necessario scaricare tutti gli oggetti e i messaggi dal primo gestore code e ricaricarli su un altro.

Non è possibile utilizzare serie di pagine superiori a 4 GB in un gestore code che esegue una release precedente a V6. Durante il periodo di migrazione, quando è probabile che sia necessario tornare a una release precedente del codice:

- Non modificare la serie di pagine 0 in modo che sia maggiore di 4 GB.
- Altre serie di pagine superiori a 4 GB verranno lasciate offline quando si riavvia un gestore code con una release precedente.

Per ulteriori informazioni sulla migrazione delle serie di pagine esistenti in grado di espandersi oltre i 4 GB, consultare Definizione di una serie di pagine con dimensioni superiori a 4 GB.

È possibile per un amministratore aggiungere dinamicamente le serie di pagine a un gestore code in esecuzione o rimuovere le serie di pagine da un gestore code in esecuzione (ad eccezione della serie di pagine zero). Il comando DEFINE PSID può essere eseguito dopo il completamento del riavvio del gestore code, solo se il comando contiene la parola chiave DSN.

z/OS Classi di memoria per IBM MQ for z/OS

Una classe di memorizzazione è un concetto IBM MQ for z/OS che consente al gestore code di associare le code alle serie di pagine. È possibile utilizzare le classi di memoria per controllare quali dataset vengono utilizzati dalle code.

Introduzione alle classi di memoria

Una *classe di memoria* associa una o più code a una serie di pagine. Ciò significa che i messaggi per tale coda vengono memorizzati nella serie di pagine.

Le classi di memoria consentono di controllare dove vengono memorizzati i dati di messaggi non condivisi per scopi di gestione, di gestione del carico e dello spazio del dataset o per l'isolamento dell'applicazione. È inoltre possibile utilizzare le classi di memoria per definire il gruppo XCF e il nome membro di una regione IMS se si utilizza il bridge IMS (descritto in "IBM MQ e IMS" a pagina 291).

Le code condivise non utilizzano classi di memoria per ottenere un'associazione di serie di pagine poiché i messaggi su di esse non sono memorizzati nelle serie di pagine.

Funzionamento delle classi di memoria

- Definire una classe di memoria, utilizzando il comando DEFINE STGCLASS, specificando un identificativo della serie di pagine (PSID).
- Quando si definisce una coda, si specifica la classe di memorizzazione nell'attributo STGCLASS.

Nel seguente esempio, la coda locale QE5 è associata alla serie di pagine 21 tramite la classe di memoria ARC2.

```
DEFINE STGCLASS(ARC2) PSID(21)
DEFINE QLOCAL(QE5) STGCLASS(ARC2)
```

Ciò significa che i messaggi inseriti nella coda QE5 vengono memorizzati nella serie di pagine 21 (se rimangono nella coda abbastanza a lungo da essere scritti in DASD).

Più di una coda può utilizzare la stessa classe di memoria ed è possibile definire tutte le classi di memoria desiderate. Ad esempio, è possibile estendere l'esempio precedente per includere più definizioni di classe di memoria e coda, come segue:

```
DEFINE STGCLASS(ARC1) PSID(05)
DEFINE STGCLASS(ARC2) PSID(21)
DEFINE STGCLASS(MAXI) PSID(05)
DEFINE QLOCAL(QE1) STGCLASS(ARC1) ...
DEFINE QLOCAL(QE2) STGCLASS(ARC1) ...
DEFINE QLOCAL(QE3) STGCLASS(MAXI) ...
DEFINE QLOCAL(QE4) STGCLASS(ARC2) ...
DEFINE QLOCAL(QE5) STGCLASS(ARC2) ...
```

In Figura 71 a pagina 234, le classi di memoria ARC1 e MAXI sono associate alla serie di pagine 05. Pertanto, le code QE1, QE2e QE3 vengono associate alla serie di pagine 05. Allo stesso modo, la classe di memoria ARC2 associa code QE4 e QE5 alla serie di pagine 21.

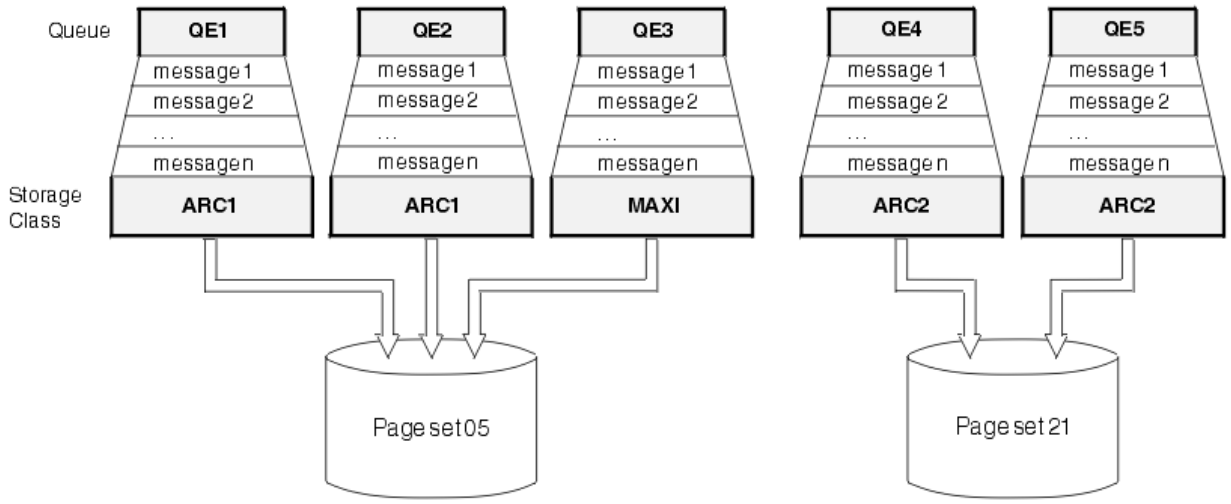


Figura 71. Associazione di code a serie di pagine tramite classi di memoria

Se si definisce una coda senza specificare una classe di memoria, IBM MQ utilizza una classe di memoria predefinita.

Se un messaggio viene inserito in una coda che denomina una classe di memoria inesistente, l'applicazione riceve un errore. È necessario modificare la definizione della coda per assegnarle un nome di classe di memoria esistente o creare la classe di memoria denominata dalla coda.

È possibile modificare una classe di memoria solo quando:

- Tutte le code che utilizzano questa classe di memoria sono vuote e non hanno attività di cui non è stato eseguito il commit.
- Tutte le code che utilizzano questa classe di memoria vengono chiuse.

z/OS Buffer e pool di buffer per IBM MQ for z/OS

IBM MQ for z/OS utilizza i buffer e i pool di buffer per memorizzare temporaneamente nella cache i dati. Utilizzare questo argomento per comprendere ulteriormente come vengono organizzati e utilizzati i buffer.

Per maggiore efficienza, IBM MQ utilizza una forma di memorizzazione nella cache in cui i messaggi (e le definizioni di oggetti) vengono memorizzati temporaneamente nei buffer prima di essere memorizzati nelle serie di pagine su DASD. I messaggi di breve durata, ossia i messaggi richiamati da una coda poco dopo la loro ricezione, potrebbero essere memorizzati solo nei buffer. Questa attività di memorizzazione nella cache è controllata da un gestore buffer, che è un componente di IBM MQ.

I buffer sono organizzati in *pool di buffer*. È possibile definire fino a 100 pool di buffer (da 0 a 99) per ciascun gestore code.

Si consiglia di utilizzare il numero minimo di pool di buffer congruenti con l'oggetto e la separazione del tipo di messaggio descritti in Figura 72 a pagina 235e con eventuali requisiti di isolamento dei dati dell'applicazione. Ogni buffer è lungo 4 KB. I pool di buffer utilizzano la memoria a 31 bit per impostazione predefinita, in questa modalità, il numero massimo di buffer è determinato dalla quantità di memoria a 31 bit disponibile nello spazio di indirizzo del gestore code; non utilizzare più del 70% circa per i buffer. In alternativa, l'assegnazione della memoria del pool di buffer può essere effettuata dalla memoria a 64 bit (utilizzare l'attributo LOCATION del comando **DEFINE BUFFPOOL**). L'utilizzo di LOCATION (SUPERIORE) in modo da utilizzare lo storage a 64 bit ha due vantaggi. In primo luogo, c'è molto più storage a 64 bit disponibile, quindi i pool di buffer possono essere molto più grandi, e in secondo luogo, lo storage a 31 bit è reso disponibile per l'utilizzo da parte di altre funzioni. Di solito, più buffer si hanno, più efficiente è il buffer e migliori sono le prestazioni di IBM MQ.

Figura 72 a pagina 235 mostra le relazioni tra messaggi, buffer, pool di buffer e serie di pagine. Un pool di buffer è associato a una o più serie di pagine; ogni serie di pagine è associata a un singolo pool di buffer.

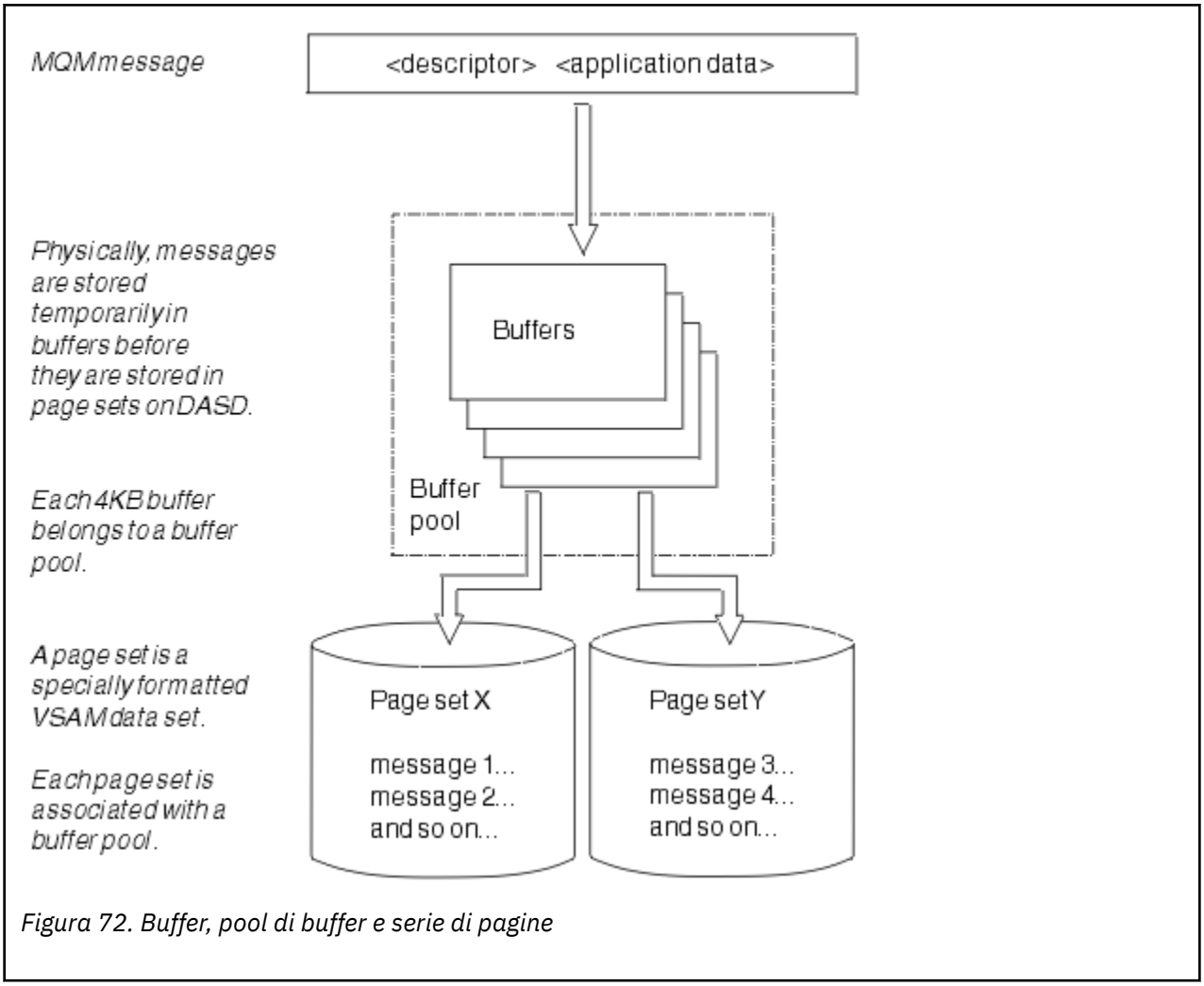


Figura 72. Buffer, pool di buffer e serie di pagine

È possibile immettere dinamicamente i comandi per modificare la dimensione e l'ubicazione del pool di buffer, utilizzando il comando **ALTER BUFFPOOL** . Le serie di pagine possono essere aggiunte in modo dinamico utilizzando il comando **DEFINE PSID** o eliminate utilizzando il comando **DELETE PSID** .

Se un pool di buffer è troppo piccolo, IBM MQ emette il messaggio CSQP020E. È quindi possibile aggiungere dinamicamente più buffer al pool di buffer interessato (notare che potrebbe essere necessario rimuovere i buffer da altri pool di buffer per eseguire questa operazione).

Si specifica il numero di buffer in un pool con il comando **DEFINE BUFFPOOL** ed è possibile ridimensionare dinamicamente i pool di buffer con il comando **ALTER BUFFPOOL** . Il numero corrente di buffer in un pool viene determinato dinamicamente visualizzando una serie di pagine che utilizza il pool di buffer, utilizzando il comando **DISPLAY USAGE** .

Per motivi di prestazioni, non inserire messaggi e definizioni oggetto nello stesso pool di buffer. Utilizzare un pool di buffer (ad esempio il numero zero) esclusivamente per la serie di pagine zero, dove vengono conservate le definizioni degli oggetti. Allo stesso modo, conservare messaggi di breve durata e messaggi di lunga durata in pool di buffer differenti e quindi in serie di pagine differenti e in code diverse.

Il comando **DEFINE BUFFPOOL** non può essere utilizzato dopo il riavvio per creare un nuovo pool di buffer. Invece, se un comando **DEFINE PSID** utilizza la parola chiave DSN, può identificare esplicitamente un bufferpool che non è attualmente definito. Il nuovo pool di buffer verrà creato.

z/OS Dove trovare ulteriori informazioni sulla gestione dell'archiviazione per IBM MQ for z/OS

Utilizzare questo argomento come riferimento per trovare ulteriori informazioni sulla gestione della memoria per IBM MQ for z/OS.

È possibile trovare ulteriori informazioni sugli argomenti in questa sezione dalle seguenti origini:

Tabella 21. Dove trovare ulteriori informazioni sulla gestione della memoria

Argomento	**Dove cercare**
Di quanta memoria hai bisogno	Pianificazione dei requisiti di archiviazione e prestazioni su z/OS
Quanto è grande rendere i set di pagine e bufferpool	Pianificare le serie di pagine e i pool di buffer
Gestione delle serie di pagine	Gestione delle serie di pagine
Comandi MQSC	Comandi MQSC

z/OS accessoIBM MQ for z/OS

IBM MQ conserva i *log* delle modifiche dei dati e degli eventi significativi quando si verificano. Questi log possono essere utilizzati per ripristinare i dati ad uno stato precedente, se necessario.

Il *dataset bootstrap* (BSDS) memorizza le informazioni relative ai dataset che contengono i log.

Il log non contiene informazioni per le statistiche, le tracce o la valutazione delle prestazioni. Per ulteriori dettagli sulle informazioni statistiche e di monitoraggio raccolte da IBM MQ , consultare Monitoraggio e statistiche.

Per ulteriori informazioni sulla registrazione, consultare i seguenti argomenti:

- “File di log in IBM MQ for z/OS” a pagina 236
- “Struttura del log” a pagina 240
- “Come vengono scritti i log IBM MQ for z/OS” a pagina 241
- “Indirizzo byte relativo log più grande” a pagina 244
- “Il dataset bootstrap” a pagina 245

Attività correlate

Pianificazione dell'ambiente di registrazione

Impostazione dei log utilizzando il modulo dei parametri di sistema

z/OS Amministrazionez/OS

Riferimenti correlati

z/OS Messaggi per IBM MQ for z/OS

z/OS File di log in IBM MQ for z/OS

I file di log contengono informazioni necessarie per il ripristino della transazione. I file di log attivi possono essere archiviati in modo da poter conservare i dati di log per un lungo periodo.

Cos' è un file di log

IBM MQ registra tutti gli eventi significativi come si verificano in un *log attivo*. Il file di log contiene le informazioni necessarie per il ripristino:

- Messaggi persistenti
- Oggetti IBM MQ , ad esempio code
- Il gestore code IBM MQ

Il log attivo comprende una raccolta di dataset (fino a 310) che vengono utilizzati ciclicamente.

È possibile abilitare l'archiviazione dei log in modo che quando un log attivo riempie una copia venga eseguita in un dataset di archiviazione. L'utilizzo dell'archiviazione consente di conservare i dati di log per un periodo prolungato. Se non si utilizza l'archiviazione, il ritorno a capo dei log e i dati precedenti vengono sovrascritti. Per ripristinare una serie di pagine o recuperare i dati in una struttura CF, sono necessari i dati di log da quando è stato eseguito il backup della serie di pagine o della struttura. Un log di archivio può essere creato su disco o su nastro.

Archivio

Poiché il log attivo ha una dimensione fissa, IBM MQ copia periodicamente il contenuto di ciascun dataset di log in un *log di archiviazione*, che di norma è un dataset su un DASD (direct - access storage device) o su un nastro magnetico. Se si verifica un malfunzionamento del sottosistema o della transazione, IBM MQ utilizza il log attivo e, se necessario, il log di archivio per il recupero.

Il log di archivio può contenere fino a 1000 dataset sequenziali. È possibile catalogare ogni dataset utilizzando l'ICF (integrated catalog facility) z/OS .

L'archiviazione è un elemento essenziale del ripristino IBM MQ . Se un'unità di ripristino è di lunga durata, i record di log all'interno di tale unità di ripristino potrebbero essere trovati nel log di archivio. In questo caso, il ripristino richiede i dati dal log di archivio. Tuttavia, se l'archiviazione è disabilitata, il log attivo con nuovi record di log viene riportato a capo, sovrascrivendo i record di log precedenti. Ciò significa che IBM MQ potrebbe non essere in grado di eseguire il backout dell'unità di ripristino e i messaggi potrebbero andare persi. Il gestore code termina in modo anomalo.

Pertanto, in un ambiente di produzione, **non disattivare mai l'archiviazione**. In tal caso, si corre il rischio di perdere dati dopo un errore di sistema o di transazione. Solo se si è in esecuzione in un ambiente di test è possibile disattivare l'archiviazione. Se è necessario, utilizzare la macro CSQ6LOGP , descritta in Utilizzo di CSQ6LOGP.

Per evitare problemi con unità di lavoro di lunga durata non pianificate, IBM MQ emette un messaggio (CSQJ160I o CSQJ161I) se viene rilevata un'unità di lavoro di lunga durata durante l'elaborazione dello scaricamento del log attivo.

registrazione doppia

Nella registrazione doppia, ogni record di log viene scritto in due diversi dataset di log attivi per ridurre al minimo la possibilità di problemi di perdita di dati durante il riavvio.

È possibile configurare IBM MQ per l'esecuzione con *registrazione singola* o *registrazione doppia*. Con la registrazione singola, i record di log vengono scritti una volta in un dataset di log attivo. Ogni dataset di log attivo è un LDS (linear data set) VSAM a singola estensione. Con la registrazione doppia, ogni record di log viene scritto in due diversi dataset di log attivi. La registrazione doppia riduce al minimo la probabilità di problemi di perdita di dati durante il riavvio.

Scollegamento log

Lo scollegamento del log fa sì che i record del log per alcune unità di lavoro vengano scritti più in basso nel log. Ciò riduce la quantità di dati di log che devono essere letti al riavvio del gestore code o al backout, per unità di lavoro di lunga esecuzione o a lungo termine in dubbio.

Quando un'unità di lavoro è considerata lunga, una rappresentazione di ogni record di log viene scritta più in basso nel log. Questa tecnica è nota come *shunting*. Quando l'intera unità di lavoro è stata elaborata, l'unità di lavoro si trova nello stato *deviato* . Qualsiasi attività di ripristino o di riavvio relativa all'unità di

lavoro deviata può utilizzare i record di log deviati invece di utilizzare l'unità originale dei record di log di lavoro.

Il rilevamento di un'unità di lavoro di lunga durata è una funzione del processo di checkpoint. Al momento del checkpoint, ogni unità di lavoro attiva viene controllata per stabilire se deve essere deviata. Se l'unità di lavoro ha superato due punti di controllo precedenti dalla sua creazione o dall'ultima volta che è stata deviata, l'unità di lavoro è adatta per essere deviata. Ciò significa che una singola unità di lavoro potrebbe essere deviata più di una volta. È nota come unità di lavoro *multi - shunted* .

Un'unità di lavoro viene deviata ogni tre checkpoint. Tuttavia, il punto di controllo viene eseguito in modo asincrono sul commutatore di log (o la scrittura del record di log che ha causato il superamento di LOGLOAD).

Esiste un solo punto di controllo alla volta, quindi potrebbero essere presenti più interruttori di log prima del completamento di un punto di controllo.

Ciò significa che se non ci sono abbastanza log attivi o se sono troppo piccoli, lo shunting di una grande unità di lavoro potrebbe non essere completato prima che tutti i log vengano riempiti.

Il messaggio CSQR027I risulta se non è possibile completare lo shunting.

Se l'archiviazione dei log è disattivata, l'ABEND 5C6 con motivo 00D1032A si verifica se si tenta di eseguire il backout dell'unità di lavoro per cui non è riuscita l'operazione di scollegamento. Per evitare questo problema è necessario utilizzare OFFLOAD=YES.

Lo shunting dei log è sempre attivo e viene eseguito indipendentemente dal fatto che l'archiviazione dei log sia abilitata o meno.

Nota: Sebbene tutti i record di log per un'unità di lavoro siano deviati, l'intero contenuto di ogni record non è deviato, ma solo la parte necessaria per il backout. Ciò significa che la quantità di dati di log scritti viene mantenuta al minimo e che i record deviati non possono essere utilizzati se si verifica un errore della serie di pagine. Un'unità di lavoro di lunga durata è un'unità in esecuzione da più di tre checkpoint del gestore code.

Per ulteriori informazioni sullo shunting dei log, vedi Gestione dei log.

Compressione log

È possibile configurare IBM MQ for z/OS per comprimere e decomprimere i record di log quando vengono scritti e letti dal dataset di log.

La compressione log può essere utilizzata per ridurre la quantità di dati scritti nel log per i messaggi persistenti sulle code private. La quantità di compressione raggiunta dipende dal tipo di dati contenuti nei messaggi. Ad esempio, RLE (Run Length Encoding) funziona compattando istanze ripetute di byte che possono fornire buoni risultati in modo efficiente per dati strutturati o orientati ai record.

Attenzione: I messaggi persistenti che vengono inseriti in una coda condivisa non sono soggetti alla compressione log.

È possibile utilizzare i campi all'interno della sezione Gestore log dei record SMF (System Management Facility 115) per monitorare la quantità di dati compressi. Per ulteriori informazioni su SMF, consultare Utilizzo di System Management Facility e Messaggi di statistiche e di account.

La compressione del log aumenta l'utilizzo del processore del sistema. Si consiglia di utilizzare la compressione solo se la velocità di trasmissione del gestore code è vincolata dalla scrittura della larghezza di banda IO nei dataset di log o se si è vincolati dalla memoria disco necessaria per conservare i dataset di log. Se si utilizzano code condivise, i vincoli della larghezza di banda IO possono essere alleviati aggiungendo ulteriori gestori code al gruppo di condivisione code e distribuendo il carico di lavoro su più gestori code.

L'opzione di compressione del log può essere abilitata e disabilitata come richiesto senza la necessità di arrestare e riavviare il gestore code. Il gestore code può leggere qualsiasi record di log compresso indipendentemente dall'impostazione di compressione del log corrente.

Il gestore code supporta 3 impostazioni per la compressione dei log.

Nessuna
Non viene utilizzata alcuna compressione dati di log. Questo è il valore predefinito.

RLE
La compressione dei dati di log viene eseguita utilizzando RLE (run - length encoding).

ANY
Abilitare il gestore code per selezionare l'algoritmo di compressione che fornisce il maggior grado di compressione dei record di log. Questa opzione risulta nella compressione RLE.

È possibile controllare la compressione dei record di log utilizzando uno dei seguenti:

- I comandi SET e DISPLAY LOG in MQSC; consultare SET LOG e DISPLAY LOG
- Le funzioni Imposta log e Richiedi log nell'interfaccia PCF; consultare Imposta log e Richiedi log
- La macro CSQ6LOGP nel modulo del parametro di sistema; consultare Utilizzo di CSQ6LOGP

Inoltre, il programma di utilità Log Print CSQ1LOGP supporta l'espansione dei record di log compressi.

Dati di log

Il log può contenere fino a 18 milioni di milioni ($1.8* 10^{19}$) byte. Ogni byte può essere indirizzato dal relativo offset dall'inizio del log e tale offset è noto come *RBA (relative byte address*).

L'RBA è indicato da un campo a 6 byte o a 8 byte che fornisce un intervallo indirizzabile totale di 2^{48} byte o 2^{64} byte, a seconda se sono in uso RBA di log a 6 byte o a 8 byte.

Tuttavia, quando IBM MQ rileva che l'intervallo utilizzato è superiore a F00000000000 (se sono in uso RBA a 6 byte) o FFFF800000000000 (se sono in uso RBA di log a 8 byte), vengono emessi i messaggi CSQI045, CSQI046, CSQI047e CSQJ032 , con l'avvertenza di reimpostare l'RBA di log.

Se il valore RBA raggiunge FFF800000000 (se sono in uso RBA di log a 6 byte) o FFFFFFC000000000 (se sono in uso RBA di log a 8 byte), il gestore code termina con il codice motivo 00D10257.

Una volta emessi i messaggi di avvertenza relativi all'intervallo di log utilizzato, è necessario pianificare un'interruzione del gestore code durante la quale il gestore code può essere convertito per utilizzare RBA di log a 8 byte oppure è possibile reimpostare il log. La procedura per reimpostare il log è documentata in Reimpostazione del log del gestore code.

Se il gestore code sta utilizzando RBA di log a 6 byte, considerare la possibilità di convertire il gestore code per utilizzare RBA di log a 8 byte invece di reimpostare il log del gestore code, seguendo la procedura documentata in Implementazione dell'indirizzo RBA (Relative Byte Address) di log più grande.

Il log è composto da *record di log*, ognuno dei quali è una serie di dati di log trattati come una singola unità. Un record di log è identificato dall'RBA del primo byte della sua intestazione o dal suo LRSN (log record sequence number). RBA o LRSN identificano in modo univoco un record che inizia in un particolare punto del log.

Se si utilizza l'RBA o l'LRSN per identificare i punti di log dipende dal fatto che si stiano utilizzando i gruppi di condivisione code. In un ambiente di condivisione della coda, non è possibile utilizzare l'indirizzo byte relativo per identificare in modo univoco un punto di log, poiché più gestori code possono aggiornare la stessa coda contemporaneamente e ciascuno ha il proprio log. Per risolvere questo problema, il numero di sequenza del record di log è derivato da un valore data / ora e non rappresenta necessariamente lo spostamento fisico del record di log all'interno del log.

Ogni record di log ha un'intestazione che fornisce il suo tipo, il componente secondario IBM MQ che ha creato il record e, per i record di unità di ripristino, un identificativo di unità di recupero.

Esistono quattro tipi di record di log, descritti nelle seguenti intestazioni:

- Unità di record log di ripristino
- Record checkpoint
- Record di controllo della serie di pagine

- Record di backup struttura CF

Unità dei record di registrazione per il recupero

La maggior parte dei record di log descrivono le modifiche alle code IBM MQ . Tutte queste modifiche vengono effettuate all'interno delle unità di recupero.

IBM MQ utilizza tecniche di registrazione speciali che coinvolgono *undo / redo* e *compensano i record di log* per ridurre i tempi di riavvio e migliorare la disponibilità del sistema.

Un effetto di ciò è che il tempo di riavvio è limitato. Se si verifica un errore durante un riavvio, in modo che il gestore code debba essere riavviato una seconda volta, non è necessario riapplicare tutte le attività di ripristino completate fino al punto di errore nel primo riavvio durante un secondo riavvio. Ciò significa che i riavvii successivi non richiedono tempi progressivamente più lunghi per essere completati.

Record checkpoint

Per ridurre il tempo di riavvio, IBM MQ utilizza punti di controllo periodici durante il normale funzionamento. Questi si verificano come segue:

- Quando è stato scritto un numero predefinito di record di log. Questo numero viene definito dall'operando di frequenza del punto di controllo denominato LOGLOAD della macro del parametro di sistema CSQ6SYSP, descritto in Utilizzo di CSQ6SYSP.
- Al termine di un riavvio riuscito.
- Al termine normale.
- Ogni volta che IBM MQ passa al successivo dataset di log attivo nel ciclo.

Quando viene eseguito un punto di controllo, IBM MQ emette il comando DISPLAY CONN (descritto in DISPLAY CONN) internamente in modo che un elenco di connessioni attualmente in dubbio venga scritto nel log della console z/OS .

Record di controllo serie di pagine

Questi record registrano le serie di pagine e i pool di buffer noti al gestore code IBM MQ ad ogni punto di controllo e registrano le informazioni sugli intervalli di log richiesti per eseguire il ripristino del supporto della serie di pagine al momento del punto di controllo.

Alcune modifiche dinamiche alle serie di pagine e ai pool di buffer vengono scritte anche come record di controllo delle serie di pagine, in modo che le modifiche possano essere recuperate e reinserite automaticamente al successivo riavvio del gestore code.

Record di backup della struttura CF

Questi record contengono i dati letti da una struttura di elenco CF (Coupling Facility) in risposta a un comando BACKUP CFSTRUCT. Nel caso improbabile di un malfunzionamento della struttura CFS, questi record vengono utilizzati, insieme ai record unità di recupero, dal comando RECOVER CFSTRUCT per eseguire il recupero del supporto della struttura CFS fino al punto di errore.

Attività correlate

Implementazione dell'indirizzo di byte relativo del log più grande

z/OS Struttura del log

Utilizzare questo argomento per comprendere la terminologia utilizzata per descrivere i record di log.

Ogni dataset di log attivo deve essere un LDS (linear data set) VSAM. L'unità di output fisica scritta nel dataset del log attivo è un CI (control interval) di 4 KB. Ogni IC contiene un record VSAM.

Record di log fisici e logici

Un IC VSAM è un record *fisico* . Le informazioni registrate in un determinato momento formano un record *logico* , con una lunghezza che varia indipendentemente dallo spazio disponibile nell'IC. Quindi, un record fisico potrebbe contenere:

- Diversi record logici
- Uno o più record logici e parte di un altro record logico
- Parte di un solo record logico

Il termine *record di log* fa riferimento al record *logico* , indipendentemente dal numero di record *fisici* necessari per memorizzarlo.

z/OS Come vengono scritti i log IBM MQ for z/OS

Utilizzare questo argomento per comprendere come IBM MQ elabora i record del file di log.

IBM MQ scrive ciascun record di log in un dataset DASD denominato *log attivo*. Quando il log attivo è pieno, IBM MQ copia il suo contenuto in un DASD o in un dataset nastro denominato *log di archiviazione*. Questo processo è denominato *offload*.

Figura 73 a pagina 242 illustra il processo di registrazione. I record di log generalmente passano attraverso il seguente ciclo:

1. IBM MQ nota le modifiche ai dati e gli eventi significativi nei record del log di ripristino.
2. IBM MQ elabora i record del log di ripristino e li suddivide in segmenti, se necessario.
3. I record di log vengono posizionati in modo sequenziale nei *buffer di log di output*, formattati come CI (Controls Interval) VSAM. Ogni record di log è identificato da un indirizzo di byte relativo compreso tra zero e 2^{64} -1.
4. Gli IC vengono scritti in una serie di dataset di log attivi DASD predefiniti, che vengono utilizzati in modo sequenziale e riciclati.
5. Se l'archiviazione è attiva, quando ogni dataset di log attivo si riempie, il relativo contenuto viene automaticamente scaricato in un nuovo dataset di log di archivio.

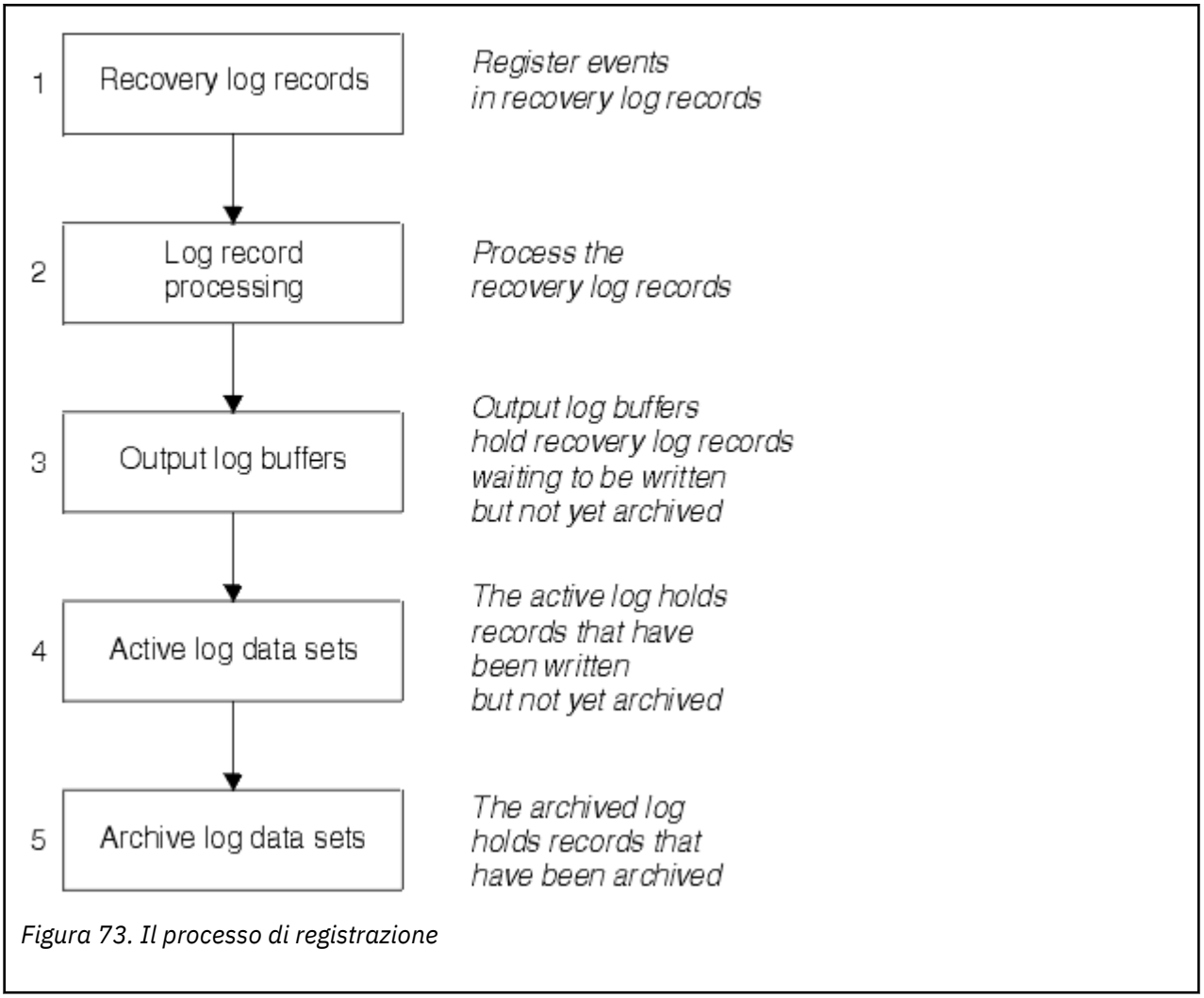


Figura 73. Il processo di registrazione

Quando viene scritto il log attivo

I buffer di log in memoria vengono scritti in un dataset di log attivo ogni volta che si verifica una delle seguenti condizioni:

- I buffer di log diventano pieni.
- La soglia di scrittura viene raggiunta (come specificato nella macro CSQ6LOGP).
- Si verificano alcuni eventi significativi, come un punto di commit o quando viene emesso un comando IBM MQ BACKUP CFSTRUCT.

Quando il gestore code viene inizializzato, i data set di log attivi denominati in BSDS vengono assegnati in modo dinamico per l'uso esclusivo da parte del gestore code e rimangono assegnati esclusivamente a IBM MQ fino a quando il gestore code non termina.

Aggiunta dinamica di dataset di log

È possibile definire dinamicamente nuovi dataset di log attivi mentre il gestore code è in esecuzione. Questa funzione riduce il problema di un blocco del gestore code quando l'archiviazione non è in grado di scaricare i log attivi a causa di un problema transitorio. Per ulteriori informazioni, consultare il comando DEFINE LOG .

Nota: Per ridefinire o rimuovere i log attivi, è necessario terminare e riavviare il gestore code.

IBM MQ e Storage Management Subsystem

I parametri IBM MQ consentono di specificare le classi di archiviazione MVS/DFP SMS (Storage Management Subsystem) durante l'allocazione dinamica dei dataset di log di archivio IBM MQ . IBM MQ avvia l'archiviazione dei dataset di log, ma è possibile utilizzare SMS per eseguire l'allocazione del dataset di archivio.

Riferimenti correlati

"Quando viene scritto il log di archivio IBM MQ for z/OS" a pagina 243
Utilizzare questo argomento per comprendere il processo di copia dei log attivi nei log di archiviazione e quando si verifica il processo.

z/OS *Quando viene scritto il log di archivio IBM MQ for z/OS*

Utilizzare questo argomento per comprendere il processo di copia dei log attivi nei log di archiviazione e quando si verifica il processo.

Il processo di copia dei log attivi nei log di archiviazione è denominato *offload*. La relazione di offload ad altri eventi di registrazione viene mostrata schematicamente in Figura 74 a pagina 243.

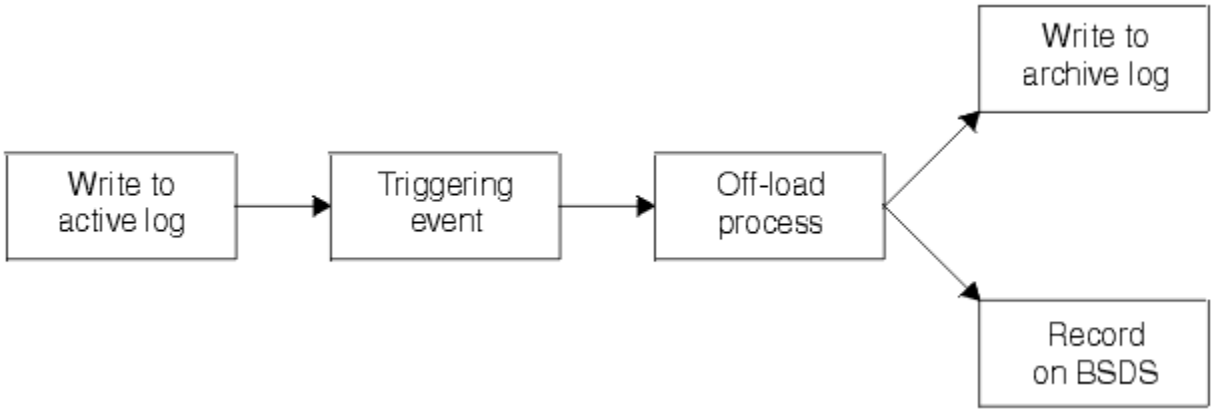


Figura 74. Il processo di scaricamento

Attivazione del processo di scaricamento

Il processo di offload di un log attivo in un log di archiviazione può essere attivato da diversi eventi. Ad esempio:

- Riempimento di un dataset di log attivo.
- Utilizzo del comando MQSC ARCHIVE LOG.
- Si è verificato un errore durante la scrittura in un dataset di log attivo.

Il dataset viene troncato prima del punto di errore e il record che non è stato scritto diventa il primo record del nuovo dataset. L'offload viene attivato per il dataset troncato come per un dataset di log completo normale. Se sono presenti due log attivi, entrambe le copie vengono troncate in modo che le due copie rimangano sincronizzate.

Il messaggio CSQJ110E viene emesso quando l'ultimo log attivo disponibile è pieno al 5% e successivamente aumenta del 5%, indicando la percentuale della capacità del log in uso. Se tutti i log attivi diventano pieni, IBM MQ arresta l'elaborazione, fino a quando non si verifica lo scaricamento ed emette questo messaggio:

```
CSQJ111A +CSQ1 OUT OF SPACE IN ACTIVE LOG DATA SETS
```

Il processo di offload

Quando tutti i log attivi diventano pieni, IBM MQ esegue il processo di scaricamento e arresta l'elaborazione fino a quando il processo di scaricamento non è stato completato. Se l'elaborazione dell'offload ha esito negativo quando i log attivi sono pieni, IBM MQ si interrompe.

Quando un log attivo è pronto per essere scaricato, viene inviata una richiesta all'operatore della console z/OS per montare un nastro o preparare un'unità DASD. Il valore dell'opzione di registrazione ARCWTOR (per ulteriori informazioni, consultare Utilizzo di CSQ6ARVP) determina se la richiesta viene ricevuta. Se si sta utilizzando un nastro per lo scaricamento, specificare ARCWTOR=YES. Se il valore è YES, la richiesta è preceduta da un WTOR (numero messaggio CSQJ008E) indicare all'operatore di preparare un dataset di log di archivio da assegnare.

L'operatore non deve rispondere immediatamente a questo messaggio. Tuttavia, ritardare la risposta ritarda il processo di offload. Non influisce sulle prestazioni di IBM MQ a meno che l'operatore non ritardi la risposta così a lungo che IBM MQ esaurisce i log attivi.

L'operatore può rispondere annullando il processo di scaricamento. In questo caso, se l'assegnazione è per la prima copia dei dataset di archiviazione duali, il processo di scaricamento viene semplicemente ritardato fino a quando il successivo dataset di log attivo diventa pieno. Se l'assegnazione è per la seconda copia, il processo di archiviazione passa alla modalità di copia singola, ma solo per questo dataset.

Interruzioni ed errori durante lo scaricamento

Una richiesta di arresto del gestore code non ha effetto fino al termine dell'elaborazione dell'offload. Se IBM MQ non riesce mentre l'offload è in corso, l'offload inizia di nuovo quando il gestore code viene riavviato.

Messaggi durante l'elaborazione dell'offload

I messaggi scaricati vengono inviati alla console z/OS da IBM MQ e dal processo di scaricamento. È possibile utilizzare questi messaggi per trovare gli intervalli RBA nei vari dataset di log.

z/OS Indirizzo byte relativo log più grande

Questa funzione migliora la disponibilità del gestore code aumentando il periodo di tempo prima che sia necessario reimpostare il log.

I dati di ripristino vengono scritti nel log in modo che i messaggi persistenti siano disponibili quando il gestore code viene riavviato. Il termine RBA (Relative Byte Address) di log viene utilizzato per fare riferimento all'ubicazione dei dati come offset dall'inizio del log.

Prima di IBM MQ 8.0, l'RBA di log a 6 byte poteva indirizzare fino a 256 terabyte di dati. Prima di scrivere questa quantità di record di log, è necessario reimpostare il log del gestore code seguendo la procedura documentata in Reimpostazione del log del gestore code.

La reimpostazione dei log dei gestori code non è un processo rapido e può richiedere un'interruzione estesa, a causa della necessità di reimpostare le serie di pagine come parte del processo. Per un gestore code di utilizzo elevato, questa operazione può essere eseguita una volta all'anno.

Da IBM MQ 8.0, l'RBA di log può essere lungo 8 byte e il gestore code può ora indirizzare oltre 64.000 volte la quantità di dati (16 exabyte) prima che l'RBA di log debba essere reimpostato. L'impatto dell'utilizzo dell'RBA di log più grande è che la dimensione dei dati di log scritti aumenta di alcuni byte.

Quando è abilitata questa funzione?

V 9.3.0 Questa funzione è già abilitata per i gestori code creati in IBM MQ 9.3.0 o versioni successive.

Se l'RBA di log corrente si sta avvicinando alla fine dell'intervallo di RBA di log, considerare la conversione del gestore code per utilizzare un RBA di log a 8 byte invece di reimpostare il log del gestore code. La

conversione di un gestore code per l'utilizzo di RBA di log a 8 byte richiede un'interruzione più breve rispetto alla reimpostazione del log e aumenta significativamente il periodo di tempo prima di dover reimpostare il log.

Il messaggio CSQJ034I, emesso durante l'inizializzazione del gestore code, indica la fine dell'intervallo RBA di log per il gestore code come configurato e può essere utilizzato per stabilire se sono in uso RBA di log a 6 o 8 byte.

Come è abilitata questa funzione?

RBA di log a 8 byte è abilitato avviando il gestore code con un formato BSDS versione 2. In sintesi, ciò si ottiene:

1. Verifica che tutti i gestori code nel gruppo di condivisione code soddisfino i requisiti per l'abilitazione dell'RBA di log a 8 byte
2. Chiusura del gestore code in modo pulito
3. Esecuzione del programma di utilità di conversione BSDS per creare una copia di BSDS in formato versione 2.
4. Riavvio del gestore code con il BSDS convertito.

Una volta che un gestore code è stato convertito per utilizzare RBA di log a 8 byte, non può tornare a utilizzare RBA di log a 6 byte.

Consultare Implementazione dell'RBA (Relative Byte Address) di log per la procedura dettagliata su come abilitare gli RBA di log di 8 byte.

Attività correlate

Pianificazione per aumentare l'intervallo di log indirizzabile massimo

Riferimenti correlati

Programma di utilità di conversione BSDS (CSQJUCNV)

z/OS Il dataset bootstrap

Il dataset di avvio è richiesto da IBM MQ come meccanismo per fare riferimento ai dataset di log e ai record di log. Queste informazioni sono richieste durante la normale elaborazione e riavviare il ripristino.

Scopo del dataset di bootstrap

BSDS (*bootstrap data set*) è un KSDS (key - sequenced data set) VSAM che contiene le informazioni richieste da IBM MQ. Contiene quanto segue:

- Un inventario di tutti i dataset di log attivi e archiviati noti a IBM MQ. IBM MQ utilizza questo inventario per:
  - Traccia dei dataset di log attivi e archiviati
  - Individuare i record di log in modo che possano soddisfare le richieste di lettura del log durante la normale elaborazione
  - Individuare i record di log in modo che possano gestire l'elaborazione del riavvio

  IBM MQ memorizza le informazioni nell'inventario ogni volta che viene definito un dataset di log di archivio o viene riutilizzato un dataset di log attivo. Per i log attivi, l'inventario mostra quali sono pieni e quali disponibili per il riutilizzo. L'inventario contiene l'RBA (relative byte address) di ogni parte del log contenuta in tale dataset.
- Un inventario *a capo* di tutte le attività IBM MQ recenti. Ciò è necessario se è necessario riavviare il gestore code.

Il BSDS è richiesto se il gestore code ha un errore ed è necessario riavviarlo. IBM MQ **deve** avere un BSDS. Per ridurre al minimo la probabilità di problemi durante un riavvio, è possibile configurare IBM MQ con BSDS doppi, ciascuno dei quali registra le stesse informazioni. L'utilizzo di BSDS doppi è noto come esecuzione in *modalità duale*. Se possibile, posizionare le copie su volumi separati. Ciò riduce il rischio

che entrambi vadano persi se il volume è danneggiato o distrutto. Utilizzare BSDS doppi piuttosto che la scrittura duale su DASD.

Il BSDS viene impostato quando IBM MQ è personalizzato ed è possibile gestire l'inventario utilizzando il programma di utilità inventario del log di modifica (CSQJU003). Per ulteriori informazioni su questo programma di utilità, consultare Amministrazione IBM MQ for z/OS. Viene indicato da un'istruzione DD nella procedura di avvio del gestore code.

Di solito, IBM MQ conserva copie duplicate di BSDS. Se si verifica un errore I/O, annulla l'allocazione della copia in errore e continua con un singolo BSDS. È possibile ripristinare l'operazione in modalità duale, come descritto in Amministrazione IBM MQ for z/OS.

I log attivi vengono prima registrati in BSDS quando viene installato IBM MQ . Non è possibile sostituire i log attivi senza terminare e riavviare il gestore code.

I dataset di log di archivio sono allocati in modo dinamico. Quando ne viene assegnato uno, il nome del dataset viene registrato in BSDS. L'elenco dei dataset di log di archiviazione si espande quando vengono aggiunti gli archivi e si riavvolge quando viene raggiunto un numero di voci determinato dall'utente. Il numero massimo di voci è 1000 per la registrazione di archivi singoli e 2000 per la registrazione doppia.

È possibile utilizzare un sistema di gestione nastri per eliminare i dataset di log di archiviazione (IBM MQ non dispone di un metodo automatico). Pertanto, le informazioni su un dataset del log di archivio possono trovarsi in BSDS molto tempo dopo che il dataset del log di archivio è stato eliminato dall'amministratore di sistema.

Al contrario, è possibile che sia stato superato il numero massimo di dataset di log di archivio e che i dati provenienti da BSDS siano stati eliminati molto prima che il dataset abbia raggiunto la data di scadenza.

È possibile utilizzare il seguente comando MQSC per determinare l'estensione del log e il nome del dataset del log attivo o di archivio che contiene il primo RBA di log, richiesto per vari tipi di ripristino del gestore code o del supporto:

```
DISPLAY USAGE TYPE(DATASET)
```

Se il modulo dei parametri di sistema specifica che i dataset di log di archiviazione vengono catalogati quando vengono assegnati, BSDS punta al catalogo ICF (integrated catalog facility) per le informazioni necessarie per le assegnazioni successive. Altrimenti, le voci BSDS per ogni volume registrano il numero di serie del volume e le informazioni sull'unità necessarie per le assegnazioni successive.

La versione BSDS

Il formato del BSDS varia in base alla versione. L'aumento della versione di BSDS consente l'utilizzo di nuove funzioni. Le seguenti versioni BSDS sono supportate da IBM MQ:

Versione 1
Supportato da tutte le release di IBM MQ. Un BSDS versione 1 supporta valori RBA di log a 6 byte.

Versione 2
Supportato da IBM MQ 8.0 e versioni successive. Un BSDS versione 2 abilita valori RBA di log a 8 byte e fino a 310 dataset in ogni copia di log attiva.

V 9.3.0 Abilitato per impostazione predefinita per i gestori code creati in IBM MQ 9.3.0 e versioni successive.

Versione 3
Supportato da IBM MQ 8.0 e versioni successive. BSDS viene automaticamente convertito nella versione 3, dalla versione 2, quando più di 31 dataset vengono aggiunti alla copia di log attiva.

È possibile determinare la versione di un BSDS eseguendo il programma di utilità di stampa della mappa del log (CSQJU004). Per convertire BSDS dalla Versione 1 alla Versione 2, eseguire il programma di utilità di conversione BSDS (CSQJUCNV).

Consultare “Indirizzo byte relativo log più grande” a pagina 244 per ulteriori informazioni sugli RBA di log a 6 byte e a 8 byte.

Dataset di log di archivio e copie BSDS

Ogni volta che viene creato un nuovo dataset di log di archiviazione, viene creata anche una copia di BSDS. Se il log di archivio è su nastro, BSDS è il primo dataset sul primo volume di output. Se il log di archivio è su DASD, BSDS è un dataset separato.

I nomi dataset del log di archivio e della copia BSDS sono gli stessi, ad eccezione del fatto che il qualificatore di livello più basso del nome del log di archivio inizia con A e la copia BSDS inizia con B, ad esempio:

Nome log di archivio
CSQ.ARCHLOG1.E00186.T2336229.*A*0000001

Nome copia BSDS
CSQ.ARCHLOG1.E00186.T2336229.*B*0000001

Se si verifica un errore di lettura durante la copia di BSDS, la copia non viene creata, viene emesso il messaggio CSQJ125E e l'operazione di scaricamento nel nuovo dataset del log di archivio continua senza la copia BSDS.

z/OS Definizione di sistema su z/OS

IBM MQ for z/OS utilizza molte definizioni di oggetti predefinite e fornisce un JCL di esempio per creare tali oggetti predefiniti. Utilizzare questo argomento per comprendere questi oggetti predefiniti e il JCL di esempio.

Impostazione dei parametri di sistema

In IBM MQ for z/OS, un modulo di parametri di sistema controlla gli ambienti di registrazione, archiviazione, traccia e connessione che IBM MQ utilizza nella sua operazione. I parametri di sistema sono specificati da tre macro assembler, come segue:

CSQ6SYSP
Parametri di sistema, inclusa l'impostazione dell'ambiente di connessione e traccia.

CSQ6LOGP
Parametri di registrazione.

CSQ6ARVP
Parametri di archiviazione log.

I moduli dei parametri predefiniti vengono forniti con IBM MQ for z/OS. Se questi non contengono i valori che si desidera utilizzare, è possibile creare i propri moduli parametro utilizzando l'esempio fornito con IBM MQ. L'esempio è `thlqual.SCSQPROC(CSQ4ZPRM)`.

È possibile modificare alcuni parametri di sistema mentre un gestore code è in esecuzione. Consultare i comandi SET SYSTEM, SET LOG e SET ARCHIVE in Comandi MQSC.

Per ulteriori informazioni sulla definizione, consultare i seguenti argomenti:

- “Definizione degli oggetti di sistema per IBM MQ for z/OS” a pagina 248
- “Ottimizzazione del gestore code su IBM MQ for z/OS” a pagina 252
- “Definizioni di esempio fornite con IBM MQ for z/OS” a pagina 253

Concetti correlati
Personalizza i dataset di input di inizializzazione di esempio
Origini da cui è possibile emettere comandi MQSC e PCF su IBM MQ for z/OS
Attività correlate
Amministrazionez/OS
Configurazione dei cluster
MonitoraggioIBM MQ

z/OS Definizione degli oggetti di sistema per IBM MQ for z/OS

IBM MQ for z/OS richiede ulteriori oggetti predefiniti per le applicazioni di pubblicazione / sottoscrizione, per il controllo del cluster e del canale e per altre funzioni di amministrazione del sistema.

Gli oggetti di sistema richiesti da IBM MQ for z/OS possono essere suddivisi nelle seguenti categorie:

- Oggetti di pubblicazione / sottoscrizione
- Oggetti predefiniti di sistema
- Oggetti comando di sistema
- Oggetti di amministrazione del sistema
- Code canali
- code cluster
- Code gruppi di condivisione code
- Classi di memoria
- Definizione della coda di messaggi non recapitabili dell'oggetto di sistema
- Coda di trasmissione predefinita
- Code interne
- “Coda di autenticazione canale” a pagina 252

Oggetti di pubblicazione / sottoscrizione

Ci sono diversi oggetti di sistema che è necessario definire prima di poter utilizzare le applicazioni di pubblicazione / sottoscrizione con IBM MQ for z/OS. Le definizioni di esempio vengono fornite con IBM MQ per facilitare la definizione di questi oggetti. Questi esempi sono descritti in CSQ4INSG.

Per utilizzare la pubblicazione / sottoscrizione è necessario definire i seguenti oggetti:

- Una coda locale denominata SYSTEM.RETAINED.PUB.QUEUE, utilizzato per conservare una copia di ciascuna pubblicazione conservata nel gestore code. Ogni nome argomento completo potrebbe avere fino a una pubblicazione conservata memorizzata su questa coda. Se le applicazioni utilizzano le pubblicazioni conservate su molti argomenti differenti o se i messaggi di pubblicazione conservati sono messaggi di grandi dimensioni, i requisiti di memoria per questa coda devono essere attentamente pianificati, inclusa l'assegnazione alla relativa serie di pagine se i requisiti di memoria sono elevati. Per migliorare le prestazioni, è necessario definire questa coda con un tipo di indice MSGID (come mostrato nella definizione della coda di esempio fornita).
- Una coda locale denominata SYSTEM.DURABLE.SUBSCRIBER.QUEUE, utilizzato per conservare una copia persistente delle sottoscrizioni durevoli nel gestore code. Per migliorare le prestazioni, è necessario definire questa coda con un tipo di indice CORRELID (come mostrato nella definizione della coda di esempio fornita).
- Una coda locale denominata SYSTEM.DURABLE.MODEL.QUEUE, utilizzato come modello per le sottoscrizioni durevoli gestite.
- Una coda locale denominata SYSTEM.NDURABLE.MODEL.QUEUE, utilizzato come modello per le sottoscrizioni non durevoli gestite.
- Un elenco nomi denominato SYSTEM.QPUBSUB.QUEUE.NAMELIST, che contiene un elenco di nomi di coda monitorati dall'interfaccia di pubblicazione / sottoscrizione in coda.
- Un elenco nomi denominato SYSTEM.QPUBSUB.SUBPOINT.NAMELIST, che contiene un elenco di oggetti argomento utilizzati dall'interfaccia di pubblicazione / sottoscrizione in coda per far corrispondere gli oggetti argomento ai punti di sottoscrizione.
- Un argomento denominato SYSTEM.BASE.TOPIC, utilizzato come argomento di base per la risoluzione degli attributi.

- Un argomento denominato SYSTEM.BROKER.DEFAULT.STREAM, che è lo stream predefinito utilizzato dall'interfaccia di pubblicazione / sottoscrizione in coda.
- Un argomento denominato SYSTEM.BROKER.DEFAULT.SUBPOINT, che è il punto di sottoscrizione RFH2 predefinito utilizzato dall'interfaccia di pubblicazione / sottoscrizione accodata.
- Un argomento denominato SYSTEM.BROKER.ADMIN.STREAM, che è il flusso admin utilizzato dall'interfaccia di pubblicazione / sottoscrizione accodata.
- Una sottoscrizione denominata SYSTEM.DEFAULT.SUB, che è un oggetto sottoscrizione predefinito utilizzato per fornire i valori predefiniti sui comandi DEFINE SUB.

Oggetti predefiniti del sistema

Gli oggetti predefiniti del sistema vengono utilizzati per fornire attributi predefiniti quando si definisce un oggetto e non si specifica il nome di un altro oggetto su cui basare la definizione.

I nomi delle definizioni degli oggetti di sistema predefiniti iniziano con i caratteri "SYSTEM.DEFAULT"o"SYSTEM.DEF. " Ad esempio, la coda locale predefinita del sistema è denominata SYSTEM.DEFAULT.LOCAL.QUEUE.

Questi oggetti definiscono i valori predefiniti di sistema per gli attributi di questi oggetti IBM MQ :

- Code locali
- Code modello
- Code alias
- Code remote
- Processi
- Elenchi nomi
- Canali
- Classi di memoria
- Informazioni di autenticazione

Le code condivise sono un tipo speciale di coda locale, quindi quando si definisce una coda condivisa, la definizione si basa su SYSTEM.DEFAULT.LOCAL.QUEUE. È necessario ricordare di fornire un valore per il nome della struttura CF (Coupling Facility) poiché non ne è stato specificato uno nella definizione predefinita. In alternativa, è possibile definire la propria definizione di coda condivisa predefinita da utilizzare come base per le code condivise in modo che ereditino tutti gli attributi richiesti. È necessario definire una coda condivisa su un gestore code solo nel gruppo di condivisione code.

Oggetti comando di sistema

I nomi degli oggetti comando di sistema iniziano con i caratteri SYSTEM.COMMAND. È necessario definire questi oggetti prima di poter utilizzare le operazioni e i pannelli di controllo di IBM MQ per immettere comandi in un sottosistema IBM MQ .

Esistono due oggetti comando di sistema:

1. La coda di immissione comandi di sistema è una coda locale in cui i comandi vengono inseriti prima di essere elaborati dal processore comandi IBM MQ . Deve essere denominato SYSTEM.COMMAND.INPUT. Un alias denominato SYSTEM.ADMIN.COMMAND.QUEUE per la compatibilità con IBM MQ for Multiplatformse per l'utilizzo da parte di IBM MQ Console e administrative REST API.
2. SYSTEM.COMMAND.REPLY.MODEL è una coda modello che definisce la coda di risposta del comando di sistema.

Esistono due oggetti aggiuntivi che possono essere utilizzati da IBM MQ Explorer:

- SYSTEM.MQEXPLORER.REPLY.MODEL coda

- SYSTEM.ADMIN.SVRCONN

SYSTEM.REST.REPLY.QUEUE è la coda di risposta utilizzata da IBM MQ administrative REST API.

I comandi vengono normalmente inviati utilizzando messaggi non persistenti, quindi entrambi gli oggetti del comando di sistema devono avere l'attributo DEFPSIST (NO) in modo che le applicazioni che li utilizzano (incluse le applicazioni fornite come il programma di utilità e le operazioni e i pannelli di controllo) ricevano messaggi non persistenti per impostazione predefinita. Se si dispone di un'applicazione che utilizza messaggi persistenti per i comandi, impostare l'attributo DEFTYPE (PERMDYN) per la coda di risposta, poiché i messaggi di risposta a tali comandi sono persistenti.

Oggetti di gestione del sistema

I nomi degli oggetti di gestione del sistema iniziano con il carattere SYSTEM.ADMINADMIN.

Esistono sette oggetti di gestione del sistema:

- Il SISTEMA SYSTEM.ADMIN.CHANNEL.EVENT
- Il SISTEMA SYSTEM.ADMIN.COMMAND.EVENT
- Il SISTEMA SYSTEM.ADMIN.CONFIG.EVENT
- Il SISTEMA SYSTEM.ADMIN.PERFM.EVENT
- Il SISTEMA SYSTEM.ADMIN.QMGR.EVENT
- Il SISTEMA SYSTEM.ADMIN.TRACE.ROUTE.QUEUE QUEUE
- Il SISTEMA SYSTEM.ADMIN.ACTIVITY.QUEUE QUEUE

Code canali

Per utilizzare l'accodamento distribuito, è necessario definire i seguenti oggetti:

- Una coda locale con il nome SYSTEM.CHANNEL.SYNCQ, che viene utilizzato per mantenere i numeri di sequenza e gli identificatori LUWID (logical units of work) dei canali. Per migliorare le prestazioni del canale, è necessario definire questa coda con un tipo di indice MSGID (come mostrato nella definizione della coda di esempio fornita).
- Una coda locale con il nome SYSTEM.CHANNEL.INITQ, utilizzato per i comandi del canale.

Non è possibile definirle come code condivise.

Code cluster

Per utilizzare i cluster IBM MQ , è necessario definire i seguenti oggetti:

- Una coda locale denominata SYSTEM.CLUSTER.COMMAND.QUEUE, utilizzato per comunicare le modifiche del repository tra i gestori code. I messaggi scritti in questa coda contengono aggiornamenti ai dati del repository da applicare alla copia locale del repository o richieste di dati del repository.
- Una coda locale denominata SYSTEM.CLUSTER.REPOSITORY.QUEUE, utilizzato per conservare una copia permanente del repository.
- Una coda locale denominata SYSTEM.CLUSTER.TRANSMIT.QUEUE, che è la coda di trasmissione per tutte le destinazioni nel cluster. Per motivi di prestazioni, è necessario specificare questa coda con un tipo di indice CORRELID (come mostrato nella definizione della coda di esempio).

Queste code generalmente contengono un numero elevato di messaggi.

Non è possibile definirle come code condivise.

Code del gruppo di condivisione code

Per utilizzare i canali condivisi e l'accodamento all'interno del gruppo, è necessario definire i seguenti oggetti:

- Una coda condivisa con il nome SYSTEM.QSG.CHANNEL.SYNCQ, che viene utilizzato per conservare le informazioni di sincronizzazione per i canali condivisi.
- Una coda condivisa con il nome SYSTEM.QSG.TRANSMIT.QUEUE, che viene utilizzato come coda di trasmissione per l'accodamento all'interno del gruppo. Se si è in esecuzione in un gruppo di condivisione code, è necessario definire questa coda, anche se non si utilizza l'accodamento all'interno del gruppo.

Classi di memoria

Si consiglia di definire le seguenti sei classi di memoria. È necessario definirne quattro perché sono richiesti da IBM MQ. Le altre definizioni di classe di archiviazione sono consigliate perché sono utilizzate in definizioni di coda campione.

DEFAULT (obbligatorio)
Questa classe di archiviazione viene utilizzata per tutte le code di messaggi che non sono critiche per le prestazioni e che non rientrano in nessuna delle altre classi di archiviazione. È anche la classe di memoria predefinita fornita se non se ne specifica una durante la definizione di una coda.

NODEFINE (obbligatorio)
Questa classe di memorizzazione viene utilizzata se la classe di memorizzazione specificata quando si definisce una coda non è definita.

REMOTO (obbligatorio)
Questa classe di archiviazione viene utilizzata principalmente per le code di trasmissione, ossia le code correlate al sistema con messaggi critici per le prestazioni di breve durata.

SYSLNGLV
Questa classe di memoria è utilizzata per messaggi di lunga durata, critici per le prestazioni.

SISTEMA (obbligatorio)
Questa classe di memoria viene utilizzata per le code messaggi relative al sistema, critiche per le prestazioni, ad esempio SYSTEM.CHANNEL.SYNQ e SYSTEM.CLUSTER.* .

SSVOLAT
Questa classe di storage viene utilizzata per messaggi di breve durata, critici per le prestazioni.

È possibile modificare i relativi attributi e aggiungere altre definizioni di classe di archiviazione come richiesto.

Definizione della coda di messaggi non recapitabili dell'oggetto di sistema

La coda di messaggi non recapitabili viene utilizzata se la destinazione del messaggio non è valida. IBM MQ inserisce tali messaggi in una coda locale denominata coda di messaggi non instradabili. Anche se avere una coda di messaggi non recapitabili non è obbligatorio, è necessario considerarla essenziale, soprattutto se si utilizza l'accodamento distribuito o uno dei bridge IBM MQ .

Non definire la coda di messaggi non recapitabili come coda condivisa. Un inserimento in una coda locale su un gestore code potrebbe essere inserito nella coda dei messaggi non recapitabili. Se la coda di messaggi non recapitabili era una coda condivisa, un gestore code di messaggi non recapitabili su un sistema diverso potrebbe elaborare il messaggio e inserirlo in una coda con lo stesso nome, ma poiché si trova su un gestore code diverso, potrebbe essere la coda non corretta o avere un profilo di protezione diverso. Se la coda non esiste, non sarà possibile rielaborarla.

Se si decide di definire una coda di messaggi non recapitabili, è necessario indicare anche il nome del gestore code. A tale scopo, utilizzare il comando ALTER QMGR DEADQ (*nome - coda*). Per ulteriori informazioni, consultare Visualizzazione e modifica degli attributi del gestore code.

Coda di trasmissione predefinita

La coda di trasmissione predefinita viene utilizzata quando non è disponibile nessun' altra coda di trasmissione adatta per l'invio di messaggi ad un altro gestore code. Se si definisce una coda di trasmissione predefinita, occorre anche definire un canale che serva la coda. In caso contrario, i messaggi che vengono inseriti nella coda di trasmissione predefinita non vengono trasmessi al gestore code remoto e rimangono nella coda.

Se si decide di definire una coda di trasmissione predefinita, è necessario indicare anche il nome del gestore code. A tale scopo, utilizzare il comando ALTER QMGR.

Code interne

- **Coda dati in sospeso**
  - Una coda definita per uso interno, SYSTEM.PENDING.DATA.QUEUE, supporta l'utilizzo di sottoscrizioni durevoli in un ambiente di pubblicazione / sottoscrizione JMS .
- **JMS 2.0 coda di staging ritardo recapito**
  - Se viene utilizzata la funzionalità di ritardo del recapito fornita da JMS 2.0 , viene utilizzata una coda di staging interna, SYSTEM.DDELAY.LOCAL.QUEUE, deve essere definito. Questa coda viene utilizzata dal gestore code per memorizzare temporaneamente i messaggi inviati con un ritardo di consegna diverso da zero fino a quando il ritardo di consegna non viene completato e il messaggio viene inserito nella destinazione di destinazione. Una definizione di esempio per questa coda viene fornita, commentata, in CSQ4INSG.
  - Quando si definisce SYSTEM.DDELAY.LOCAL.QUEUE , è necessario impostare gli attributi STGCLASS, MAXMSGL e MAXDEPTH per il numero anticipato di messaggi che verranno inviati con ritardo di consegna. Inoltre, quando si definisce SYSTEM.DDELAY.LOCAL.QUEUE QUEUE garantisce che solo il gestore code può inserire messaggi in questa coda. Accertarsi che nessun identificativo utente disponga dell'autorità per inserire i messaggi in questa coda.

Coda di autenticazione canale

Per l'utilizzo interno dell'autenticazione di canale, il SISTEMA SYSTEM.CHLAUTH.DATA.QUEUE QUEUE è obbligatoria. Le definizioni di esempio vengono fornite con IBM MQ per facilitare la definizione di questi oggetti. Questo esempio è descritto in CSQ4INSA, che definisce anche alcune regole predefinite.

z/OS Ottimizzazione del gestore code su IBM MQ for z/OS

Esistono pochi semplici passi che è possibile eseguire per garantire che il gestore code sia ottimizzato per evitare problemi di prestazioni di base.

Esistono diversi modi in cui è possibile migliorare le prestazioni del gestore code, che sono controllati dagli attributi del gestore code impostati dal comando ALTER QMGR. Questa sezione contiene informazioni su come eseguire questa operazione impostando il numero massimo di messaggi consentiti sul gestore code o eseguendo la 'manutenzione' sul gestore code. IBM MQ SupportPac MP16 - IBM MQ per z/OS Capacity planning & ottimizzazione fornisce ulteriori informazioni sulle prestazioni e sull'ottimizzazione.

Punti di sincronizzazione

Uno dei ruoli del gestore code è il controllo del punto di sincronizzazione in un'applicazione. Un'applicazione crea un'unità di lavoro contenente un numero qualsiasi di chiamate MQPUT o MQGET terminate con una chiamata MQCMIT.

Poiché il numero di chiamate MQPUT o MQGET nell'ambito di un MQCMIT aumenta, il costo delle prestazioni del commit aumenta in modo significativo. Le applicazioni, in genere, devono essere progettate in modo da non utilizzare MQPUT/MQGET per un numero elevato di messaggi in un singolo punto di sincronizzazione.

È possibile limitare amministrativamente il numero di messaggi all'interno di un singolo punto di sincronizzazione utilizzando l'attributo del gestore code MAXUMSGS. Se un'applicazione supera questo limite, riceve MQRC_SYNCPOINT_LIMIT_RAGGIUNTA nella chiamata MQPUT,MQPUT1o MQGET che supera il limite. L'applicazione deve quindi emettere MQCMIT o MQBACK come appropriato.

Il valore predefinito di MAXUMSGS è 10000. Questo valore può essere ridotto se si desidera applicare un limite inferiore, che può anche aiutare a proteggersi dalle applicazioni in loop. Prima di ridurre MAXUMSGS assicurarsi di comprendere le applicazioni esistenti per assicurarsi che non superino il limite o che sia possibile tollerare il codice di ritorno MQRC_SYNCPOINT_LIMIT_SNC

Messaggi scaduti

I messaggi scaduti vengono eliminati dalla successiva chiamata MQGET appropriata. Tuttavia, se non si verifica tale chiamata, i messaggi scaduti non vengono eliminati e, per alcune code, in particolare per quelle in cui il richiamo dei messaggi viene eseguito da MessageId, CorrelIdo GroupId e la coda è indicizzata per le prestazioni, molti messaggi scaduti possono accumularsi. Il gestore code può eseguire periodicamente la scansione di qualsiasi coda per rilevare la presenza di messaggi scaduti, che vengono quindi eliminati. È possibile scegliere la frequenza di questa scansione, se del caso. Ci sono due modi per farlo:

Richiesta esplicita
: È possibile controllare quali code vengono sottoposte a scansione e quando. Immettere il comando REFRESH QMGR TYPE (SCADENZA), specificando la coda o le code di cui si desidera eseguire la scansione.

Scansione periodica
: È possibile specificare un intervallo di scadenza nell'oggetto gestore code utilizzando l'attributo EXPRYINT. Il gestore code conserva le informazioni sui messaggi scaduti su ciascuna coda e sa a che ora vale la pena eseguire una scansione dei messaggi scaduti. Ogni volta che viene raggiunto l'intervallo EXPRYINT, il gestore code ricerca le code candidate che vale la pena scannerizzare per i messaggi scaduti ed esegue la scansione solo delle code che ritiene utili. Non esegue la scansione di tutte le code. Ciò evita che il tempo del processore venga sprecato in scansioni non necessarie.

  Le code condivise vengono sottoposte a scansione solo da un gestore code nel gruppo di condivisione code. In genere, la scansione viene eseguita dal primo gestore code da riavviare o dal primo gestore code che ha impostato EXPRYINT.

  Nota: È necessario impostare lo stesso valore EXPRYINT per tutti i gestori code all'interno di un gruppo di condivisione code.

z/OS Definizioni di esempio fornite con IBM MQ for z/OS

Utilizzare questo argomento come riferimento per il JCL di esempio e il codice fornito con IBM MQ for z/OS.

Le definizioni di esempio riportate di seguito vengono fornite con IBM MQ nella libreria thlqual.SCSQPROC . È possibile utilizzarle per definire gli oggetti di sistema e per personalizzarli. È possibile includere alcuni di essi nei dataset di input di inizializzazione (descritti in Comandi di inizializzazione).

Tabella 22. Definizioni di esempio IBM MQ per oggetti di sistema	
serie di dati di input di inizializzazione	**Nome esempio**
CSQINP1	CSQ4INP1 CSQ4INPR
CSQINP2	CSQ4INSA CSQ4INYS [1] CSQ4INSX CSQ4INSG CSQ4INSR CSQ4INSS CSQ4INSJ CSQ4INSM CSQ4INYG CSQ4INYR CSQ4INYC CSQ4INYD CSQ4INSC
CSQINPT	CSQ4INST CSQ4INYT
Altro	CSQ4DISP CSQ4INPX CSQ4IVPQ CSQ4IVPG CSQ4MSTR CSQ4MSRR CSQ4QMIN

Nota:

1. L'ordine di queste definizioni di esempio è importante: si verifica un errore se INYS, INSX e INSG sono ordinati in modo non corretto.

Esempi CSQINP1

Utilizzare il dataset CSQINP1 di esempio thlqual.SCSQPROC(CSQ4INP1) quando si utilizza una serie di pagine per ciascuna classe di messaggi o thlqual.SCSQPROC(CSQ4INPR) quando si utilizzano più serie di pagine per le classi di messaggi principali. Contiene definizioni di pool di buffer, serie di pagine per le associazioni di pool di buffer e un comando ALTER SECURITY. Includere l'esempio nella concatenazione CSQINP1 della procedura dell'attività avviata del gestore code.

Esempi CSQINP2

Esempio di oggetto di sistema CSQ4INSG

Il dataset CSQINP2 di esempio thlqual.SCSQPROC(CSQ4INSG) contiene le definizioni per i seguenti oggetti di sistema per uso generale:

- Oggetti predefiniti del sistema

- Oggetti comando di sistema
- Oggetti di gestione del sistema
- Altri oggetti per uso di sistema

È necessario definire gli oggetti in questo esempio, ma è necessario farlo solo una volta quando il sottosistema viene avviato per la prima volta. L'inclusione delle definizioni nel dataset CSQINP2 è il modo migliore per farlo. Vengono mantenuti durante l'arresto e il riavvio del gestore code. Non è necessario modificare i nomi degli oggetti, ma è possibile modificarne gli attributi, se necessario.

Quando vengono soddisfatte le seguenti condizioni, viene inserito un messaggio nel SISTEMA SYSTEM.DURABLE.SUBSCRIBER.QUEUE (anche se la sottoscrizione di pubblicazione non è attiva):

- L'attributo QMGR PSMODE è impostato su DISABLED
- L'istruzione CSQ4INST dell'oggetto di esempio `DEFINE SUB('SYSTEM.DEFAULT.SUB')` è presente.

Per evitare ciò, eliminare o commentare l'istruzione `DEFINE SUB('SYSTEM.DEFAULT.SUB')` .

La coda di staging di ritardo recapito JMS 2.0 , SYSTEM.DDELAY.LOCAL.QUEUE deve essere definito solo se viene utilizzato il ritardo di consegna JMS 2.0 . Per impostazione predefinita, la definizione della coda è commentata, che è possibile eliminare il commento, se necessario.

Esempio di autenticazione e oggetto di sistema CSQ4INSA

Il dataset CSQINP2 di esempio thlqual.SCSQPROC(CSQ4INSA) contiene la definizione della coda del sistema di autenticazione di canale. Questa coda contiene i record di autenticazione di canale. Contiene anche le regole di autenticazione del canale predefinite.

È necessario definire gli oggetti in questo esempio se CHLAUTH è ENABLED sul gestore code e si desidera eseguire i canali oppure si desidera eseguire il record SET o DISPLAY CHLAUTH. È necessario definirle una sola volta quando il sottosistema viene avviato per la prima volta. L'inclusione delle definizioni nel dataset CSQINP2 è il modo migliore per farlo. Vengono conservati durante l'arresto e il riavvio di un gestore code, non è necessario modificare il nome della coda.

Esempio di oggetto di sistema CSQ4INSS

È possibile definire ulteriori oggetti di sistema se si utilizzano gruppi di condivisione code.

Il dataset di esempio thlqual.SCSQPROC(CSQ4INSS) contiene i comandi di esempio da utilizzare con le strutture CF e una serie di definizioni per gli oggetti di sistema richiesti per i canali condivisi e l'accodamento all'interno del gruppo.

Non è possibile utilizzare questo esempio così com' è; è necessario personalizzarlo prima dell'utilizzo. Quindi, è possibile includere questo membro nella concatenazione DD CSQINP2 della procedura di avvio del gestore code oppure utilizzarlo come input per la funzione COMMAND del programma di utilità CSQUTIL per immettere i comandi richiesti.

Quando si definiscono oggetti condivisi o di gruppo, è necessario includerli nella concatenazione DD CSQINP2 per un solo gestore code nel gruppo di condivisione code.

Esempio di oggetto di sistema CSQ4INSX

È necessario definire ulteriori oggetti di sistema se si stanno utilizzando l'accodamento distribuito e il clustering.

Il dataset di esempio thlqual.SCSQPROC(CSQ4INSX) contiene le definizioni di coda richieste. È possibile includere questo membro nella concatenazione DD CSQINP2 della procedura di avvio del gestore code oppure utilizzarlo come input per la funzione COMMAND nel programma di utilità CSQUTIL per immettere i comandi DEFINE richiesti.

Esistono due tipi di definizioni oggetto:

- SYSTEM.CHANNEL.xx, necessario per qualsiasi accodamento distribuito

- SYSTEM.CLUSTER.xx, necessario per il clustering

Esempio di oggetto CSQ4INSJ system JMS

Definisce le code utilizzate nel dominio di pubblicazione / sottoscrizione di JMS .

Esempio di oggetto di sistema CSQ4INSM

Se si utilizza la sicurezza avanzata dei messaggi, è necessario definire ulteriori oggetti di sistema. Il dataset di esempio thlqual.SCSQPROC(CSQ4INSM) contiene le definizioni di coda richieste.

Esempio di oggetto CSQ4INSR

Definisce le code utilizzate da WebSphere Application Server e dai broker.

Esempio di oggetto CSQ4INYD

Se si sta utilizzando l'accodamento distribuito ed è necessario impostare le proprie code, processi e canali.

Il dataset di esempio thlqual.SCSQPROC(CSQ4INYD) contiene le definizioni di esempio che è possibile utilizzare per personalizzare gli oggetti di accodamento distribuiti. Comprende:

- Una serie di definizioni per l'estremità di invio
- Un insieme di definizioni per l'estremità ricevente
- Un insieme di definizioni per l'utilizzo dei client

non è possibile utilizzare questo esempio così com' è - è necessario personalizzarlo prima di utilizzarlo. Quindi, è possibile includere questo membro nella concatenazione DD CSQINP2 della procedura di avvio del gestore code oppure utilizzarlo come input per la funzione COMMAND del programma di utilità CSQUTIL per immettere i comandi DEFINE richiesti. (Questo è preferibile perché significa che non è necessario ridefinire questi oggetti ogni volta che si riavvia il gestore code).

Esempio di oggetto CSQ4INYC

Se si utilizza il clustering, le definizioni equivalenti alle definizioni di canale e alle definizioni di coda remota dell'accodamento distribuito vengono create automaticamente, quando necessario. Tuttavia, sono necessarie alcune definizioni di canale manuali - un canale ricevente del cluster per il cluster e una definizione mittente del cluster per almeno un gestore code del repository del cluster.

Il dataset di esempio: thlqual.SCSQPROC(CSQ4INYC), contiene le seguenti definizioni di esempio che è possibile utilizzare per personalizzare gli oggetti cluster:

- Definizioni per il gestore code
- Definizioni per il canale ricevente
- Definizioni per il canale di invio
- Definizioni per code cluster
- Definizioni per elenchi di cluster

non è possibile utilizzare questo esempio così com' è - è necessario personalizzarlo prima di utilizzarlo. Quindi, è possibile includere questo membro nella concatenazione DD CSQINP2 della procedura di avvio del gestore code oppure utilizzarlo come input per la funzione COMMAND del programma di utilità CSQUTIL per immettere i comandi DEFINE richiesti. Ciò è preferibile perché significa che non è necessario ridefinire questi oggetti ogni volta che si riavvia IBM MQ.

Esempio di oggetto CSQ4INYG

Il dataset di esempio: thlqual.SCSQPROC(CSQ4INYG) contiene le seguenti definizioni di esempio che è possibile utilizzare per personalizzare i propri oggetti per un utilizzo generale:

- Coda di messaggi non recapitabili
- Coda di trasmissione predefinita

- CICS oggetti adattatore

non è possibile utilizzare questo esempio così com' è - è necessario personalizzarlo prima di utilizzarlo. Quindi, è possibile includere questo membro nella concatenazione DD CSQINP2 della procedura di avvio del gestore code oppure utilizzarlo come input per la funzione COMMAND del programma di utilità CSQUTIL per immettere i comandi DEFINE richiesti. Ciò è preferibile perché significa che non è necessario ridefinire questi oggetti ogni volta che si riavvia IBM MQ.

Oltre alle definizioni di esempio, è possibile utilizzare le definizioni degli oggetti di sistema come base per le proprie definizioni delle risorse. Ad esempio, è possibile creare una copia di lavoro di SYSTEM.DEFAULT.LOCAL.QUEUE e denominarlo MY.DEFAULT.LOCAL.QUEUE. È quindi possibile modificare uno qualsiasi dei parametri in questa copia come richiesto. È quindi possibile immettere un comando DEFINE in base al metodo scelto, purché si disponga dell'autorizzazione per creare risorse di quel tipo.

Coda di trasmissione predefinita
Leggere la descrizione Coda di trasmissione predefinita prima di decidere se si desidera definire una coda di trasmissione predefinita.

- Se si decide che si desidera definire una coda di trasmissione predefinita, è necessario definire anche un canale per utilizzarla.
- Se si decide che non si desidera definirne uno, ricordarsi di rimuovere l'istruzione DEFXMITQ dal comando ALTER QMGR nell'esempio.

CICS oggetti adattatore
L'esempio definisce una coda di avvio denominata CICS01.INITQ. Questa coda viene utilizzata dalla transazione CKTI fornita da IBM MQ . È possibile modificare il nome di questa coda; tuttavia, deve corrispondere al nome specificato nella tabella di inizializzazione del sistema CICS (SIT) o nella sostituzione SYSIN nell'istruzione INITPARM.

Esempi di oggetti CSQ4INYS/CSQ4INYR

Definizioni della classe di memoria per l'utilizzo:

- una serie di pagine per ogni classe del messaggio
- più serie di pagine per le classi principali di messaggi

Ad esempio, SYSTEM.COMMAND.INPUT utilizza STGCLASS ('SYSVOLAT') e SYSTEM.CLUSTER.TRANSMIT.QUEUE utilizza STGCLASS ('REMOTE'). In CSQ4INYS, entrambe le classi di memoria utilizzano la stessa serie di pagine. In CSQ4INYR, tali classi di memoria utilizzano diverse serie di pagine per ridurre l'impatto del riempimento della coda di trasmissione.

Esempi di CSQINPT

CSQ4INST
Il dataset di esempio: thlqual.SCSQPROC(CSQ4INST) contiene le definizioni per la sottoscrizione predefinita del sistema.

È necessario definire l'oggetto in questo esempio, ma è necessario farlo solo una volta quando il motore di pubblicazione / sottoscrizione viene avviato per la prima volta. L'inclusione della definizione nel dataset CSQINPT è il modo migliore per eseguire questa operazione. Viene conservata durante l'arresto e il riavvio del gestore code. Non è necessario modificare il nome dell'oggetto, ma è possibile modificarne gli attributi, se necessario.

CSQ4INYT
Il dataset di esempio: thlqual.SCSQPROC(CSQ4INYT) contiene una serie di comandi che è possibile eseguire quando viene avviato il motore di pubblicazione / sottoscrizione. Questo esempio visualizza le informazioni su argomento e sottoscrizione.

Altro

Esempio di visualizzazione CSQ4DISP

Il dataset di esempio: thlqual.SCSQPROC(CSQ4DISP) contiene una serie di comandi DISPLAY generici che visualizzano tutte le risorse definite sul gestore code. Ciò include le definizioni per tutti gli oggetti IBM MQ e le definizioni come le classi di archiviazione e la traccia. Questi comandi possono generare una grande quantità di output. È possibile utilizzare questo esempio nel dataset CSQINP2 o come input per la funzione COMMAND del programma di utilità CSQUTIL.

Esempio CSQ4INPX

Il dataset di esempio: thlqual.SCSQPROC(CSQ4INPX) contiene una serie di comandi che è possibile eseguire ogni volta che viene avviato l'iniziatore di canali. È necessario personalizzare questo esempio prima dell'utilizzo; è quindi possibile includerlo nel dataset CSQINPX per l'iniziatore di canali.

Esempi CSQ4IVPQ e CSQ4IVPG

I dataset di esempio: thlqual.SCSQPROC(CSQ4IVPQ) e thlqual.SCSQPROC(CSQ4IVPG) contengono serie di comandi DEFINE richiesti per eseguire i programmi di verifica dell'installazione (IVP).

È possibile includere questi esempi nel dataset CSQINP2 . Una volta eseguiti correttamente gli IVP, non è necessario eseguirli nuovamente ogni volta che il gestore code viene riavviato. Pertanto, non è necessario conservare questi esempi nella concatenazione CSQINP2 in modo permanente.

Esempi CSQ4MSTR e CSQ4MSRR

Si tratta di procedure di attività avviate di esempio per il gestore code: thlqual.SCSQPROC(CSQ4MSTR) e thlqual.SCSQPROC(CSQ4MSRR).

CSQ4MSRR utilizza CSQ4INYR nella concatenazione CSQINP2 in modo che le code importanti vengano distribuite su diverse serie di pagine.

È possibile rimuovere i commenti, in modo da poter utilizzare la scheda CSQMINI per i gestori code appena creati, se necessario.

Esempio CSQ4QMIN

Un dataset QMINI di esempio, thlqual.SCSQPROC(CSQ4QMIN).

Consultare Serie di dati QMINI per i dettagli relativi alla serie di dati QMINI e alla stanza **`TransportSecurity`** .

z/OS Ripristino e riavvio su z/OS

Utilizzare i link in questo argomento per informazioni sulle funzioni di IBM MQ for z/OS per il riavvio e il recupero.

IBM MQ for z/OS dispone di funzioni avanzate per il riavvio e il recupero. Per informazioni sul modo in cui un gestore code viene ripristinato dopo l'arresto e su cosa accade quando viene riavviato, consultare i topic secondari riportati di seguito:

- “Modalità di modifica dei dati in IBM MQ for z/OS” a pagina 259
- “Come viene mantenuta la congruenza in IBM MQ for z/OS” a pagina 260
- “Cosa accade durante la terminazione in IBM MQ for z/OS” a pagina 262
- “Cosa accade durante il riavvio e il ripristino in IBM MQ for z/OS” a pagina 264
- “Come vengono risolte le unità di ripristino in dubbio” a pagina 266
- “Ripristino coda condivisa” a pagina 268

Concetti correlati

z/OS Azioni di recupero IBM MQ for z/OS

Attività correlate

Pianificazione del backup e del ripristino

z/OS Amministrazionez/OS

Riferimenti correlati

z/OS Messaggi per IBM MQ for z/OS

z/OS Modalità di modifica dei dati in IBM MQ for z/OS

IBM MQ for z/OS deve interagire con altri sottosistemi per mantenere tutti i dati congruenti. Questo argomento contiene informazioni su *unità di recupero*, quali sono e come vengono utilizzate nei *backout*.

Unità di recupero

Un' *unità di ripristino* è l'elaborazione eseguita da un singolo gestore code per un programma applicativo, che modifica i dati IBM MQ da un punto di congruenza all'altro. Un *punto di congruenza* - chiamato anche *punto di sincronizzazione* o *punto di commit* - è un momento temporale in cui tutti i dati recuperabili a cui accede un programma applicativo sono congruenti.

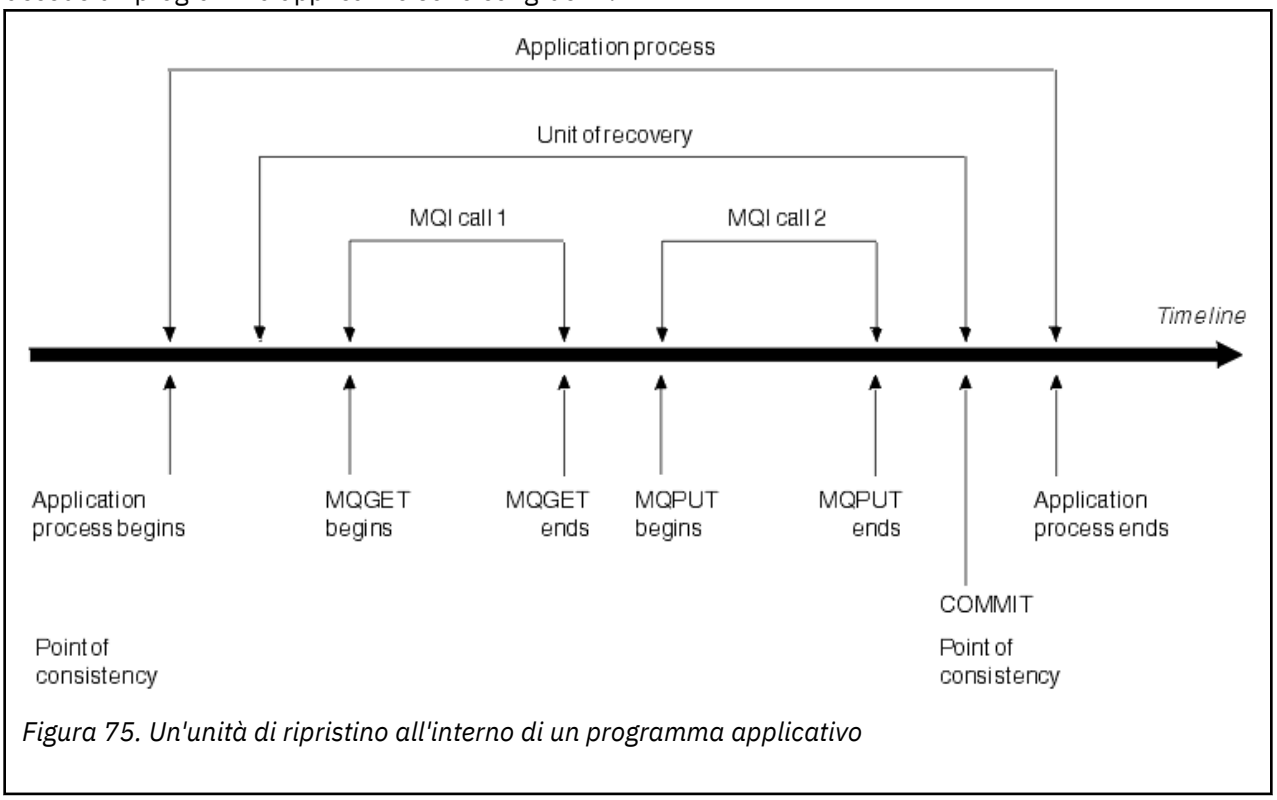


Figura 75. Un'unità di ripristino all'interno di un programma applicativo

Un'unità di ripristino inizia con la prima modifica ai dati dopo l'inizio del programma o dopo il punto di coerenza precedente; termina con un punto di coerenza successivo. Figura 75 a pagina 259 mostra le relazioni tra unità di recupero, il punto di congruenza e un programma di applicazione. In questo esempio, il programma applicativo apporta modifiche alle code tramite le chiamate MQI 1 e 2. Il programma applicativo può includere più di un'unità di ripristino o solo una. Tuttavia, qualsiasi unità di ripristino completa termina in un punto di commit.

Ad esempio, una transazione bancaria trasferisce fondi da un conto all'altro. In primo luogo, il programma sottrae l'importo dal primo conto, conto A. Successivamente, aggiunge l'importo al secondo conto, B. Dopo aver sottratto l'ammontare da A, i due account sono incongruenti e IBM MQ non può eseguire il commit. Diventano congruenti quando l'importo viene aggiunto al conto B. Quando entrambi i passi sono completi, il programma può annunciare un punto di coerenza attraverso un commit, rendendo le modifiche visibili ad altri programmi applicativi.

La normale chiusura di un programma applicativo causa automaticamente un punto di coerenza. Alcune richieste di programmi in CICS e IMS causano anche un punto di congruenza, ad esempio EXEC CICS SYNCPOINT.

Lavoro di backout

Se si verifica un errore all'interno di un'unità di recupero, IBM MQ rimuove tutte le modifiche ai dati, riportando i dati al loro stato all'inizio dell'unità di ripristino, ovvero IBM MQ esegue il backout del lavoro. Gli eventi vengono mostrati in Figura 76 a pagina 260.

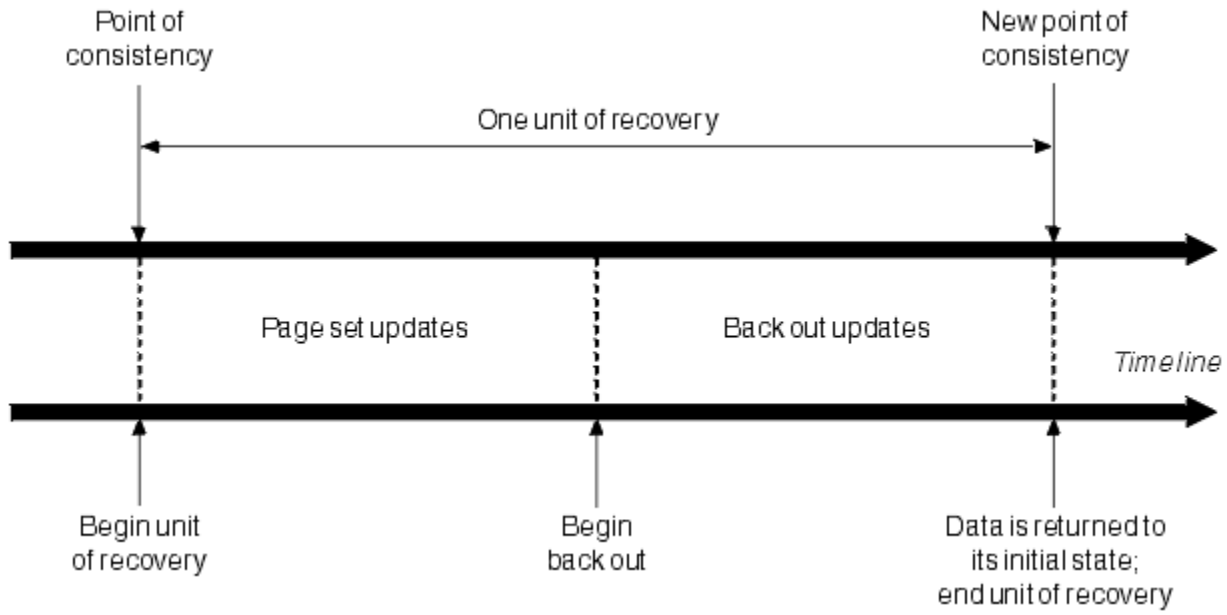


Figura 76. Un'unità di recupero che mostra il backout

z/OS Come viene mantenuta la congruenza in IBM MQ for z/OS

I dati in IBM MQ for z/OS devono essere congruenti con batch, CICS, IMSo TSO. Qualsiasi dato modificato in uno deve corrispondere a una modifica nell'altro.

Prima che un sistema esegue il commit dei dati modificati, deve sapere che l'altro sistema può apportare la modifica corrispondente. Quindi, i sistemi devono comunicare.

Durante un *commit a due fasi* (ad esempio in CICS), un sottosistema coordina il processo. Tale sottosistema viene chiamato il *coordinatore*; l'altro è il *partecipante*. CICS o IMS è sempre il coordinatore nelle interazioni con IBM MQe IBM MQ è sempre il partecipante. Nell'ambiente batch o TSO, IBM MQ può partecipare a protocolli di commit a due fasi coordinati da z/OS RRS.

Durante un *commit a fase singola* (ad esempio in TSO o batch), IBM MQ è sempre il coordinatore nelle interazioni e controlla completamente il processo di commit.

In un ambiente WebSphere Application Server , la semantica dell'oggetto sessione JMS determina se viene utilizzata la coordinazione di commit a fase singola o a due fasi.

Congruenza con CICS o IMS

La connessione tra IBM MQ e CICS o IMS supporta i seguenti protocolli del punto di sincronizzazione:

- Commit a due fasi - per le transazioni che aggiornano le risorse di proprietà di più di un gestore risorse.

Questo è il protocollo syncpoint distribuito standard. Comporta più registrazione e flussi di messaggi rispetto a un commit a fase singola.

- Commit a fase singola - per le transazioni che aggiornano le risorse di proprietà di un singolo gestore risorse (IBM MQ).

  Questo protocollo è ottimizzato per la registrazione e i flussi di messaggi.

- Bypass del punto di sincronizzazione - per le transazioni che coinvolgono IBM MQ ma che non fanno nulla nel gestore code che richiede un punto di sincronizzazione (ad esempio, sfogliare una coda).

In ogni caso, CICS o IMS agisce come gestore del punto di sincronizzazione.

Le fasi del commit a due fasi che IBM MQ utilizza per comunicare con CICS o IMS sono le seguenti:

1. Nella fase 1, ogni sistema determina indipendentemente se ha registrato abbastanza informazioni di ripristino nel proprio log e può eseguire il commit del proprio lavoro.

   Al termine della fase, i sistemi comunicano. Se sono d'accordo, ognuno inizia la fase successiva.
2. Nella fase 2, le modifiche vengono rese permanenti. Se uno dei sistemi termina in modo anomalo durante la fase 2, l'operazione viene completata dal processo di ripristino durante il riavvio.

Illustrazione del processo di commit in due fasi

Figura 77 a pagina 261 illustra il processo di commit in due fasi. Gli eventi nel coordinatore CICS o IMS vengono mostrati nella riga superiore, gli eventi in IBM MQ nella riga inferiore.

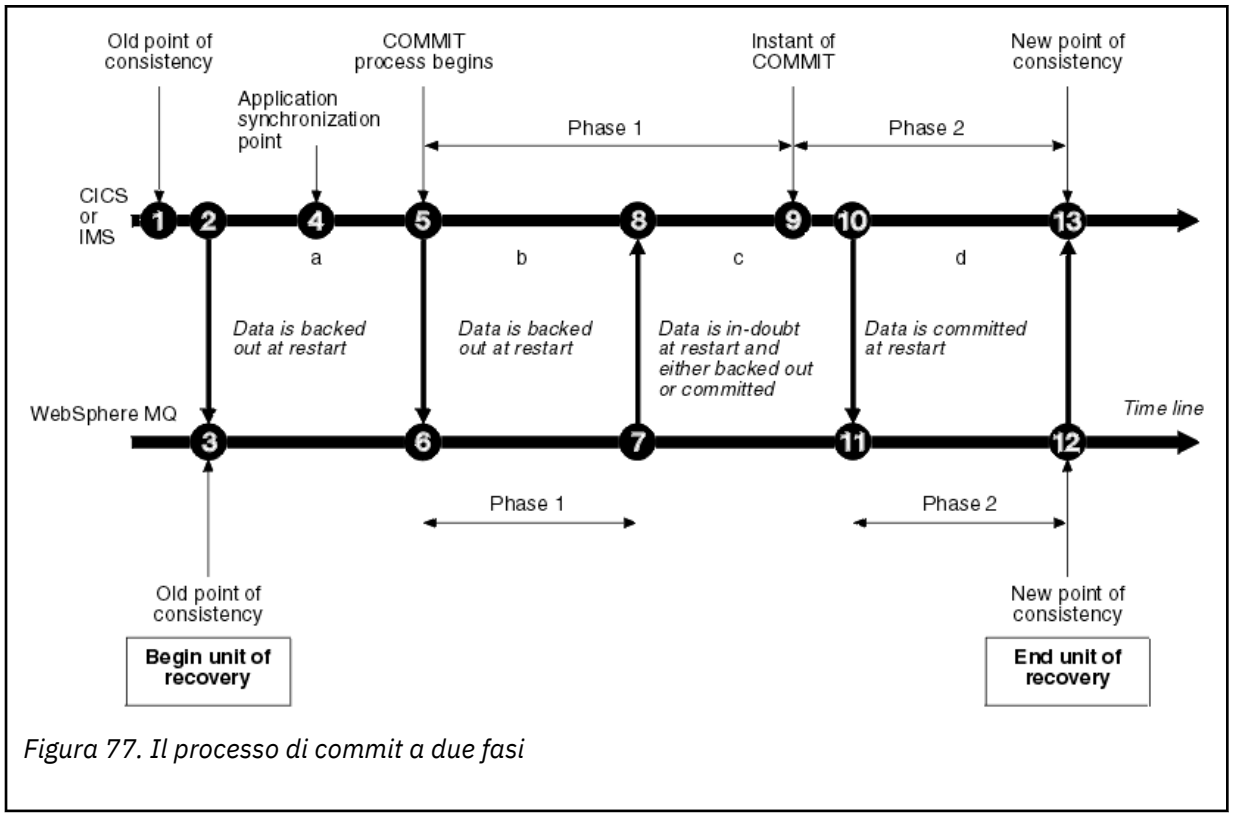


Figura 77. Il processo di commit a due fasi

I numeri nella sezione seguente sono collegati a quelli mostrati nella figura.

1. I dati nel coordinatore si trovano in un punto di coerenza.
2. Un programma applicativo nel coordinatore richiama IBM MQ per aggiornare una coda aggiungendo un messaggio.
3. Questo avvia un'unità di recupero in IBM MQ.
4. L'elaborazione continua nel coordinatore fino a quando non viene raggiunto un punto di sincronizzazione dell'applicazione.

5. Il coordinatore avvia quindi l'elaborazione del commit. I programmi CICS utilizzano un comando SYNCPOINT o una normale chiusura dell'applicazione per avviare il commit. I programmi IMS possono avviare il commit utilizzando una chiamata CHKP, una chiamata SYNC, una chiamata GET UNIQUE all'IOPCB o una normale chiusura dell'applicazione. Inizia la fase 1 dell'elaborazione del commit.
6. Come il coordinatore inizia l'elaborazione fase 1, così fa IBM MQ.
7. IBM MQ completa correttamente la fase 1, scrive questo fatto nel relativo log e notifica il coordinatore.
8. Il coordinatore riceve la notifica.
9. Il coordinatore ha completato con esito positivo l'elaborazione della fase 1. Ora entrambi i sottosistemi accettano di eseguire il commit delle modifiche dei dati, perché entrambi hanno completato la fase 1 e potrebbero recuperare da eventuali errori. Il coordinatore registra nel suo registro l'istante del commit - la decisione irrevocabile dei due sottosistemi di apportare le modifiche.

   Il coordinatore inizia ora la fase 2 dell'elaborazione - l'impegno effettivo.
10. Il coordinatore notifica a IBM MQ di iniziare la fase 2.
11. IBM MQ registra l'inizio della fase 2.
12. La fase 2 è stata completata correttamente e questo è ora un nuovo punto di coerenza per IBM MQ. IBM MQ notifica quindi al coordinatore di aver terminato l'elaborazione della fase 2.
13. Il coordinatore termina l'elaborazione della fase 2. I dati controllati da entrambi i sottosistemi sono ora coerenti e disponibili per altre applicazioni.

Come viene mantenuta la coerenza dopo una terminazione anomala

Quando un gestore code viene riavviato dopo una chiusura anomala, deve determinare se eseguire il commit o il backout delle unità di recupero che erano attive al momento della chiusura. Per alcune unità di recupero, IBM MQ ha informazioni sufficienti per prendere la decisione. Per gli altri, non lo è, e deve ottenere informazioni dal coordinatore quando la connessione viene ristabilita.

Figura 77 a pagina 261 mostra quattro periodi nelle due fasi: a, b, c e d. Lo stato di un'unità di recupero dipende dal periodo in cui si è verificata la chiusura. Lo stato può essere uno dei seguenti:

In volo
Il gestore code è terminato prima di terminare la fase 1 (periodo a o b); durante il riavvio, IBM MQ esegue il backout degli aggiornamenti.

In dubbio
Il gestore code è terminato dopo la fase finale 1 e prima della fase iniziale 2 (punto c); solo il coordinatore sa se l'errore si è verificato prima o dopo il commit (punto 9). Se si è verificato in precedenza, IBM MQ deve eseguire il backout delle modifiche; se si è verificato in seguito, IBM MQ deve apportare le modifiche ed eseguirne il commit. Al riavvio, IBM MQ attende le informazioni dal coordinatore prima di elaborare questa unità di recupero.

Nel commit
Il gestore code è terminato dopo che ha iniziato la propria elaborazione fase 2 (periodo d); apporta modifiche di cui è stato eseguito il commit.

In backout
Il gestore code è terminato dopo il ripristino di un'unità di ripristino, ma prima del completamento del processo (non mostrato nella figura) durante il riavvio, IBM MQ continua a ripristinare le modifiche.

z/OS Cosa accade durante la terminazione in IBM MQ for z/OS

Un gestore code termina normalmente in risposta al comando STOP QMGR. Se un gestore code viene arrestato per qualsiasi altro motivo, la chiusura è anomala.

Si noti che durante la chiusura del gestore code, IBM MQ immette internamente il comando

```
DISPLAY CONN(*) TYPE(CONN) ALL WHERE (APPLTYPE NE SYSTEMAL)
```

in modo da essere consapevoli dei thread che potrebbero impedire al gestore code di completare l'arresto.

SYSTEMAL corrisponde a APPLTYPES di SYSTEM o CHINIT, quindi il comando DISPLAY CONN che filtra i tipi di applicazione non corrispondenti a SYSTEMAL, ritorna alle informazioni della registrazione lavori relative ai sottoprocessi che potrebbero impedire la normale chiusura.

Terminazione normale

In una terminazione normale, IBM MQ arresta tutte le attività in modo ordinato. È possibile arrestare IBM MQ utilizzando la modalità quiesce, force o restart. Gli effetti sono riportati in Tabella 23 a pagina 263.

Tabella 23. Terminazione utilizzando QUIESCE, FORCE e RESTART

Tipo thread	**QUIESCE**	**Forza**	**RIAVVIA**
Thread attivi	Esegui fino al completamento	Backout	Backout
Nuovi thread	Può iniziare	Non consentito	Non consentito
Nuove connessioni	Non consentito	Non consentito	Non consentito

Le applicazioni batch vengono notificate se si verifica una chiusura mentre l'applicazione è ancora connessa.

Con CICS, un thread corrente viene eseguito solo alla fine dell'unità di ripristino. Con CICS, l'arresto di un gestore code in modalità di sospensione arresta l'adattatore CICS e, quindi, se un'attività attiva contiene più di un'unità di recupero, l'attività non viene necessariamente eseguita fino al completamento.

Se si arresta un gestore code in modalità di riavvio o di forzatura, non viene assegnato alcun nuovo thread e viene eseguito il rollback delle operazioni sui thread connessi. L'uso di queste modalità può creare unità di ripristino in dubbio per i thread che si trovano tra le fasi di elaborazione del commit. Vengono risolti quando IBM MQ viene riconnesso al sottosistema di controllo CICS, IMSo RRS.

Quando si arresta un gestore code, in qualsiasi modalità, i passi sono:

1. I collegamenti sono terminati.
2. IBM MQ smette di accettare comandi.
3. IBM MQ garantisce che tutti gli aggiornamenti in sospeso alle serie di pagine siano completati.
4. Il comando DISPLAY USAGE viene emesso internamente da IBM MQ in modo che l'RBA di riavvio venga registrato sul log della console z/OS .
5. Il punto di controllo di chiusura viene eseguito e BSDS viene aggiornato.

Le terminazioni che specificano la modalità di sospensione non influenzano le unità di ripristino in dubbio. Qualsiasi unità in dubbio rimane in dubbio.

Interruzione anomala

Una terminazione anomala può lasciare i dati in uno stato incongruente, ad esempio:

- Un'unità di ripristino è stata interrotta prima di raggiungere un punto di congruenza.
- I dati di cui è stato eseguito il commit non sono stati scritti sulle serie di pagine.
- I dati di cui non è stato eseguito il commit sono stati scritti in serie di pagine.

- Un programma applicativo è stato interrotto tra la fase 1 e la fase 2 del processo di commit, lasciando l'unità di ripristino in dubbio.

IBM MQ risolve eventuali incongruenze di dati derivanti da una terminazione anomala durante il riavvio e il recupero.

z/OS Cosa accade durante il riavvio e il ripristino in IBM MQ for z/OS

IBM MQ utilizza il log di recupero e il dataset di bootstrap (BSDS) per determinare cosa ripristinare al riavvio. Il BSDS identifica i dataset di log attivi e di archivio e l'ubicazione del punto di controllo IBM MQ più recente nel log.

Introduzione al riavvio e al ripristino

Dopo che IBM MQ è stato inizializzato, il processo di riavvio del gestore code avviene nel modo seguente:

- Inizializzazione log
- Ricreazione stato corrente
- Recupero log di inoltro
- Ripristino log all'indietro
- Rigenerazione indice coda

Quando il ripristino è stato completato:

- Le modifiche sottoposte a commit si riflettono nei dati.
- L'attività in dubbio si riflette nei dati. Tuttavia, i dati sono bloccati e non possono essere utilizzati fino a quando IBM MQ non riconosce e agisce sulla decisione in dubbio.
- Le modifiche in corso interrotte e interrotte sono state rimosse dalle code. I messaggi sono congruenti e possono essere utilizzati.
- È stato eseguito un nuovo punto di controllo.
- Sono stati creati nuovi indici per le code indicizzate contenenti messaggi persistenti (descritti in “Ricreazione degli indici della coda” a pagina 265).

Se sono in uso BSDS doppi, IBM MQ verifica la congruenza delle date / ore in BSDS:

- Se entrambe le copie di BSDS sono correnti, IBM MQ verifica se le due date / ore sono uguali. In caso contrario, IBM MQ emette il messaggio CSQJ120E e termina. Ciò può verificarsi quando le due copie di BSDS sono conservate su volumi DASD separati e uno dei volumi è stato ripristinato mentre il gestore code era arrestato. IBM MQ rileva la situazione al riavvio.
- Se è stata annullata l'allocazione di una copia di BSDS e la registrazione è continuata con un singolo BSDS, potrebbe verificarsi un problema. Se entrambe le copie di BSDS vengono conservate su un singolo volume e il volume è stato ripristinato o se entrambe le copie BSDS sono state ripristinate separatamente, IBM MQ potrebbe non rilevare il ripristino. In tal caso, i record di log non indicati in BSDS sarebbero sconosciuti al sistema.

Le applicazioni batch non vengono notificate quando si verifica un riavvio *dopo* che l'applicazione ha richiesto una connessione.

Comprensione dell'intervallo di log richiesto per il ripristino

Durante il riavvio, l'intervallo di dati di log che devono essere letti dipende da molti fattori:

- Al momento di una terminazione anomala, ci sono in genere molte unità di lavoro incomplete nel sistema. Come descritto in precedenza, il riavvio dell'elaborazione porterà il sistema ad uno stato di coerenza, che potrebbe implicare il ripristino delle unità di lavoro in corso o il ripristino dei blocchi sulle unità di lavoro in dubbio. Il ripristino dell'unità di lavoro richiede che tutti i record del log dell'unità di lavoro per le unità di lavoro in entrata, in backout e in dubbio siano disponibili. IBM MQ 'triturerà' le vecchie unità di lavoro, in modo che il ripristino dell'unità di lavoro possa essere eseguito utilizzando una gamma molto più piccola di dati di log.

- Al momento di una chiusura anomala, di solito ci sono molti aggiornamenti persistenti che vengono conservati solo nella cache del bufferpool. Non sono ancora stati scritti su disco. Queste modifiche devono essere lette dal log e riapplicate ai dati contenuti nelle serie di pagine. Gli RBA di ripristino della serie di pagine nel punto di controllo descrivono l'RBA di log più basso richiesto per l'aggiornamento delle serie di pagine ad uno stato congruente.
- Se le vecchie serie di pagine sono state introdotte nel sistema, ad esempio, è stato introdotto un backup della serie di pagine per eseguire il ripristino da un errore del supporto, tutte le modifiche devono essere lette dal log dal momento in cui è stato eseguito il backup. Queste modifiche vengono riapplicate ai dati contenuti nella serie di pagine in fase di recupero. Gli RBA di ripristino della serie di pagine contenuti nella pagina 0 della serie di pagine descrivono l'RBA di log più basso richiesto per il ripristino del supporto di una serie di pagine.
- Se si utilizzano i messaggi persistenti sulle code condivise, è necessario un intervallo di dati di log per ripristinare i CFSTRUCTs che contengono messaggi persistenti. I primi dati di log che sarebbero necessari per eseguire un ripristino CFSTRUCT, risalgono all'incirca all'ora del vecchio CFSTRUCT BACKUP.

Durante l'esecuzione normale, il comando DISPLAY USAGE TYPE (DATASET) può essere utilizzato per visualizzare l'intervallo di log di ripristino associato a questi fattori (non è in grado di fornire informazioni a causa della reintroduzione delle vecchie serie di pagine, ovviamente). Per evitare eventuali problemi che potrebbero prolungare il riavvio di un gestore code in caso di chiusura anomala, monitorare regolarmente l'output dei valori da DISPLAY USAGE TYPE (DATASET).

Inoltre, il gestore code emette messaggi informativi relativi a questi fattori:

- CSQJ160I e CSQJ161I avvertono di unità di lavoro di lunga durata.
- CSQR026I e CSQR027I forniscono informazioni che indicano se queste unità di lavoro di lunga durata sono state deviate correttamente.
- CSQE040I e CSQE041E avvertono che i backup della struttura stanno diventando obsoleti e, di conseguenza, un'operazione RECOVER CFSTRUCT richiede molto tempo.

Determinazione di quale applicazione ha un'unità di lavoro di lunga durata

È possibile determinare l'applicazione con l'unità di lavoro di lunga durata. A tale scopo, utilizzare il comando DISPLAY CONN.

Il comando DISPLAY CONN restituisce le informazioni di connessione per tutte le applicazioni connesse al gestore code, insieme alle informazioni aggiuntive che consentono di determinare quali applicazioni dispongono attualmente di un'unità di lavoro di lunga durata. Le informazioni restituite dal comando DISPLAY CONN sono simili a quelle restituite dal comando DISPLAY QSTATUS, ma la differenza principale è che DISPLAY CONN visualizza le informazioni sugli oggetti e le informazioni transazionali per una connessione particolare, piuttosto che i dettagli delle connessioni associate a un oggetto particolare.

Per ogni applicazione connessa, il comando DISPLAY CONN restituisce le seguenti informazioni:

- Informazioni di base, inclusi ID connessione e PID.
- Le informazioni transazionali per tale connessione, inclusa l'ora e la data in cui è stata creata la transazione (ovvero, quando è stato eseguito il primo MQGET/PUT nel punto di sincronizzazione) e quando la transazione ha scritto per la prima volta nel log.
- Registrare le informazioni di tempo che indicano quale applicazione ha ancora un'unità di lavoro di lunga durata.
- Un elenco di tutti gli oggetti attualmente aperti dalla connessione. I dettagli per ogni oggetto vengono restituiti come un messaggio separato, con l'ID connessione utilizzato come chiave. Poiché esistono diversi tipi di oggetto, come code e gestori code, le informazioni visualizzate con l'oggetto sono specifiche per il tipo di oggetto.

Ricreazione degli indici della coda

Per incrementare la velocità delle operazioni MQGET su una coda in cui i messaggi non vengono richiamati in modo sequenziale, è possibile specificare che si desidera che IBM MQ mantenga un indice del messaggio o degli identificativi di correlazione o del gruppo per tutti i messaggi su tale coda.

Quando un gestore code viene riavviato, questi indici vengono ricreati per ogni coda. Ciò si applica solo ai messaggi persistenti; i messaggi non persistenti vengono eliminati al riavvio. Se le code indicizzate contengono un numero elevato di messaggi persistenti, ciò aumenta il tempo impiegato per riavviare il gestore code.

È possibile scegliere di ricreare gli indici in modo asincrono all'avvio del gestore code utilizzando il parametro QINDXBLD della macro CSQ6SYSP . Se si imposta QINDXBLD=NOWAIT, IBM MQ viene riavviato senza attendere la rigenerazione degli indici.

z/OS Come vengono risolte le unità di ripristino in dubbio

Se IBM MQ perde la connessione ad un altro gestore risorse, in genere tenta di ripristinare tutti gli oggetti incongruenti al riavvio.

Se IBM MQ perde la connessione a CICS, IMSo RRS, normalmente tenta di ripristinare tutti gli oggetti incongruenti al riavvio. Le informazioni richieste per risolvere le unità di recupero in dubbio devono provenire dal sistema di coordinamento. Le sezioni successive descrivono il processo di risoluzione per ambienti differenti.

- Modalità di risoluzione delle unità di ripristino in dubbio da CICS
- Modalità di risoluzione delle unità di ripristino in dubbio da IMS
- Modalità di risoluzione delle unità di ripristino in dubbio da RRS
- Come vengono risolte le unità di recupero in dubbio con un'unità di recupero GROUP

Come vengono risolte le unità di ripristino in dubbio da CICS

In alcune circostanze, CICS non può eseguire il processo IBM MQ per risolvere le unità di ripristino in dubbio. Quando ciò si verifica, IBM MQ invia uno dei seguenti messaggi:

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E

seguiti dal messaggio CSQC408I.

Per i dettagli sul significato di questi messaggi, consultare il manuale IBM MQ for z/OS , completamento, e codici di errore .

La risoluzione delle unità in dubbio non influisce sulle risorse CICS . CICS controlla il coordinamento del ripristino e, quando viene riavviato, esegue automaticamente il commit o il backout di ciascuna unità, a seconda che vi sia un record di log che contrassegna l'inizio del commit. L'esistenza di oggetti in dubbio non blocca le risorse CICS durante la ricollegazione di IBM MQ .

Una delle funzioni dell'adattatore CICS consiste nel mantenere i dati sincronizzati tra CICS e IBM MQ. Se un gestore code termina in modo anomalo durante la connessione a CICS, è possibile che CICS esegua il commit o il backout del lavoro senza che IBM MQ ne sia a conoscenza. Quando il gestore code viene riavviato, tale lavoro viene definito *in dubbio*.

IBM MQ non può risolvere queste unità di ripristino in dubbio (ovvero, eseguire il commit o il backout delle modifiche apportate alle risorse IBM MQ) fino a quando la connessione a CICS non viene riavviata o riconnessa.

Un processo per risolvere le unità di ripristino in dubbio viene avviato durante l'avvio dell'adattatore CICS . Il processo inizia quando l'adattatore richiede un elenco di unità di ripristino in dubbio. Quindi:

- L'adattatore riceve un elenco di unità di ripristino in dubbio per questo ID connessione da IBM MQe le trasmette a CICS per la risoluzione.

- CICS confronta le voci di questo elenco con le voci del proprio log. CICS determina dal proprio elenco quale azione ha intrapreso per ogni unità di ripristino in dubbio.

Per tutte le unità risolte, IBM MQ aggiorna le code come necessario e rilascia i blocchi corrispondenti. Le unità non risolte possono rimanere dopo il riavvio. Risolverli con i metodi descritti in Amministrazione IBM MQ for z/OS.

Come vengono risolte le unità di ripristino in dubbio da IMS

La risoluzione delle unità di ripristino in dubbio in IMSnon influisce sulle risorse DL/I. IMS ha il controllo della coordinazione del ripristino e, quando viene riavviato, esegue automaticamente il commit o il backout del lavoro DL/I incompleto. La decisione di eseguire il commit o il backout per le regioni in linea (non - fast - path) è sulla presenza o assenza dei tipi di record di log IMS X'3730 'e X'3801'. L'esistenza di unità di ripristino in dubbio non implica che i record DL/I siano bloccati fino a quando IBM MQ non si connette.

Durante il riavvio del gestore code, IBM MQ crea un elenco di unità di recupero in dubbio. IMS crea un proprio elenco di RRE (Residue Recovery Entries). Gli RRE vengono registrati nei checkpoint IMS fino a quando non vengono risolte tutte le voci.

Durante la riconnessione di una regione IMS a IBM MQ, IMS indica a IBM MQ se eseguire il commit o il backout delle unità di lavoro contrassegnate in IBM MQ come in dubbio.

Quando le unità in dubbio vengono risolte:

1. Se IBM MQ riconosce di aver contrassegnato una voce per il commit e IMS l'ha contrassegnata per il backout, IBM MQ emette il messaggio CSQQ010E. IBM MQ emette questo messaggio per tutte le incongruenze di questo tipo tra IBM MQ e IMS.
2. Se IBM MQ dispone di unità in dubbio rimanenti, l'adattatore emette il messaggio CSQQ008I.

Per tutte le unità risolte, IBM MQ aggiorna le code come necessario e rilascia i blocchi corrispondenti.

IBM MQ mantiene i blocchi sul lavoro in dubbio che non è stato risolto. Ciò può causare un backlog nel sistema se vengono mantenuti blocchi importanti. La connessione rimane attiva in modo da poter risolvere gli RRE IMS . Ripristinare i thread in dubbio utilizzando i metodi descritti in Amministrazione IBM MQ for z/OS.

Tutto il lavoro in dubbio deve essere risolto a meno che non vi siano problemi software o operativi, ad esempio con un avvio a freddo IMS . La risoluzione in dubbio da parte della regione di controllo IMS avviene in due circostanze:

1. All'inizio della connessione a IBM MQ, durante la quale la risoluzione viene effettuata in modo sincrono.
2. Quando un programma termina in modo anomalo, durante il quale la risoluzione viene eseguita in modo asincrono.

Come le unità di ripristino in dubbio vengono risolte da RRS

Una delle funzioni dell'adattatore RRS consiste nel mantenere sincronizzati i dati tra IBM MQ e altri gestori risorse partecipanti RRS. Se si verifica un errore quando IBM MQ ha completato la fase uno del commit ed è in attesa di una decisione da parte di RRS (il coordinatore del commit), l'unità di recupero entra nello stato in dubbio.

Quando la comunicazione viene ristabilita tra RRS e IBM MQ, RRS esegue automaticamente il commit o il backout di ogni unità di recupero, a seconda se c'era un record di log che contrassegnava l'inizio del commit. IBM MQ non è in grado di risolvere queste unità di ripristino in dubbio (ovvero, eseguire il commit o il backout delle modifiche apportate alle risorse IBM MQ) fino a quando non viene ristabilita la connessione a RRS.

In alcune circostanze, RRS non può risolvere le unità di ripristino in dubbio. Quando ciò accade, IBM MQ invia uno dei seguenti messaggi alla console z/OS :

- CSQ3011I
- CSQ3013I
- CSQ3014I
- CSQ3016I

Per i dettagli sul significato di questi messaggi, consultare il manuale IBM MQ for z/OS , completamento, e codici di errore .

Per tutte le unità di ripristino risolte, IBM MQ aggiorna le code come necessario e rilascia i blocchi corrispondenti. Le unità di ripristino non risolte possono rimanere dopo il riavvio. Risolverli con il metodo descritto in Amministrazione IBM MQ for z/OS.

Come vengono risolte le unità di recupero in dubbio con un'unità di recupero GROUP

Le transazioni in dubbio che hanno una disposizione GROUP Unit of Recovery possono essere risolte dal coordinatore della transazione da qualsiasi gestore code nel QSG (queue sharing group) in cui è abilitato l'attributo GROUPUR del gestore code. Ogni volta che un coordinatore di transazioni si riconnette, di solito richiede un elenco di transazioni in dubbio in sospeso e le risolve utilizzando le informazioni dai relativi log.

Quando un coordinatore della transazione, che si è collegato a un'unità di disposizione di recupero GROUP, richiede l'elenco di transazioni in dubbio, l'elenco restituito comprende tutte le transazioni in dubbio con un'unità di disposizione di recupero GROUP che esistono in tutto il gruppo di condivisione code. Questo elenco non dipende da quale gestore code sono state avviate le transazioni in dubbio. Un gestore code che elabora una richiesta di questo tipo compila l'elenco comunicando con tutti gli altri gestori code attivi nel gruppo di condivisione code utilizzando il SISTEMA SYSTEM.QSG.UR.RESOLUTION.QUEUE. Il gestore code legge quindi i log di qualsiasi gestore code inattivo, dall'ultimo punto di controllo, per identificare eventuali ulteriori transazioni in dubbio che avrebbero riportato se fossero state attive.

Quando un coordinatore della transazione richiede la risoluzione di una transazione in dubbio, il gestore code a cui è connesso identifica se la transazione è stata originata su se stessa e, in tal caso, la risolve nello stesso modo delle transazioni con un'unità QMGR di disposizione del ripristino. Se la transazione è stata originata su un altro gestore code attivo nel QSG, una richiesta di completamento della risoluzione viene instradata a tale gestore code utilizzando il SISTEMA SYSTEM.QSG.UR.RESOLUTION.QUEUE. Nel caso in cui la transazione sia stata originata su un gestore code inattivo nel QSG, qualsiasi lavoro della coda condivisa viene risolto immediatamente e una richiesta di risoluzione di qualsiasi lavoro della coda privata rimanente viene inserita nel SISTEMA SYSTEM.QSG.UR.RESOLUTION.QUEUE. Il gestore code inattivo elabora questa richiesta all'avvio prima di accettare nuovo lavoro. In questo scenario, i log del gestore code originale riflettono ancora che l'unità di recupero è in dubbio fino a quando non viene riavviata ed elaborata la richiesta.

z/OS Ripristino coda condivisa

Utilizzare questo argomento per comprendere il ripristino e la resilienza IBM MQ di vari componenti nell'ambiente del gruppo di condivisione code.

- “Gestione della resilienza per gli errori di connettività CF (Coupling Facility)” a pagina 272
- “Funzionamento operativo” a pagina 275

Ripristino transazionale

Quando un'applicazione emette una chiamata MQBACK o termina in modo anomalo (ad esempio, a causa di un EXEC CICS ROLLBACK o di un'interruzione IMS), le informazioni a livello di thread memorizzate nel gestore code garantiscono che venga eseguito il rollback dell'unità di lavoro incompleta. Le operazioni MQPUT e MQGET all'interno del punto di sincronizzazione sulle code condivise vengono sottoposte a rollback allo stesso modo degli aggiornamenti alle code non condivise.

Ripristino peer

Se un gestore code ha esito negativo, si disconnette in modo anomalo dalle strutture CFS a cui è attualmente connesso. Se la connessione tra l'istanza z/OS e la CF (Coupling Facility) ha esito negativo (ad esempio, un errore di collegamento fisico o lo spegnimento di una CF o di una partizione), ciò viene rilevato anche come una terminazione anomala della connessione tra il gestore code e le strutture CF coinvolte. Altri gestori code nello stesso gruppo di condivisione code che rimangono connessi a tale struttura rilevano la disconnessione anomala e tentano tutti di avviare il *ripristino peer* per il gestore code in errore su tale struttura. Solo uno di questi gestori code avvia correttamente il recupero peer, ma tutti gli altri gestori code cooperano nel recupero delle unità di lavoro che erano di proprietà del gestore code in errore.

Se un gestore code non riesce quando non ci sono peer connessi a una struttura, il ripristino viene eseguito quando un altro gestore code si connette a tale struttura o quando il gestore code che ha avuto esito negativo viene riavviato.

Il ripristino peer, spesso indicato come Peer Level Recovery (PLR), viene eseguito in base alla struttura ed è possibile che un singolo gestore code partecipi al recupero di più di una struttura contemporaneamente. Tuttavia, la serie di peer che cooperano nel ripristino di strutture differenti potrebbe variare in base a quali gestori code sono stati connessi alle diverse strutture al momento dell'errore.

Quando il gestore code in errore viene riavviato, si riconnette alle strutture a cui era connesso al momento dell'errore e recupera tutte le unità di lavoro non risolte rimanenti che non sono state ripristinate dal ripristino peer.

Il ripristino peer è un processo a più fasi. Durante la prima fase, vengono recuperate le unità di lavoro che avevano superato la fase in corso; ciò potrebbe comportare il commit dei messaggi per le unità di lavoro che sono in fase di commit e il blocco dei messaggi per le unità di lavoro che sono in dubbio. Durante la seconda fase, vengono controllate le code con thread attivi nel gestore code in errore, viene eseguito il rollback dei messaggi non sottoposti a commit relativi alle unità di lavoro in corso e vengono reimpostati gli handle attivi sulle code condivise nel gestore code in errore. Ciò significa che IBM MQ reimposta tutti gli indicatori che indicano che il gestore code in errore aveva una coda condivisa aperta per l'immissione esclusiva, consentendo ad altri gestori code attivi di aprire la coda per l'immissione.

Definizioni di code condivise

Gli oggetti coda che rappresentano gli attributi di una coda condivisa vengono conservati nel repository Db2 condiviso utilizzato dal gruppo di condivisione code. Accertarsi che siano in atto procedure adeguate per il backup e il recupero delle tabelle Db2 utilizzate per conservare gli oggetti IBM MQ . È inoltre possibile utilizzare il programma di utilità IBM MQ CSQUTIL per creare comandi MQSC da ripetere in un gestore code per ridefinire gli oggetti IBM MQ , incluse le definizioni di gruppi e code condivise memorizzate in Db2.

Registrazione

I gruppi di condivisione code possono supportare messaggi persistenti, poiché i messaggi sulle code condivise possono essere registrati nei log del gestore code.

Errori della struttura e della CF (Coupling Facility)

Ci sono due tipi di errore che possono essere riportati per una struttura CF (coupling facility): errore della struttura e perdita di connettività. I servizi Sysplex per la condivisione dei dati (XES) informano IBM MQ di un malfunzionamento della struttura CF o di un errore CF con un evento di errore della struttura. Se XES crea un evento di perdita di connettività, ciò non indica necessariamente che si è verificato un problema con la struttura, è possibile che non vi sia alcuna connessione disponibile per comunicare con la struttura. È possibile che non tutti i gestori code ricevano una perdita di evento di connettività per la struttura; dipende dalla configurazione delle connessioni alla CF. Una perdita di evento di connettività può essere ricevuta anche a causa dei comandi dell'operatore, ad esempio VARY PATH OFFLINE o CONFIG CHP OFFLINE.

Le strutture CF utilizzate da IBM MQ possono essere configurate per utilizzare il duplex gestito dal sistema. Ciò significa che se si verifica un singolo errore, l'elaborazione del failover gestito dal sistema nasconde l'errore di una struttura o la perdita di connettività e il gestore code non è informato dell'errore. Se si verifica un errore di entrambe le istanze di una struttura duplex o di una connessione, il gestore code riceve l'evento appropriato e lo gestisce allo stesso modo di un evento di errore per una struttura simplex. I dettagli sul modo in cui il gestore code gestisce gli eventi sono descritti in Scenari.

Nell'improbabile caso di errore di una CF o di una struttura, tutti i messaggi non persistenti memorizzati nelle strutture dell'applicazione interessate vengono persi. È possibile ripristinare i messaggi persistenti utilizzando il comando RECOVER CFSTRUCT. Se una struttura dell'applicazione ripristinabile ha avuto esito negativo, qualsiasi ulteriore attività dell'applicazione a questa struttura viene impedita fino a quando la struttura non viene ripristinata.

Per assicurarsi che sia possibile ripristinare una struttura CF in un periodo di tempo ragionevole, eseguire backup frequenti utilizzando il comando BACKUP CFSTRUCT. È possibile scegliere di eseguire i backup su qualsiasi gestore code del gruppo di condivisione code oppure dedicare un gestore code per eseguire tutti i backup. Automatizzare il processo di esecuzione dei backup per garantire che vengano eseguiti regolarmente.

Ogni backup viene scritto nel dataset del log attivo del gestore code che esegue il backup. Il repository della coda condivisa Db2 registra il nome della struttura CF di cui si sta eseguendo il backup, il nome del gestore code che sta eseguendo il backup, l'intervallo RBA per questo backup nel log del gestore code e l'ora di backup.

La struttura di gestione contiene informazioni relative alle unità di lavoro incomplete sulle code condivise al momento dell'errore della struttura dell'applicazione, pertanto la struttura di gestione deve essere disponibile durante l'elaborazione di RECOVER CFSTRUCT. Se la struttura di gestione ha avuto esito negativo, tutti i gestori code nel gruppo di condivisione code devono aver ricreato le voci della struttura di gestione prima di poter immettere il comando RECOVER CFSTRUCT.

I gestori code ricreano le voci della struttura di gestione automaticamente e senza terminare. Se un gestore code non è in esecuzione al momento dell'errore, le relative voci della struttura di amministrazione possono essere ricreate da un altro gestore code nel gruppo di condivisione code in esecuzione allo stesso livello o ad un livello superiore.

Per ripristinare una struttura dell'applicazione, immettere un comando RECOVER CFSTRUCT sul gestore code per cui si desidera eseguire il ripristino. È possibile recuperare una singola struttura CF o è possibile recuperare più strutture CF contemporaneamente. È possibile eseguire il ripristino utilizzando qualsiasi gestore code nel gruppo di condivisione code, non è necessario che sia quello che ha eseguito il backup o che sia stato precedentemente connesso alla struttura non riuscita.

Il comando RECOVER CFSTRUCT utilizza il backup, che si trova attraverso le informazioni del repository Db2 (Db2 deve quindi essere disponibile sul gestore code in cui viene eseguito il ripristino) e lo recupera fino al punto in cui si è verificato l'errore.

Il comando RECOVER CFSTRUCT esegue questa operazione applicando i record di log da ogni gestore code nel gruppo di condivisione code che ha eseguito MQPUT o MQGET tra l'inizio del backup e l'ora dell'errore, a qualsiasi coda condivisa associata alla struttura CF. L'unione risultante dei log potrebbe richiedere la lettura di una quantità considerevole di dati di log poiché tutti i dati di log scritti dai gestori code partecipanti a partire dalla lettura del backup. Si consiglia vivamente di eseguire backup frequenti (ad esempio, ogni ora), soprattutto se ci sono messaggi di grandi dimensioni all'interno del backup.

Scenari di errore della struttura

Scenari

Se viene riportato un errore per una struttura CF, l'azione eseguita dai gestori code connessi dipende da quanto segue:

- Il tipo di errore riportato dal componente XES di z/OS a IBM MQ.
- Il tipo di struttura (applicazione o amministrazione)
- Il livello del gestore code
- Il CFLEVEL dell'oggetto IBM MQ CFSTRUCT (2, 3, 4 o 5. Questo non è il CFLEVEL del microcodice CFCC)
- L'attributo RECAUTO di un oggetto IBM MQ CFSTRUCT in CFLEVEL (5)

I seguenti scenari descrivono cosa accade quando viene riportato un errore per la struttura di gestione:

- Se si riceve un evento di errore della struttura per la struttura di amministrazione, la struttura viene riassegnata e rigenerata automaticamente senza che il gestore code venga terminato. Una nuova istanza della struttura viene assegnata da XES quando un gestore code tenta di connettersi ad essa.

  Quando il gestore code è connesso alla nuova istanza della struttura, scrive le voci nella struttura. Questa elaborazione viene eseguita dal gestore code e non fa parte dell'elaborazione di rigenerazione XES.

  Se un gestore code non era in esecuzione al momento dell'errore o viene terminato prima che sia stato completato il ripristino della sua parte della struttura di amministrazione, le voci della struttura di amministrazione possono essere ricreate da un altro gestore code nel gruppo di condivisione code.

  Le voci della struttura di gestione di un gestore code possono essere ricreate solo da un altro gestore code in esecuzione allo stesso livello o ad un livello superiore. Se le voci della struttura di gestione di un gestore code non possono essere ricreate da un altro gestore code nel gruppo di condivisione code, riavviare il gestore code in modo che possa completare la rigenerazione della sua parte della struttura.

  Alcune azioni vengono sospese fino a quando le voci della struttura di amministrazione per tutti i gestori code non vengono ricreate. Le azioni sospese includono:

  – Apertura e chiusura di code condivise.
  – Esecuzione del commit o del backout delle unità di ripristino.
  – Applicazioni serializzate che si collegano o si disconnettono dal gestore code.
  – Backup o ripristino di una struttura dell'applicazione.

  Tutte le applicazioni serializzate già connesse al gestore code possono continuare l'elaborazione. Tutte le applicazioni serializzate che tentano di connettersi con i parametri MQCNO_SERIALIZE_CONN_TAG_QSG o MQCNO_RESTRICT_CONN_TAG_QSG ricevono il codice di ritorno MQRC_CONN_TAG_NOT_USABLE.

  Quando le voci della struttura di amministrazione per il gestore code sono state ricreate, le azioni sospese vengono riprese.

I seguenti scenari descrivono cosa accade quando viene riportato un errore per una struttura dell'applicazione:

- Se si riceve un evento di errore della struttura per una struttura dell'applicazione e CFLEVEL è 1 o 2, il gestore code viene terminato. Riavviare il gestore code. Il primo gestore code che tenta di connettersi nuovamente alla struttura fa sì che XES assegni una nuova istanza della struttura.
- Se si riceve un evento di errore della struttura per una struttura dell'applicazione e CFLEVEL è 3, 4 o 5, i gestori code connessi alla struttura continuano l'esecuzione. Le applicazioni che non utilizzano le code nella struttura in errore possono continuare la normale elaborazione.

  Tuttavia, le applicazioni che tentano le operazioni sulle code nella struttura non riuscita ricevono un errore MQRC_CF_STRUC_FAILED fino a quando la struttura non viene ricreata correttamente, a quel punto l'applicazione può aprire nuovamente le code.

  La ricostruzione della struttura viene avviata automaticamente per le strutture dell'applicazione CFLEVEL (5) definite con RECAUTO (YES). In caso contrario, la struttura verrà ricreata quando viene immesso il comando RECOVER CFSTRUCT.

Resilienza a errori di connettività CF (Coupling Facility)

Cos' è la resilienza agli errori di connettività della CF?

La resilienza agli errori di connettività della CF (Coupling Facility) fa riferimento alla capacità dei gestori code in un gruppo di condivisione code di tollerare la perdita di connettività a una struttura CF (Coupling Facility) senza terminazione. Questa funzione tenta anche di ricreare la struttura in un'altra CF (Coupling Facility) con una migliore connettività in modo da riottenere l'accesso alle code condivise il più presto possibile.

Cos' è la perdita parziale di connettività?

IBM MQ definisce la perdita parziale della connettività come una situazione in cui uno o più sistemi nel sysplex perdono la connettività alla CF (Coupling Facility) in cui è allocata la struttura a cui accede il sistema, ma almeno un sistema nel sysplex mantiene la connettività alla stessa CF.

Che cos' è la perdita totale di connettività?

IBM MQ definisce una perdita totale di connettività come una situazione in cui nessun sistema nel sysplex dispone di connettività alla CF e la struttura allocata al suo interno.

Perché abilitare questa funzione?

La resilienza agli errori di connettività della CF (Coupling Facility) migliora la disponibilità di IBM MQ, consentendo alle code non condivise di rimanere disponibili dopo che un gestore code ha perso la connettività a una o più strutture CFS. Inoltre, i gestori code che perdono la connettività a una struttura CFS tentano automaticamente di ricreare la struttura in un'altra CFS disponibile, migliorando la disponibilità delle code condivise all'interno del gruppo di condivisione code.

Considerazioni sull'abilitazione di questa funzione

Un gestore code che tollera la perdita di connettività alle strutture CFS (coupling facility structure) senza terminazione potrebbe non essere in grado di riconnettersi a una struttura CFS per un certo periodo di tempo se non è disponibile alcuna CFS alternativa. Le code condivise definite su una struttura che ha subito una perdita di connettività rimangono non disponibili fino a quando non viene ripristinata la connessione alla struttura. In questa situazione, le applicazioni che si collegano ai membri del gruppo di condivisione code per eseguire il lavoro della coda condivisa potrebbero scoprire che le code condivise a cui devono accedere non sono disponibili. Per evitare questa situazione, si consiglia di configurare i gestori code in modo che terminino quando si perde la connettività a una struttura CF (Coupling Facility). Questa terminazione forza le applicazioni a collegarsi ad un altro membro del gruppo di condivisione code che ha la connettività alle strutture CF (Coupling Facility) in cui sono definite le code condivise richieste dall'applicazione.

Gestione della resilienza per gli errori di connettività CF (Coupling Facility)

Come posso abilitare questa funzionalità?

È necessario eseguire le seguenti operazioni per abilitare la resilienza alla connettività CF (Coupling Facility)

1. Verificare che il dataset di coppia CFRM sia stato formattato per supportare la rigenerazione gestita dal sistema. Ciò consente ai gestori code di avviare una rigenerazione gestita dal sistema per creare nuovamente una struttura in una CF (coupling facility) disponibile. Utilizzare il comando **DISPLAY XCF,COUPLE,TYPE=CFRM** per determinare il formato del dataset di coppia CFRM. Per supportare la rigenerazione gestita dal sistema, il dataset di coppia CFRM deve essere formattato specificando:

   ```
   "ITEM NAME(SMREBLD) NUMBER(1)"
   ```

   Fare riferimento alla documentazione di z/OS MVS Setting Up a Sysplex per ulteriori informazioni sulla formattazione di una coppia di dati CFRM.
2. Assicurarsi che sia disponibile una CF alternativa e che sia presente nell'elenco delle preferenze CFRM per tutte le strutture CF (Coupling Facility) IBM MQ . Ciò consente ai gestori code di tentare di ricreare le strutture in una CF (Coupling Facility) alternativa disponibile per ripristinare l'accesso alle strutture il più presto possibile.

   Le strutture IBM MQ devono essere definite con ENFORCEORDER (NO) nella politica CFRM, in modo che XCF sia in grado di scegliere la CF ottimale nella configurazione se IBM MQ deve riassegnare la struttura.

   Fare riferimento alla documentazione z/OS MVS Impostazione di un Sysplex per ulteriori informazioni sugli elenchi di preferenze della struttura.
3. Modificare tutte le strutture application coupling facility che devono tollerare la perdita di connettività a CFLEVEL (5). Questo è il livello minimo che può tollerare una perdita di connessione.
4. Determinare i valori richiesti per gli attributi **QMGR CFCONLOS** e **CFSTRUCT CFCONLOS** e modificarli di conseguenza. L'attributo **QMGR CFCONLOS** controlla se la perdita di connettività per la struttura di gestione è tollerata e l'attributo **CFSTRUCT CFCONLOS** controlla se la perdita di connettività è tollerata da ciascuna struttura CFS (application coupling facility structure). Se i valori predefiniti per questi attributi vengono conservati, il gestore code viene terminato in seguito alla perdita di connettività a qualsiasi struttura CFS (coupling facility structure).
5. Determinare i valori richiesti per l'attributo **CFSTRUCT RECAUTO** per ogni struttura CFS (application coupling facility) e modificarli di conseguenza. Questo attributo controlla se le strutture CF devono essere ripristinate automaticamente utilizzando i dati registrati in seguito alla perdita totale di connettività. Se viene conservato il valore predefinito per questo attributo, non viene eseguito alcun ripristino automatico per le strutture delle applicazioni in seguito alla perdita totale di connettività.

Scenario 1 - Perdita di connettività alla struttura di gestione

I gestori code possono tollerare la perdita di connettività alla struttura di gestione senza terminare.

Quando la connettività alla struttura di gestione viene persa da qualsiasi gestore code configurato per tollerare la perdita di connettività alla struttura di gestione, tutti i membri del gruppo di condivisione code si disconnettono dalla struttura di amministrazione. Tutti i gestori code attivi nel gruppo di condivisione code tentano quindi di riconnettersi alla struttura di gestione, facendola riallocare nella CF con la migliore connettività a tutti i sistemi nel sysplex e ricreare i dati della struttura di gestione.

Nota: Questa potrebbe non essere necessariamente la CF (Coupling Facility) che ha la migliore connettività a tutti i sistemi che hanno gestori code attivi.

Se un gestore code non riesce a riconnettersi alla struttura di amministrazione, ad esempio perché nessuna delle funzioni di accoppiamento nell'elenco delle preferenze CFRM per la struttura di amministrazione è disponibile, alcune operazioni della coda condivisa restano non disponibili fino a quando il gestore code non riesce a riconnettersi correttamente alla struttura di amministrazione e a ricreare i dati della struttura di amministrazione. La riconnessione si verifica automaticamente quando una CF adatta diventa disponibile sul sistema.

La connessione alla struttura di amministrazione durante l'avvio del gestore code a causa di una mancanza di connettività alla CF o di una funzione di accoppiamento non è tollerata. Tutti i gestori code attivi nel gruppo di condivisione code tentano quindi di riconnettersi alla struttura di amministrazione, facendola riassegnare in un'altra CF, se disponibile, e ricreano i dati della struttura di amministrazione.

Scenario 2 - Perdita di connettività alla struttura dell'applicazione

La perdita di connettività alle strutture dell'applicazione **CFLEVEL(5)** o superiore può essere tollerata senza che il gestore code venga terminato. I gestori code connessi alle strutture dell'applicazione in **CFLEVEL(4)** o in una posizione inferiore o alle strutture in **CFLEVEL(5)** che non sono state configurate per tollerare la perdita di connettività, si chiudono con il codice di errore 00C510AB quando si perde la connettività alla struttura.

Quando la connettività viene persa per una struttura dell'applicazione configurata per tollerare la perdita di connettività, tutti i gestori code che hanno perso la connettività alla struttura si disconnettono. Il comportamento successivo del gestore code dipende dal fatto che la perdita di connettività sia parziale o totale.

Perdita parziale di connettività a un'applicazione

Se si determina che la perdita di connettività è parziale, i gestori code che hanno perso la connettività alla struttura tentano di avviare una rigenerazione gestita dal sistema per spostare la struttura in un'altra CF con connettività migliorata. Se questa rigenerazione ha esito positivo, i messaggi persistenti e non persistenti nella struttura vengono copiati nell'altra CF e l'accesso alle code nella struttura viene ripristinato. I gestori code che non hanno perso la connettività rimangono connessi alla struttura, tuttavia, le operazioni che accedono alla struttura vengono ritardate durante il processo di rigenerazione gestito dal sistema.

Se una struttura dell'applicazione non può essere ricreata in un'altra CF (Coupling Facility) con connettività migliorata o alcuni gestori code non hanno ancora la connettività alla struttura dopo che è stata ricreata in un'altra CF, le code definite sulla struttura rimangono non disponibili sui gestori code che non hanno la connettività alla struttura fino a quando la connettività non viene ripristinata nella CF (Coupling Facility). I gestori code si riconnettono automaticamente alla struttura quando diventa disponibile e l'accesso alle code condivise definite sulla struttura viene ripristinato.

Perdita totale di connettività a una struttura dell'applicazione

Se tutti i sistemi MVS nel sysplex hanno perso la connettività alla CF in cui è allocata la struttura dell'applicazione, z/OS annulla l'allocazione della struttura dalla CF ogni volta che viene effettuato un tentativo di riconnessione alla struttura. È possibile che il gestore code tenti di riconnettersi alla struttura per diversi motivi, ad esempio un tentativo da parte di un'applicazione di aprire una coda condivisa o una notifica dal sistema che le nuove risorse CF potrebbero essere diventate disponibili. È quindi probabile che tutti i messaggi non persistenti nella struttura interessata vadano persi in seguito alla perdita totale di connettività a una struttura dell'applicazione.

Le strutture dell'applicazione ripristinabili vengono ripristinate automaticamente in seguito alla perdita totale di connettività, se sono state definite con **RECAUTO(YES)**. Il ripristino inizia quasi immediatamente se è disponibile una funzione di accoppiamento alternativa per assegnare la struttura in, o ogni volta che tale funzione di accoppiamento diventa disponibile. Se una struttura non è stata definita con **RECAUTO(YES)**, è possibile avviare il ripristino immettendo il comando **RECOVER CFSTRUCT** . Ciò recupera tutti i messaggi persistenti nella struttura, ma tutti i messaggi non persistenti vengono persi. Poiché questo processo implica la lettura del log del gestore code, può richiedere del tempo per il completamento, pertanto si consiglia di eseguire regolarmente i backup della struttura per ridurre il tempo fino al ripristino dell'accesso alle code condivise sulla struttura.

I gestori code tentano di riconnettersi a strutture di applicazioni non ripristinabili non appena un'applicazione tenta di aprire una coda condivisa definita sulla struttura o quando viene ricevuta una notifica dal sistema che indica che le nuove risorse CF sono diventate disponibili. Se è disponibile una funzione di accoppiamento adatta per assegnare la struttura, viene assegnata una nuova struttura e viene ripristinato l'accesso alle code condivise definite sulla struttura. Poiché i messaggi persistenti non possono essere inseriti nelle code definite in strutture non recuperabili, tutti i messaggi nelle code condivise vengono persi.

Funzionamento operativo

Se un gestore code IBM WebSphere MQ 7.1, o successivo, configurato per tollerare la perdita di connettività a una particolare struttura CFS perde la connettività, i membri del gruppo di condivisione code tentano il ripristino automatico dall'errore e la riconnessione alla struttura. Questa attività potrebbe comportare la riallocazione della struttura in un'altra CF (Coupling Facility) con una migliore connettività, se disponibile. Tuttavia, l'intervento dell'operatore potrebbe essere ancora necessario per recuperare la perdita di connettività.

Generalmente, l'azione dell'operatore richiesta è:

1. Risolvere la causa dell'errore che ha causato la perdita di connettività.
2. Verificare che una CF (Coupling Facility) in cui è possibile allocare le strutture IBM MQ sia disponibile su tutti i sistemi nel sysplex

Tutte le strutture che sono state riassegnate automaticamente in un'altra CF (Coupling Facility) dopo la perdita dell'evento di connettività, possono essere spostate nella CF (Coupling Facility) con la connettività ottimale a tutti i gestori code nel gruppo di condivisione code. Se necessario, questa operazione può essere eseguita avviando il comando di rigenerazione gestito dal sistema **SETXCF START,REBUILD** come documentato in z/OS MVS System Commands Reference.

Nel caso di una perdita parziale di connettività a una struttura dell'applicazione, i gestori code che hanno perso la connettività alla struttura tentano di avviare una rigenerazione gestita dal sistema. Questo processo assegna la struttura in un'altra CF solo se tale CF ha la connettività a tutti i gestori code attivi attualmente connessi alla struttura. Pertanto, è possibile che laddove la maggior parte dei gestori code in un gruppo di condivisione code ha perso la connettività a una struttura dell'applicazione, non siano in grado di ricreare la struttura in un'altra CF a causa dei gestori code ancora connessi alla struttura originale. In questa situazione, i gestori code ancora connessi alla struttura originale possono essere arrestati per consentire la ricostruzione della struttura oppure è possibile immettere il comando **RESET CFSTRUCT ACTION(FAIL)** per far fallire la struttura. Il ripristino può essere avviato sulle strutture applicabili immettendo il comando **RECOVER CFSTRUCT** .

Nota: In caso di errore e ripristino della struttura, tutti i messaggi non persistenti sulla struttura vengono persi.

z/OS Concetti di sicurezza in IBM MQ for z/OS

Utilizzare questo argomento per comprendere l'importanza della sicurezza per IBM MQe le implicazioni di non disporre di impostazioni di sicurezza adeguate sul sistema.

Perché è necessario proteggere risorse IBM MQ

IBM MQ gestisce il trasferimento di informazioni potenzialmente prezioso. L'applicazione della sicurezza garantisce che le risorse che IBM MQ possiede e gestisce siano protette da accessi non autorizzati. Tale accesso potrebbe comportare la perdita o la divulgazione delle informazioni.

È necessario assicurarsi che nessuna delle seguenti risorse sia acceduta o modificata da qualsiasi utente o processo non autorizzato:

- Connessioni a IBM MQ
- Oggetti IBM MQ come code, processi e elenchi nomi
- IBM MQ link di trasmissione, ovvero, IBM MQ canali
- IBM MQ comandi di controllo del sistema
- IBM MQ messaggi
- Informazioni di contesto associate ai messaggi

Per fornire la sicurezza necessaria, IBM MQ utilizza SAF (system authorization facility) z/OS per instradare le richieste di autorizzazione a un ESM (External Security Manager), ad esempio Security Server (precedentemente noto come RACF). IBM MQ non esegue alcuna verifica di sicurezza. Dove vengono

utilizzati l'accodamento distribuito o i client, è possibile richiedere ulteriori misure di sicurezza, per cui IBM MQ fornisce record di autenticazione di canale, uscite di canale, l'attributo del canale MCAUSER e TLS.

La decisione di consentire l'accesso a un oggetto viene presa dal MES e IBM MQ segue tale decisione. Se l'ESM non può prendere una decisione, IBM MQ impedisce l'accesso all'oggetto.

Cosa succede se non proteggi le risorse IBM MQ

Se non si fa nulla per quanto riguarda la sicurezza, l'effetto più probabile è che *tutti* gli utenti possano accedere e modificare *ogni* risorsa. Ciò include non solo gli utenti locali, ma anche quelli sui sistemi remoti che utilizzano l'accodamento distribuito o i client, in cui i controlli di sicurezza di accesso potrebbero essere meno rigorosi di quanto avviene normalmente per z/OS.

Per abilitare il controllo di sicurezza è necessario effettuare le seguenti operazioni:

- Installare e attivare un ESM (ad esempio, Security Server).
- Definire la classe MQADMIN se si sta utilizzando un ESM diverso da Security Server.
- Attivare la classe MQADMIN.

È necessario considerare se l'utilizzo di nomi di risorse con maiuscole e minuscole potrebbe essere utile per la propria azienda. Se si utilizzano nomi di risorse con maiuscole e minuscole nei profili ESM, è necessario definire e attivare la classe MXADMIN.

z/OS Codifica dataset

DSE (Data Set Encryption) consente di codificare i dataset z/OS , in modo che i dati in essi contenuti possano essere visualizzati o modificati solo dagli ID utente a cui è stata concessa l'autorizzazione specifica. Ciò fornisce la crittografia dei dati inattivi nel file system e impedisce la divulgazione involontaria di informazioni sensibili agli utenti che hanno una legittima necessità di business e le autorizzazioni per gestire i dataset stessi.

Prima di IBM MQ for z/OS 9.1.4, IBM MQ for z/OS non supportava l'utilizzo di DSE con i log attivi, le serie di pagine e i data set di messaggi condivisi (SMDS) che forniscono i meccanismi di persistenza principali per i messaggi IBM MQ .

Al contrario, Advanced Message Security fornisce una soluzione di crittografia end - to - end per la messaggistica IBM MQ , che comprende l'intera rete IBM MQ , la crittografia dei dati in corso, inattivi e anche all'interno dei processi IBM MQ di runtime.

Altri dataset sequenziali e VSAM utilizzati in un sottosistema IBM MQ possono essere codificati utilizzando DSE. Ad esempio:

- bootstrap data set (BSDS)
- Comandi MQSC (Sequential files holding system configuration) letti all'avvio utilizzando CSQINPx DDNAMEs
- Log di archivio IBM MQ , spesso utilizzati per l'archiviazione a lungo termine dei dati di log IBM MQ per scopi di controllo.

È possibile crittografare utilizzando DSE allocando una classe di dati definita con un'etichetta chiave del dataset. Per ulteriori informazioni, consultare Pianificazione della memoria dell'archivio di log.

Da IBM MQ for z/OS 9.1.4, IBM MQ for z/OS supporta l'utilizzo di DSE con i log attivi e le serie di pagine oltre al supporto fornito nelle release precedenti.

IBM MQ for z/OS non supporta l'utilizzo di DSE per gli SMDS (shared message data set).

Consultare la sezione, Riservatezza per i dati inattivi su IBM MQ for z/OS con la crittografia del dataset. per ulteriori informazioni.

Concetti correlati

Concetti di sicurezza

Record di autenticazione di canale
Autorizzazione per gestire gli oggetti IBM MQ su z/OS
Protocolli di sicurezza crittografici: TLS

Attività correlate

Impostazione della sicurezza su z/OS
Confronto tra sicurezza a livello di collegamento e sicurezza a livello di applicazione

Riferimenti correlati

Messaggi per IBM MQ for z/OS

z/OS Controlli di sicurezza e opzioni in IBM MQ for z/OS

È possibile specificare se la sicurezza è attivata per l'intero sistema secondario IBM MQ e se si desidera eseguire controlli di sicurezza a livello di gestore code o di gruppo di condivisione code. È anche possibile controllare il numero di ID utente controllati per la sicurezza delle risorse API.

Sicurezza sottosistema

La sicurezza del sottosistema è un controllo che specifica se viene eseguito un controllo di sicurezza per l'intero gestore code. Se non si richiede un controllo di sicurezza (ad esempio, su un sistema di verifica) o se si è soddisfatti del livello di sicurezza su tutte le risorse che possono connettersi a IBM MQ (inclusi client e canali), è possibile disattivare il controllo di sicurezza per il gestore code o il gruppo di condivisione code in modo che non si verifichi un ulteriore controllo di sicurezza.

Questo è l'unico controllo che può disattivare completamente la sicurezza e determinare se vengono eseguiti o meno altri controlli di sicurezza. In altre parole, se si disattiva la verifica del gestore code o del gruppo di condivisione code, non viene eseguito alcun altro controllo IBM MQ ; se lo si lascia attivato, IBM MQ verifica i requisiti di sicurezza per altre risorse IBM MQ .

È anche possibile attivare o disattivare la sicurezza per particolari serie di risorse, come i comandi.

Controllo a livello del gestore code o del gruppo di condivisione code

È possibile implementare la sicurezza a livello di gestore code o a livello di gruppo di condivisione code. Se si implementa la sicurezza a livello di gruppo di condivisione code, tutti i gestori code del gruppo condividono gli stessi profili. Ciò significa che ci sono meno profili da definire e gestire, rendendo più semplice la gestione della sicurezza. Inoltre, semplifica l'aggiunta di un nuovo gestore code al gruppo di condivisione code poiché eredita i profili di protezione esistenti.

È anche possibile implementare una combinazione di entrambi se l'installazione lo richiede, ad esempio, durante la migrazione o se si dispone di un gestore code nel gruppo di condivisione code che richiede diversi livelli di sicurezza rispetto agli altri gestori code nel gruppo.

Sicurezza a livello di gruppo di condivisione code

Il controllo di sicurezza a livello di gruppo di condivisione code viene eseguito per l'intero gruppo di condivisione code. Consente di semplificare la gestione della protezione poiché richiede la definizione di un minor numero di profili di sicurezza. L'autorizzazione di un ID utente per utilizzare una particolare risorsa viene gestita a livello di gruppo di condivisione code ed è indipendente dal gestore code che l'ID utente sta utilizzando per accedere alla risorsa.

Ad esempio, un'applicazione server viene eseguita con l'ID utente SERVER e desidera accedere a una coda denominata SERVER.REQUESTe si desidera eseguire un'istanza di SERVER su ogni immagine z/OS nel sysplex. Piuttosto che consentire a SERVER di aprire SERVER.REQUEST su ciascun gestore code singolarmente (sicurezza a livello di gestore code), è possibile consentire l'accesso solo a livello di gruppo di condivisione code.

È possibile utilizzare i profili di sicurezza a livello di gruppo di condivisione code per proteggere tutti i tipi di risorsa, locale o condivisa.

sicurezza a livello di gestore code
È possibile utilizzare i profili di sicurezza a livello di gestore code per proteggere tutti i tipi di risorse, locali o condivise.

Combinazione di entrambi i livelli
È possibile utilizzare una combinazione di sicurezza a livello di gestore code e gruppo di condivisione code.

È possibile sovrascrivere le impostazioni di protezione a livello di gruppo di condivisione code per uno specifico gestore code membro di tale gruppo. Ciò significa che è possibile eseguire un livello diverso di controlli di sicurezza su un singolo gestore code rispetto a quelli eseguiti su altri gestori code del gruppo.

Per ulteriori informazioni, consultare Profili per controllare la sicurezza a livello di gestore code o di gruppo di condivisione code.

Controllo del numero di ID utente selezionati

RESLEVEL è un profilo del Server di Sicurezza che controlla il numero di ID utente controllati per la sicurezza della risorsa IBM MQ . Di solito, quando un utente tenta di accedere a una risorsa IBM MQ , Security Server controlla l'ID o gli ID utente rilevanti per verificare se l'accesso è consentito a tale risorsa. Definendo un profilo RESLEVEL è possibile controllare se vengono selezionati zero, uno o, se applicabile, due ID utente.

Questi controlli vengono eseguiti su una connessione in base alla connessione e durano per tutta la durata della connessione.

È presente un solo profilo RESLEVEL per ciascun gestore code. Il controllo viene implementato dall'accesso di un ID utente a questo profilo.

Classi IBM MQ RACF maiuscole o minuscole

È ora possibile utilizzare il supporto del profilo RACF con caratteri misti, che consente di utilizzare nomi di risorse con caratteri misti e definire profili IBM MQ RACF per proteggerli.

È possibile scegliere di:

- Continuare ad utilizzare solo le classi IBM MQ RACF maiuscole come nelle release precedenti oppure
- Utilizzare le nuove classi IBM MQ RACF con maiuscole e minuscole.

Senza l'uso di profili RACF con maiuscole e minuscole, è ancora possibile utilizzare nomi di risorse con maiuscole e minuscole in IBM MQ for z/OS ; tuttavia, questi nomi risorsa possono essere protetti solo da profili RACF generici nelle classi IBM MQ in maiuscolo. Quando si utilizza il supporto per il profilo IBM MQ RACF con maiuscole e minuscole, è possibile fornire un livello di protezione più dettagliato definendo i profili IBM MQ RACF nelle classi IBM MQ con maiuscole e minuscole.

z/OS Risorse che è possibile proteggere in IBM MQ for z/OS

Quando un gestore code viene avviato o quando viene richiesto da un comando dell'operatore, IBM MQ for z/OS determina le risorse che si desidera proteggere.

È possibile controllare quali controlli di sicurezza vengono eseguiti per ogni singolo gestore code. Ad esempio, è possibile implementare un numero di controlli di protezione su un gestore code di produzione, ma nessuno su un gestore code di prova.

Sicurezza connessione

Il controllo di sicurezza della connessione viene effettuato quando un programma di applicazione tenta di connettersi a un gestore code. Questa operazione viene eseguita emettendo una richiesta MQCONN

o MQCONNX oppure quando l'iniziatore del canale o l'adattatore CICS o IMS emette una richiesta di connessione.

Se si utilizza la sicurezza a livello di gestore code, è possibile disattivare il controllo di sicurezza della connessione per un determinato gestore code. Tuttavia, se si esegue questa operazione, qualsiasi utente può connettersi a tale gestore code.

Per l'adattatore CICS , viene utilizzato solo l'ID utente dello spazio di indirizzi CICS per il controllo di sicurezza della connessione, non il singolo ID utente del terminale CICS . Per l'adattatore IMS , quando il controllo IMS o le aree dipendenti si collegano a IBM MQ, l'ID utente dello spazio di indirizzo IMS viene controllato. Per l'iniziatore di canali, viene controllato l'ID utente utilizzato dallo spazio di indirizzo dell'iniziatore di canali.

È possibile attivare o disattivare il controllo di sicurezza della connessione a livello di gestore code o di gruppo di condivisione code.

Sicurezza comando

Il controllo di sicurezza dei comandi viene eseguito quando un utente immette un comando MQSC da una delle origini descritte in Origini da cui è possibile immettere comandi MQSC e PCF su IBM MQ for z/OS. È possibile effettuare un controllo separato sulla risorsa specificata dal comando come descritto in “Sicurezza della risorsa comando” a pagina 279.

Se si disattiva il controllo dei comandi, gli emittenti dei comandi non vengono controllati per verificare se dispongono dell'autorizzazione per emettere il comando.

Se i comandi MQSC vengono immessi da una console, la console deve avere l'attributo di autorizzazione della console SYS z/OS . I comandi emessi dai dataset CSQINP1 o CSQINP2 o internamente dal gestore code, sono esenti da tutti i controlli di sicurezza mentre quelli per CSQINPX utilizzano l'ID utente dello spazio di indirizzo dell'iniziatore del canale. È necessario controllare chi può aggiornare questi dataset tramite la normale protezione dei dataset.

È possibile attivare o disattivare il controllo di sicurezza del comando a livello di gestore code o di gruppo di condivisione code.

Sicurezza della risorsa comando

Alcuni comandi MQSC, ad esempio la definizione di una coda locale, implicano la manipolazione delle risorse IBM MQ . Quando la sicurezza della risorsa del comando è attiva, ogni volta che viene emesso un comando che coinvolge una risorsa, IBM MQ controlla se all'utente è consentito modificare la definizione di tale risorsa.

È possibile utilizzare la sicurezza delle risorse di comando per applicare gli standard di denominazione. Ad esempio, un amministratore delle buste paga potrebbe essere autorizzato a eliminare e definire solo le code con nomi che iniziano con "PAYROLL". Se la sicurezza della risorsa del comando è inattiva, non vengono eseguiti controlli di sicurezza sulla risorsa che viene manipolata dal comando. Non confondere la sicurezza della risorsa comando con la sicurezza del comando; i due sono indipendenti.

La disattivazione del controllo di sicurezza delle risorse dei comandi non influisce sul controllo delle risorse eseguito in modo specifico per altri tipi di elaborazione che non implicano comandi.

È possibile attivare o disattivare il controllo di sicurezza della risorsa del comando a livello di gestore code o di gruppo di condivisione code.

Considerazioni sulla sicurezza del canale

Sicurezza canale

Quando si utilizzano i canali, le funzioni di sicurezza disponibili dipendono dal protocollo di comunicazione che si intende utilizzare. Se si utilizza TCP, non vi sono funzioni di sicurezza fornite

con il protocollo di comunicazione, anche se è possibile utilizzare TLS. Se si utilizza APPC, è possibile trasmettere le informazioni sull'ID utente dall'MCA mittente attraverso la rete all'MCA di destinazione per la verifica.

Per entrambi i protocolli, è possibile specificare quali ID utente si desidera controllare per motivi di sicurezza e quanti. Anche in questo caso, le scelte disponibili dipendono da quale protocollo si sta utilizzando, cosa si specifica quando si definisce il canale e le impostazioni RESLEVEL per l'iniziatore del canale.

Per ulteriori informazioni sui tipi di sicurezza del canale disponibili, vedere Record di autenticazione di canale e Panoramica dell'uscita di sicurezza

Riferimenti correlati

“Sicurezza delle risorse API in IBM MQ for z/OS” a pagina 280
Le risorse vengono verificate quando un'applicazione apre un oggetto con una chiamata MQOPEN o MQPUT1 . L'accesso necessario per aprire un oggetto dipende dalle opzioni di apertura specificate quando la coda viene aperta.

z/OS *Sicurezza delle risorse API in IBM MQ for z/OS*

Le risorse vengono verificate quando un'applicazione apre un oggetto con una chiamata MQOPEN o MQPUT1 . L'accesso necessario per aprire un oggetto dipende dalle opzioni di apertura specificate quando la coda viene aperta.

La sicurezza delle risorse API è suddivisa nei seguenti controlli:

- Coda
- Processo
- elenco nomi
- Alterna utente
- Contesto

Non viene eseguito alcun controllo di sicurezza quando si apre l'oggetto del gestore code o quando si accede agli oggetti della classe di memoria.

Coda

Il controllo di sicurezza della coda controlla chi può aprire la coda e con quali opzioni è consentito aprirla. Ad esempio, un utente potrebbe essere autorizzato ad aprire una coda denominata PAYROLL.INCREASE.SALARY per sfogliare i messaggi nella coda (utilizzando l'opzione MQOO_BROWSE), ma non per rimuovere i messaggi dalla coda (utilizzando una delle opzioni MQOO_INPUT_ *). Se si disattiva il controllo delle code, qualsiasi utente può aprire qualsiasi coda con qualsiasi opzione di apertura valida (ovvero, qualsiasi opzione MQOO_ * valida su una chiamata MQOPEN o MQPUT1).

È possibile attivare o disattivare il controllo di sicurezza della coda a livello di gestore code o di gruppo di condivisione code.

Processo

Il controllo di sicurezza del processo viene eseguito quando un utente apre un oggetto di definizione del processo. Se si disattiva la verifica dei processi, qualsiasi utente può aprire qualsiasi processo.

È possibile attivare o disattivare il controllo di sicurezza del processo a livello di gestore code o di gruppo di condivisione code.

Elenco nomi

Il controllo di sicurezza dell'elenco nomi viene effettuato quando un utente apre un elenco nomi. Se si disattiva la verifica degli elenchi nomi, qualsiasi utente può aprire qualsiasi elenco nomi.

È possibile attivare o disattivare il controllo di sicurezza dell'elenco nomi a livello del gestore code o del gruppo di condivisione code.

Alterna utente

La sicurezza utente alternativa controlla se un ID utente può utilizzare l'autorizzazione di un altro ID utente per aprire un oggetto IBM MQ .

Ad esempio:

- Un programma server in esecuzione con ID utente PAYSERV richiama un messaggio di richiesta da una coda inserita nella coda dall'ID utente USER1.
- Quando il programma del server richiama il messaggio di richiesta, elabora la richiesta e reinserisce la risposta nella coda di risposta specificata con il messaggio di richiesta.
- Invece di utilizzare il proprio ID utente (PAYSERV) per autorizzare l'apertura della coda di risposta, il server può specificare un altro ID utente, in questo caso USER1. In questo esempio, la sicurezza dell'utente alternativo controlla se all'ID utente PAYSERV è consentito specificare l'ID utente USER1 come ID utente alternativo quando si apre la coda di risposta.

L'ID utente alternativo viene specificato nel campo *AlternateUserId* del descrittore oggetto (MQOD).

È possibile utilizzare ID utente alternativi su qualsiasi oggetto IBM MQ , ad esempio, processi o elenchi nomi. Non influisce sull'ID utente utilizzato da altri gestori risorse, ad esempio, per la sicurezza CICS o per la sicurezza del dataset z/OS .

Se la sicurezza dell'utente alternativo non è attiva, qualsiasi utente può utilizzare qualsiasi altro ID utente come ID utente alternativo.

È possibile attivare o disattivare il controllo di sicurezza dell'utente alternativo a livello di gestore code o di gruppo di condivisione code.

Contesto

Il contesto è un'informazione applicabile a un particolare messaggio e contenuta nel descrittore del messaggio (MQMD) che fa parte del messaggio. Le informazioni di contesto si trovano in due sezioni:

Sezione Identità
L'utente dell'applicazione che per prima ha inserito il messaggio in una coda. Si compone dei seguenti campi:

- *UserIdentifier*
- *AccountingToken*
- *ApplIdentityData*

Sezione Origine
L'applicazione che ha inserito il messaggio nella coda in cui è attualmente memorizzato. Si compone dei seguenti campi:

- *PutApplType*
- *PutApplName*
- *PutDate*
- *PutTime*
- *ApplOriginData*

Le applicazioni possono specificare i dati di contesto quando viene effettuata una chiamata MQPUT o MQPUT1 . L'applicazione potrebbe generare i dati, i dati potrebbero essere trasmessi da un altro messaggio o il gestore code potrebbe generare i dati per impostazione predefinita. Ad esempio, i programmi del server possono utilizzare i dati di contesto per controllare l'identità del richiedente, ovvero, questo messaggio proviene dall'applicazione corretta? Generalmente, il campo *UserIdentifier* viene utilizzato per determinare l'ID utente di un utente alternativo.

Si utilizza la sicurezza del contesto per controllare se l'utente può specificare una delle opzioni di contesto su una chiamata MQOPEN o MQPUT . Per informazioni sulle opzioni di contesto, consultare Opzioni MQOPEN relative al contesto del messaggio. Per le descrizioni dei campi descrittore del messaggio relativi al contesto, vedere MQMD - Message descriptorMQMD - Message descriptor.

Se si disattiva il controllo di sicurezza del contesto, qualsiasi utente può utilizzare una delle opzioni di contesto consentite dalla sicurezza della coda.

È possibile attivare o disattivare il controllo di sicurezza del contesto a livello di coda, gestore code o gruppo di condivisione code.

z/OS Disponibilità su z/OS

IBM MQ for z/OS ha molte funzioni per l'alta disponibilità. Questo argomento descrive alcune delle considerazioni sulla disponibilità.

Diverse funzionalità di IBM MQ possono aumentare la disponibilità del sistema in caso di errore del gestore code o dell'iniziatore di canali. Per ulteriori informazioni su queste funzioni, consultare le seguenti sezioni:

- Considerazioni sul sysplex
- Code condivise
- Canali condivisi
- IBM MQ disponibilità di rete
- Utilizzo di z/OS ARM (Automatic Restart Manager)
- Utilizzo di z/OS Extended Recovery Facility (XRF)
- Utilizzo dell'attributo z/OS GROUPUR per il ripristino in un gruppo di condivisione code
- Dove trovare ulteriori informazioni sulla disponibilità

Considerazioni sul sysplex

In un *sysplex*, un certo numero di immagini del sistema operativo z/OS collaborano in una singola immagine del sistema e comunicano utilizzando una CF (coupling facility). IBM MQ può utilizzare le funzioni dell'ambiente sysplex per una maggiore disponibilità.

La rimozione delle affinità tra un gestore code e una particolare immagine z/OS consente a un gestore code di essere riavviato su un'immagine z/OS differente in caso di errore dell'immagine. Il meccanismo di riavvio può essere manuale, utilizzare ARM o utilizzare l'automazione del sistema, se si garantisce quanto segue:

- Tutte le serie di pagine, i log, i dataset di avvio, le librerie di codice e i dataset di configurazione del gestore code sono definiti su volumi condivisi.
- La definizione del sottosistema ha un ambito sysplex e un nome univoco all'interno del sysplex.
- Il livello di *codice iniziale* installato su ogni immagine z/OS al momento dell'IPL è lo stesso.
- Gli indirizzi IP virtuali TCP (VIPA) sono disponibili su ogni stack TCP nel sysplex e sono stati configurati i listener TCP IBM MQ e le connessioni in entrata per utilizzare i VIPA piuttosto che i nomi host predefiniti.

Per ulteriori informazioni sull'utilizzo di TCP in un sysplex, consultare *TCP/IP in a sysplex*, SG24-5235, una pubblicazione IBM Redbooks .

È inoltre possibile configurare più gestori code in esecuzione su diverse immagini del sistema operativo in un sysplex per operare come un gruppo di condivisione code, che può sfruttare le code condivise e i canali condivisi per una maggiore disponibilità e bilanciamento del carico di lavoro.

Code condivise

Nell'ambiente del gruppo di condivisione code, un'applicazione può connettersi a qualsiasi gestore code all'interno del gruppo di condivisione code. Poiché tutti i gestori code nel gruppo di condivisione code possono accedere alla stessa serie di code condivise, l'applicazione non dipende dalla disponibilità di un determinato gestore code; qualsiasi gestore code nel gruppo di condivisione code può servire qualsiasi coda. Ciò fornisce una maggiore disponibilità se un gestore code viene arrestato in quanto tutti gli altri gestori code nel gruppo di condivisione code possono continuare l'elaborazione della coda. Per informazioni sull'alta disponibilità delle code condivise, consultare "Vantaggi dell'utilizzo delle code condivise" a pagina 191.

Per migliorare ulteriormente la disponibilità dei messaggi in un gruppo di condivisione code, IBM MQ rileva se un altro gestore code del gruppo si disconnette in modo anomalo dalla CF e completa le unità di lavoro per tale gestore code che sono ancora in sospeso, laddove possibile. È noto come *ripristino peer* ed è descritto in "Ripristino peer" a pagina 269.

Il ripristino peer non può recuperare le unità di lavoro che erano in dubbio al momento dell'errore. È possibile utilizzare ARM (Automatic Restart Manager) per riavviare tutti i sistemi coinvolti nell'errore (CICS, Db2e IBM MQ ad esempio) e per garantire che vengano riavviati tutti sullo stesso nuovo processore. Ciò significa che possono risincronizzarsi e fornire un rapido recupero delle unità di lavoro in dubbio. Ciò è descritto in "Utilizzo di z/OS ARM (Automatic Restart Manager)" a pagina 283.

Canali condivisi

Nell'ambiente del gruppo di condivisione code, IBM MQ fornisce funzioni che forniscono alta disponibilità alla rete. L'iniziatore di canali consente di utilizzare prodotti di rete che bilanciano le richieste di rete su una serie di server idonei e nascondono gli errori del server dalla rete (ad esempio, risorse generiche VTAM). IBM MQ utilizza una porta generica per le richieste in entrata in modo che le richieste di collegamento possano essere instradate a qualsiasi iniziatore di canale disponibile nel gruppo di condivisione code. Ciò è descritto in "Canali condivisi" a pagina 212.

I canali in uscita condivisi prendono i messaggi inviati da una coda di trasmissione condivisa. Le informazioni sullo stato di un canale condiviso vengono conservate in un'unica posizione per l'intero livello del gruppo di condivisione code. Ciò significa che un canale può essere riavviato automaticamente su un iniziatore di canali differente nel gruppo di condivisione code se l'iniziatore di canali, il gestore code o il sottosistema di comunicazioni ha esito negativo. Questo è chiamato *ripristino del canale peer* ed è descritto in Canali in uscita condivisi.

Disponibilità di rete IBM MQ

I messaggi IBM MQ vengono trasmessi dal gestore code al gestore code in una rete IBM MQ utilizzando i canali. È possibile modificare la configurazione a diversi livelli per migliorare la disponibilità di rete di un gestore code e la capacità di un canale IBM MQ di rilevare un problema di rete e di riconnettersi.

TCP *Keepalive* è disponibile per canali TCP/IP. Fa sì che TCP invii periodicamente pacchetti tra le sessioni per rilevare errori di rete. L'attributo canale KAINT determina la frequenza di questi pacchetti per un canale.

AdoptMCA consente a un canale, bloccato nell'elaborazione di ricezione a causa di un'interruzione di rete, di essere terminato e sostituito da una nuova richiesta di connessione. Si controlla AdoptMCA utilizzando la proprietà del gestore code ADOPTMCA con il programma di utilità MQSC o la proprietà MCAType AdoptNewcon l'interfaccia Programmable Command Formats.

ReceiveTimeout impedisce il blocco permanente di un canale in una chiamata di ricezione di rete. I parametri dell'iniziatore di canali RCVTIME e RCVTMIN determinano le caratteristiche di timeout di ricezione per i canali, in funzione del relativo intervallo di heartbeat. Per ulteriori dettagli, consultare Parametro del gestore code .

Utilizzo di z/OS ARM (Automatic Restart Manager)

È possibile utilizzare IBM MQ for z/OS insieme a z/OS ARM (automatic restart manager). Se un gestore code o un iniziatore di canali ha avuto esito negativo, ARM lo riavvia sulla stessa immagine z/OS . Se z/OS non riesce, anche un intero gruppo di sottosistemi e applicazioni correlati ha esito negativo. ARM può riavviare automaticamente tutti i sistemi in errore, in un ordine predefinito, su un'altra immagine z/OS all'interno del sysplex. Questo viene chiamato riavvio tra sistemi.

ARM consente il ripristino rapido delle transazioni in dubbio nell'ambiente della coda condivisa. Fornisce inoltre una maggiore disponibilità se non si utilizzano i gruppi di condivisione code.

È possibile utilizzare ARM per riavviare un gestore code su un'immagine z/OS differente all'interno del sysplex in caso di errore z/OS .

Per abilitare il riavvio automatico, è necessario:

1. Impostare un dataset di accoppiamento ARM.
2. Definire le azioni di riavvio automatico che z/OS deve eseguire in una *politica di ARM*.
3. Avviare la politica di ARM.

Se si desidera riavviare automaticamente i gestori code in immagini z/OS differenti, ogni gestore code in ciascuna immagine z/OS su cui tale gestore code potrebbe essere riavviato deve essere definito con un nome di sottosistema a 4 caratteri univoco a livello del sysplex.

L'utilizzo di ARM con IBM MQ è descritto in Utilizzo di ARM in una rete IBM MQ.

Utilizzo di z/OS Extended Recovery Facility (XRF)

È possibile utilizzare IBM MQ in un ambiente XRF (extended recovery facility). Tutti i dataset di proprietà IBM MQ(codice eseguibile, BSDS, log e serie di pagine) devono essere su DASD condivisi tra i processori XRF attivi e alternativi.

Se si utilizza XRF per il ripristino, è necessario arrestare il gestore code sul processore attivo e avviarlo sul processore alternativo. Per CICS, è possibile farlo utilizzando la tabella dell'elenco comandi (CLT) fornita da CICSoppure l'operatore di sistema può farlo manualmente. Per IMS, questa è un'operazione manuale ed è necessario eseguirla dopo che il sistema IMS di coordinamento ha completato la commutazione del processore.

I programmi di utilità IBM MQ devono essere completati o terminati prima che il gestore code possa passare al processore alternativo. Considera attentamente l'effetto di questa potenziale interruzione quando pianifichi i tuoi piani di recupero XRF.

Evitare che il gestore code venga avviato sul processore alternativo prima che il gestore code sul processore attivo termini. Un avvio prematuro può causare gravi problemi di integrità nei dati, nel catalogo e nel log. L'utilizzo di GRS (Global Resource Serialization) consente di evitare i problemi di integrità impedendo l'utilizzo simultaneo di IBM MQ sui due sistemi. È necessario includere BSDS come risorsa protetta e i processori XRF attivi e alternativi nell'anello GRS.

Utilizzo dell'attributo GROUPUR z/OS per il ripristino in un gruppo di condivisione code

I gruppi di condivisione code (QSG) consentono ulteriori funzioni transazionali descritte in questo argomento. L'attributo GROUPUR consente alle applicazioni client XA di avere qualsiasi ripristino della transazione in dubbio che può essere richiesto, eseguito su qualsiasi membro del QSG.

Se un'applicazione client XA si connette a un gruppo di condivisione code (QSG) tramite un Sysplex, non può garantire a quale gestore code specifico si connette. L'utilizzo dell'attributo GROUPUR da parte dei gestori code all'interno del gruppo di condivisione code può abilitare qualsiasi ripristino della transazione in dubbio che potrebbe essere necessario su qualsiasi membro del QSG. Anche se il gestore code a cui l'applicazione è stata inizialmente connessa non è disponibile, è possibile che venga eseguito il ripristino della transazione.

Questa funzione libera l'applicazione client XA da qualsiasi dipendenza su membri specifici del QSG e quindi estende la disponibilità del gestore code. Il gruppo di condivisione code viene visualizzato all'applicazione transazionale come una sola entità che fornisce tutte le funzioni IBM MQ e senza un singolo punto di errore del gestore code.

Questa funzionalità non è evidente per l'applicazione transazionale.

Dove trovare maggiori informazioni sulla disponibilità

È possibile trovare ulteriori informazioni su questi argomenti dalle seguenti origini:

Tabella 24. Dove trovare maggiori informazioni sulla disponibilità

Argomento	**Dove cercare**
Gruppi di condivisione code	"Code condivise e gruppi di condivisione code" a pagina 171
Parametri di sistema	Configurazione dei parametri di sistema
Utilizzo di Automatic Restart Manager Programmi di utilità	Utilizzo di ARM in una rete IBM MQ
Comandi MQSC	Comandi MQSC

z/OS Monitoraggio e statistiche su IBM MQ for z/OS

IBM MQ for z/OS dispone di una serie di funzionalità per il controllo del gestore code e la raccolta di statistiche.

IBM MQ fornisce funzioni per il monitoraggio del sistema e la raccolta di statistiche. Per ulteriori informazioni su queste funzioni, consultare le seguenti sezioni:

- "Controllo online" a pagina 285
- "IBM MQ traccia" a pagina 285
- "Eventi" a pagina 286

Controllo online

IBM MQ include i seguenti comandi per monitorare lo stato degli oggetti IBM MQ :

- DISPLAY CHSTATUS visualizza lo stato di uno specifico canale.
- DISPLAY QSTATUS visualizza uno stato di una coda specificata.
- DISPLAY CONN visualizza lo stato di un collegamento specificato.

Per ulteriori informazioni su questi comandi, consultare Comandi MQSC.

IBM MQ traccia

IBM MQ fornisce una funzione di traccia che è possibile utilizzare per raccogliere le seguenti informazioni mentre il gestore code è in esecuzione:

Statistiche delle prestazioni
La traccia delle statistiche raccoglie le informazioni riportate di seguito per consentire il monitoraggio delle prestazioni e l'ottimizzazione del sistema:

- Conteggi di richieste di MQI differenti (statistiche del gestore messaggi)

- Conteggi di diverse richieste di oggetti (statistiche del gestore dati)
- Informazioni sull'utilizzo di Db2 (statistiche del gestore Db2)
- Informazioni sull'utilizzo della funzione di accoppiamento (statistiche del gestore della funzione di accoppiamento)
- Informazioni sull'utilizzo SMDS (statistiche del dataset del messaggio condiviso)
- Informazioni sull'utilizzo del pool di buffer (statistiche del gestore buffer)
- Informazioni sulla registrazione (statistiche del gestore log)
- Informazioni sull'utilizzo della memoria (statistiche del gestore memoria)
- Informazioni sulle richieste di blocco (statistiche gestore blocchi)

Dati contabili

- La traccia di account raccoglie informazioni sul tempo del processore impiegato nell'elaborazione delle chiamate MQI e sul numero di richieste MQPUT e MQGET effettuate da un particolare utente.
- IBM MQ può anche raccogliere informazioni su ciascuna attività utilizzando IBM MQ. Questi dati vengono raccolti come record di account a livello di thread. Per ogni thread, IBM MQ raccoglie anche le informazioni su ogni coda utilizzata da tale thread.

I dati generati dalla traccia vengono inviati a SMF (System Management Facility) o GTF (Generalized Trace Facility).

Eventi

Gli eventi IBM MQ forniscono informazioni su errori, avvertenze e altre ricorrenze significative in un gestore code. Incorporando questi eventi nella propria applicazione di gestione del sistema, è possibile monitorare le attività su molti gestori code, per più applicazioni IBM MQ . In particolare, è possibile monitorare tutti i gestori code nel sistema da un singolo gestore code.

Gli eventi possono essere notificati tramite un meccanismo di report scritto dall'utente a un'applicazione di gestione che supporta la presentazione degli eventi a un operatore. Gli eventi abilitano anche le applicazioni che agiscono come agenti per altre reti di gestione, ad esempio NetView, per monitorare i report e creare gli avvisi appropriati.

Attività correlate

Utilizzo della traccia IBM MQ

Utilizzo di eventi IBM MQ

z/OS Disposizione dell'unità di recupero su z/OS

Alcune applicazioni transazionali possono utilizzare un'unità di disposizione di ripristino GROUP, anziché QMGR, quando sono connesse a un gestore code in un gruppo di condivisione code (QSG), specificando il nome QSG quando si connettono invece del nome del gestore code. Ciò consente al recupero delle transazioni di essere più flessibile e robusto, rimuovendo il requisito di riconnettersi allo stesso gestore code in QSG.

Le transazioni avviate dalle applicazioni che si sono connesse utilizzando il nome del gruppo di condivisione code hanno anche un'unità di ripristino GROUP.

Quando un'applicazione transazionale si connette a un'unità di disposizione di ripristino GROUP, viene connessa logicamente al gruppo di condivisione code e non ha un'affinità con alcun gestore code specifico. Qualsiasi transazione di commit a due fasi avviata che abbia completato la phase-1 del processo di commit, ossia che sia in dubbio, può essere interrogata e risolta, quando si è connessi a qualsiasi gestore code all'interno di QSG. In uno scenario di recupero, ciò significa che il coordinatore della transazione non deve riconnettersi allo stesso gestore code, che potrebbe non essere disponibile in quel momento.

Le applicazioni che si connettono con una disposizione di unità di ripristino QMGR hanno un'affinità diretta con il gestore code a cui sono connesse. In uno scenario di ripristino, il coordinatore delle transazioni deve riconnettersi allo stesso gestore code per risolvere eventuali transazioni in dubbio, indipendentemente dal fatto che il gestore code appartenga o meno a un gruppo di condivisione code.

Quando le applicazioni specificano un nome gruppo di condivisione code e quindi si connettono a un gestore code in un QSG con un'unità di disposizione di ripristino GROUP, il gruppo di condivisione code è logicamente un gestore risorse separato. Ciò significa che le transazioni in dubbio sono visibili a un'applicazione solo se si riconnette con la stessa unità di disposizione di ripristino. Le transazioni in dubbio con un'unità QMGR di disposizione di recupero non sono visibili per le applicazioni che hanno collegato un'unità GROUP di disposizione di recupero e viceversa.

Concetti correlati

“Abilitazione delle unità di ripristino GROUP” a pagina 287
Un gruppo di condivisione code può configurare e abilitare il supporto per le unità di ripristino GROUP.

“Supporto applicazioni” a pagina 288
Utilizzare questa pagina per determinare quali applicazioni possono connettersi a un'unità di ripristino GROUP.

z/OS Abilitazione delle unità di ripristino GROUP

Un gruppo di condivisione code può configurare e abilitare il supporto per le unità di ripristino GROUP.

Per utilizzare le unità di ripristino GROUP su un gestore code in un QSG, abilitare l'attributo GROUPUR del gestore code. Per ulteriori informazioni su questo concetto, consultare “Disposizione dell'unità di recupero su z/OS” a pagina 286 prima di leggere il resto di questo argomento.

Quando l'attributo GROUPUR gestore code è abilitato, il gestore code accetta nuove connessioni con un'unità GROUP di disposizione del recupero. Se si disabilita questo attributo, le nuove connessioni con questa disposizione non vengono accettate, anche se le applicazioni già connesse non vengono influenzate fino alla disconnessione.

Quando un'applicazione si connette a un'unità di disposizione di ripristino GROUP e interroga le transazioni in dubbio o tenta di risolvere una transazione avviata altrove nel gruppo di condivisione code (QSG), il gestore code a cui è ora connesso deve essere in grado di comunicare con gli altri membri del gruppo di condivisione code in modo che possa elaborare la richiesta. A tale scopo, utilizza una coda condivisa denominata SYSTEM.QSG.UR.RESOLUTION.QUEUE. Questa coda deve trovarsi su una struttura dell'applicazione recuperabile denominata CSQSYSAPPL. La struttura deve essere ripristinabile perché i messaggi persistenti vengono memorizzati su questa coda durante l'elaborazione delle richieste di risoluzione.

Prima di poter abilitare le unità di ripristino GROUP, è necessario assicurarsi che la struttura CFS e la coda condivisa siano definite. È possibile utilizzare le definizioni nell'esempio CSQ4INSS . Quando la coda viene definita o rilevata durante l'avvio, ogni gestore code nel gruppo di condivisione code apre la coda in modo che possa ricevere le richieste in entrata. Se si desidera eliminare o spostare la coda poiché è stata definita in modo non corretto, è possibile richiedere che i gestori code chiudano i relativi handle aperti aggiornando l'oggetto della coda per inibire le richieste MQGET. Una volta apportate le correzioni necessarie, consentire alle applicazioni di richiamare i messaggi dalla coda, indica a ciascun gestore code di riaprirlo. Utilizzare il comando DISPLAY QSTATUS per identificare quali handle sono aperti su una coda.

Una volta completata questa configurazione, è possibile abilitare le unità di ripristino GROUP su ogni gestore code a cui si desidera che le applicazioni transazionali possano connettersi con un'unità di disposizione di ripristino GROUP. Non è necessario che siano tutti i gestori code all'interno del gruppo di condivisione code, ma se si sceglie di abilitare solo questa funzionalità su un sottoinsieme del gruppo di condivisione code, è necessario assicurarsi che le applicazioni tentino solo di connettersi ai gestori code su cui è stata abilitata. Per ulteriori informazioni, consultare “Supporto applicazioni” a pagina 288.

Quando si tenta di abilitare l'attributo GROUPUR del gestore code, vengono eseguiti diversi controlli di configurazione. Il gestore code verifica che:

- Appartiene a un gruppo di condivisione code.

- La coda condivisa denominata SYSTEM.QSG.UR.RESOLUTION.QUEUE è stato definito, in base alla definizione in CSQ4INSS.
- Il SISTEMA SYSTEM.QSG.UR.RESOLUTION.QUEUE è su una struttura CF recuperabile denominata CSQSYSAPPL.

Se uno dei controlli precedenti non riesce, l'attributo GROUPUR rimane disabilitato e viene restituito un codice messaggio.

Questi controlli di configurazione vengono eseguiti anche all'avvio del gestore code se l'attributo del gestore code è abilitato. Se uno dei controlli ha esito negativo durante l'avvio, le unità di ripristino GROUP sono disabilitate e il gestore code emette un messaggio che identifica il controllo non riuscito. Una volta eseguita l'azione correttiva necessaria, è necessario riabilitare l'attributo del gestore code.

z/OS Supporto applicazioni

Utilizzare questa pagina per determinare quali applicazioni possono connettersi a un'unità di ripristino GROUP.

Il supporto per la disposizione di unità di ripristino GROUP è limitata a determinati tipi di applicazioni transazionali per cui IBM MQ for z/OS è un gestore risorse ma non il coordinatore della transazione. Le applicazioni transazionali attualmente supportate sono:

- Applicazioni client transazionali estese IBM MQ
- Applicazioni IBM MQ classes for JMS in esecuzione in un application server, ad esempio WebSphere Application Server.
- Applicazioni CICS in esecuzione in CICS Transaction Server 4.2 o successive, quando la definizione della risorsa CICS MQCONN è configurata con RESYNCMEMBER (GROUPRESYNC).

Concetti correlati

“Applicazioni client transazionali estese IBM MQ” a pagina 288
Utilizzare questa pagina per determinare il modo in cui le applicazioni client transazionali estese IBM MQ possono utilizzare la disposizione dell'unità di ripristino GROUP.

“CICS applicazioni” a pagina 289
Utilizzare questa pagina per stabilire in che modo CICS può utilizzare la disposizione dell'unità di ripristino GROUP.

z/OS *Applicazioni client transazionali estese IBM MQ*

Utilizzare questa pagina per determinare il modo in cui le applicazioni client transazionali estese IBM MQ possono utilizzare la disposizione dell'unità di ripristino GROUP.

Un esempio di applicazione client transazionale estesa IBM MQ è quello che utilizza JMS e viene eseguito in WebSphere Application Server, collegandosi a IBM MQ su TCP/IP, piuttosto che ai bind locali. Queste applicazioni client si collegano a IBM MQ for z/OS tramite connessioni di rete, ad esempio tramite TCP/IP. Per queste applicazioni, è il valore specificato per il parametro QMNAME della stringa xa_info inoltrata nella chiamata xa_open che specifica se viene utilizzata una disposizione di ripristino dell'unità QMGR o GROUP. Per ulteriori informazioni su xa_open, consultare Il formato di una stringa xa_open e Elaborazione di errori aggiuntivi per xa_open. Per le applicazioni JMS , ciò viene eseguito specificando il nome del gruppo di condivisione code (QSG) in ConnectionFactory anziché il nome di un determinato gestore code.

Per consentire alle applicazioni client XA di trarre vantaggio dall'utilizzo dell'unità GROUP di disposizione del ripristino, è necessario configurare la propria configurazione TCP/IP per consentire l'indirizzamento delle applicazioni client ai gestori code nel gruppo di condivisione code per cui è abilitato l'attributo GROUPUR, piuttosto che a un gestore code specifico. Una delle tecnologie di indirizzi IP virtuali dinamici che è possibile utilizzare a tale scopo è il distributore z/OS SysPlex . Per ulteriori dettagli, consultare z/OS Communications Server e z/OS Competenze di base: Indirizzamento virtuale dinamico . Se si desidera abilitare le unità di ripristino GROUP su un sottoinsieme di gestori code nel proprio gruppo di condivisione code, assicurarsi che le proprie applicazioni client non possano essere instradate a quelle su cui non sono abilitate.

Le applicazioni client non devono collegarsi al gruppo di condivisione code utilizzando i canali condivisi.

z/OS *CICS applicazioni*

Utilizzare questa pagina per stabilire in che modo CICS può utilizzare la disposizione dell'unità di ripristino GROUP.

CICS 4.2 e successive fornisce l'opzione di risincronizzazione del gruppo, RESYNCMEMBER (GROUPRESYNC) in una definizione della risorsa MQCONN. Un CICS configurato con questa opzione può connettersi a qualsiasi gestore code adatto in un gruppo di condivisione code in esecuzione sulla stessa LPAR della regione CICS . Per supportare l'opzione CICS GROUPRESYNC, un gestore code deve essere in esecuzione in MQ V7.1 o versione successiva ed essere abilitato per il supporto GROUPUR.

Le transazioni in esecuzione all'interno di una regione CICS connessa a MQ utilizzando GROUPRESYNC creano unità di lavoro con la disposizione dell'unità di ripristino GROUP.

È possibile utilizzare RESYNCMEMBER (GROUPRESYNC) per abilitare un ripristino più rapido dopo un malfunzionamento del gestore code in quanto consente alla regione CICS di connettersi immediatamente a un gestore code alternativo idoneo in esecuzione sulla stessa LPAR, risolvendo eventuali transazioni in dubbio, se necessario, senza attendere il riavvio del gestore code.

RESYNCMEMBER (GROUPRESYNC) consente inoltre opzioni di riavvio più flessibili per CICS. Una regione CICS con la relativa connessione MQ configurata per utilizzare le code condivise GROUPRESYNC e MQ può essere riavviata su qualsiasi LPAR in cui è presente un gestore code in esecuzione come membro dello stesso gruppo di condivisione code.

z/OS IBM MQ e altri prodotti z/OS

Utilizzare questo argomento per comprendere come IBM MQ può lavorare con altri prodotti z/OS .

Concetti correlati

“IBM MQ e CICS” a pagina 289
Tutte le versioni CICS supportate da IBM MQ 9.0.0e successive, utilizzano la versione fornita da CICS dell'adattatore e del bridge.

“IBM MQ for z/OS e WebSphere Application Server” a pagina 296
Utilizzare questo argomento per comprendere l'utilizzo di IBM MQ for z/OS da parte di WebSphere Application Server.

Riferimenti correlati

“IBM MQ e IMS” a pagina 291
Utilizzare questo argomento per comprendere come IBM MQ funziona con IMS. L'adattatore IMS consente di collegare il gestore code a IMSe consente alle applicazioni IMS di utilizzare MQI.

“IBM MQ e gli adattatori z/OS Batch, TSO e RRS” a pagina 294
Utilizzare questo argomento per comprendere come IBM MQ funziona con gli adattatori z/OS Batch, TSO e RRS.

z/OS IBM MQ e CICS

Tutte le versioni CICS supportate da IBM MQ 9.0.0e successive, utilizzano la versione fornita da CICS dell'adattatore e del bridge.

Per ulteriori informazioni sulla configurazione dell'adattatore IBM MQ CICS e dei componenti IBM MQ CICS bridge , consultare la sezione Configuring connections to IBM MQ della documentazione CICS .

Attività correlate

Utilizzo di IBM MQ con CICS

z/OS Collegamento gruppo CICS

Il collegamento del gruppo CICS consente a una regione CICS di connettersi a qualsiasi membro attivo di un gruppo di condivisione code IBM MQ sulla stessa LPAR piuttosto che specificare un singolo gestore code. CICS si connette ancora a un singolo gestore code alla volta.

Sono necessari almeno due gestori code sulla LPAR per supportare il collegamento del gruppo CICS . L'utilizzo del collegamento del gruppo fornisce una maggiore disponibilità poiché non è necessario che un determinato gestore code sia attivo. CICS si connette a qualsiasi gestore code nel gruppo di condivisione code sulla LPAR.

Per ulteriori informazioni, consultare la documentazione CICS sulla risorsa MQCONN.

CICS tenta di connettersi a MQNAME come se fosse un gestore code:

- Se il gestore code esiste ed è attivo, la connessione funzionerà.
- Se la connessione non riesce, CICS interroga lo stato dei gestori code nel gruppo per verificare quali sono attivi sulla stessa LPAR.
- Se sono attivi più gestori code, CICS ricerca RESYNCMEMBER (YES) e lo stato UOW per determinare se CICS deve connettersi o deve connettersi a un determinato membro o attendere se non è attivo.
- Se non è necessario connettersi a un determinato membro, CICS seleziona un gestore code (utilizzando un algoritmo di randomizzazione).
- CICS tenta di collegarsi al gestore code scelto.
- Se il tentativo ha esito negativo, a seconda del codice di ritorno, CICS sceglie il membro successivo, quindi passa di nuovo attraverso il loop di selezione.
- Se nessun gestore code è attivo, CICS emette più connessioni all'elenco di gestori code e attende su ECBLIST fino a quando il primo gestore code non diventa disponibile.

Concetti correlati

“Group units of recovery (GROUPUR) per CICS” a pagina 290
IBM MQ GROUPUR per CICS fornisce il ripristino peer per le unità di lavoro in dubbio in un QSG (queue sharing group). Un gestore code IBM MQ può risolvere unità di lavoro in dubbio per conto di un altro gestore code nel gruppo di condivisione code. Ciò significa che se CICS si riconnette tramite un gruppo collegato a un gestore code differente in QSG, può risolvere le transazioni in dubbio da una precedente connessione IBM MQ .

Informazioni correlate

Supporto per i gruppi di condivisione code IBM MQ

z/OS Group units of recovery (GROUPUR) per CICS

IBM MQ GROUPUR per CICS fornisce il ripristino peer per le unità di lavoro in dubbio in un QSG (queue sharing group). Un gestore code IBM MQ può risolvere unità di lavoro in dubbio per conto di un altro gestore code nel gruppo di condivisione code. Ciò significa che se CICS si riconnette tramite un gruppo collegato a un gestore code differente in QSG, può risolvere le transazioni in dubbio da una precedente connessione IBM MQ .

Se una regione CICS funziona con un gestore code e il gestore code termina in modo anomalo, tutte le transazioni in dubbio vengono recuperate. Ciò elimina la necessità per la regione CICS di attendere il riavvio del gestore code che stava gestendo e quindi risolvere eventuali unità di lavoro in dubbio. Ciò significa che sono necessari almeno due gestori code sulla LPAR, in modo che CICS possa connettersi a un altro gestore code in caso di terminazione anomala del primo gestore code.

La nuova impostazione RESYNCMEMBER (GROUPRESYNC) sulla definizione MQCONN CICS :

- Utilizza la funzione di collegamento del gruppo IBM MQ e il ripristino peer.
- Richiede un gestore code con l'attributo GROUPUR abilitato.
- Supporta ancora le impostazioni CICS MQCONN RESYNCMEMBER esistenti (YES e NO):
  - Utilizza la funzione di collegamento del gruppo CICS esistente e nessun ripristino peer.
  - La modifica delle impostazioni di RESYNCMEMBER diventa effettiva la volta successiva che CICS si connette a IBM MQ.

Concetti correlati

“Abilitazione delle unità di ripristino GROUP” a pagina 287

Un gruppo di condivisione code può configurare e abilitare il supporto per le unità di ripristino GROUP.

z/OS IBM MQ e IMS

Utilizzare questo argomento per comprendere come IBM MQ funziona con IMS. L'adattatore IMS consente di collegare il gestore code a IMSe consente alle applicazioni IMS di utilizzare MQI.

Il bridge facoltativo aggiuntivo IBM MQ - IMS consente alle applicazioni di eseguire un'applicazione IMS che non utilizza MQI. Ciò significa che è possibile utilizzare le proprie applicazioni legacy con IBM MQ, senza doverle riscrivere.

Per ulteriori informazioni su questi componenti, consultare i seguenti topic secondari:

Concetti correlati
Applicazioni bridge IMS e IMS su IBM MQ for z/OS
Attività correlate
Impostazione dell'adattatore IMS
Impostazione del bridge IMS
Utilizzo dell'adattatore IMS
Riferimenti correlati
MQIIH - Intestazione informazioni IMS

z/OS L'adattatore IMS

L'adattatore IMS è un'interfaccia tra programmi applicativi IMS e un sottosistema IBM MQ .

Gli adattatori IBM MQ abilitano ambienti di applicazioni differenti per inviare e ricevere messaggi tramite una rete di accodamento messaggi. L'adattatore IMS è l'interfaccia tra i programmi di applicazione IMS e un sottosistema IBM MQ . Consente ai programmi applicativi IMS di utilizzare MQI.

L'adattatore IMS riceve e interpreta le richieste di accesso a IBM MQ utilizzando ESAF (External Subsystem Attach Facility) fornito da IMS. Di solito, IMS si collega automaticamente a IBM MQ senza l'intervento dell'operatore.

L'adattatore IMS fornisce l'accesso alle risorse IBM MQ per i programmi in esecuzione nelle seguenti modalità o stati:

- Modalità attività (TCB)
- Stato problema
- Modalità non cross - memory
- Modalità registro di non accesso

L'adattatore fornisce un thread di collegamento da un TCB (task control block) dell'applicazione a IBM MQ.

L'adattatore supporta un protocollo di commit a due fasi per le modifiche apportate alle risorse di proprietà di IBM MQ con IMS che funge da coordinatore del punto di sincronizzazione. Le conversazioni dove IMS non è il coordinatore del punto di sincronizzazione, ad esempio le conversazioni protette da APPC (SYNCLVL = SYNCPT), non sono supportate dall'adattatore IMS .

L'adattatore fornisce anche una transazione di controllo trigger (CSQQTRMN). Ciò è descritto in “Il controllo trigger IMS” a pagina 292.

È possibile utilizzare IBM MQ con IMS XRF (Extended Recovery Facility) per facilitare il ripristino da errore IMS .

Nota: A partire da IMS 15.2 XRF (Extended Recovery Facility) non è più supportato. Consultare la documentazione di IMS per ulteriori informazioni.

Utilizzo dell'adattatore

I programmi applicativi e l'adattatore IMS vengono eseguiti nello stesso spazio di indirizzo. Il gestore code è separato, nel proprio spazio di indirizzo.

È necessario collegare - modificare ogni programma che emette una o più chiamate MQI a un modulo di interfaccia di linguaggio IMS adatto e, a meno che non utilizzi chiamate MQI dinamiche, il programma stub API fornito da IBM MQ, CSQQSTUB. Quando l'applicazione emette una chiamata MQI, lo segnaposto trasferisce il controllo all'adattatore tramite l'interfaccia del sottosistema esterno IMS , che gestisce l'elaborazione della richiesta dal gestore code messaggi.

Gestione e funzionamento del sistema con IMS

Un operatore di terminale IMS autorizzato può immettere comandi IMS per controllare e monitorare la connessione a IBM MQ. Tuttavia, l'operatore del terminale IMS non ha alcun controllo sullo spazio di indirizzi IBM MQ . Ad esempio, l'operatore non può chiudere IBM MQ da uno spazio di indirizzo IMS .

Limitazioni

Le seguenti chiamate API IBM MQ non sono supportate in una applicazione che utilizza l'adattatore IMS :

- MQCB
- MQCB_FUNZIONE
- MQCTL

Il controllo trigger IMS

Il controllo trigger IMS (**CSQQTRMN**) è un'applicazione IBM MQfornita IMS che avvia una transazione IMS quando si verifica un evento IBM MQ , ad esempio quando un messaggio viene inserito in una coda specifica.

Modalità di funzionamento

Quando un messaggio viene inserito in una coda di messaggi dell'applicazione, viene generato un trigger se vengono soddisfatte le condizioni del trigger. Il gestore code scrive quindi un messaggio (contenente alcuni dati definiti dall'utente), noto come *messaggio trigger*, nella coda di iniziazione specificata per tale coda messaggi. In ambiente IMS , è possibile avviare un'istanza di CSQQTRMN per monitorare una coda di iniziazione e richiamare i messaggi trigger da essa quando arrivano. In genere, CSQQTRMN pianifica un'altra transazione IMS mediante un INSERT (ISRT) nella coda messaggi IMS . L'applicazione IMS avviata legge il messaggio dalla coda messaggi dell'applicazione e lo elabora. CSQQTRMN deve essere eseguito come BMP non messaggio.

Ogni copia di CSQQTRMN fornisce una singola coda di iniziazione. Una volta avviato, il controllo dei trigger viene eseguito fino al termine di IBM MQ o IMS .

La macro APPLCTN per CSQQTRMN deve specificare SCHDTYP=PARALLEL.

Poiché il controllo trigger è un BMP orientato al batch, le transazioni IMS avviate dal controllo trigger contengono quanto segue:

- Spazi vuoti nel campo LTERM di IOPCB
- Il nome PSB del BMP del controllo trigger nel campo ID utente dell'IOPCB

Se la transazione di destinazione IMS è protetta da Security Server (precedentemente noto come RACF), potrebbe essere necessario definire CSQQTRMN come ID utente per Security Server.

z/OS Il bridge IBM MQ - IMS

Il bridge IBM MQ - IMS è il componente di IBM MQ for z/OS che consente l'accesso diretto dalle applicazioni IBM MQ alle applicazioni sul sistema IMS .

Il bridge IBM MQ - IMS abilita il *supporto MQI implicito*. Ciò significa che è possibile riprogettare le applicazioni legacy che erano controllate da terminali connessi a 3270 per essere controllate da messaggi IBM MQ , senza doverle riscrivere, ricompilare o ricollegare. Il bridge è un client OTMA (IMS *Open Transaction Manager Access*).

Nelle applicazioni bridge non sono presenti chiamate IBM MQ all'interno dell'applicazione IMS . L'applicazione ottiene il suo input utilizzando un GET UNIQUE (GU) all'IOPCB e invia il suo output utilizzando un ISRT all'IOPCB. Le applicazioni IBM MQ utilizzano l'intestazione IMS (la struttura MQIIH) nei dati del messaggio per garantire che le applicazioni possano essere eseguite come quando guidate da terminali non programmabili. Se si utilizza un'applicazione IMS che elabora messaggi a più segmenti, tenere presente che tutti i segmenti devono essere contenuti all'interno di un messaggio IBM MQ .

Il bridge IMS è illustrato in Figura 78 a pagina 293.

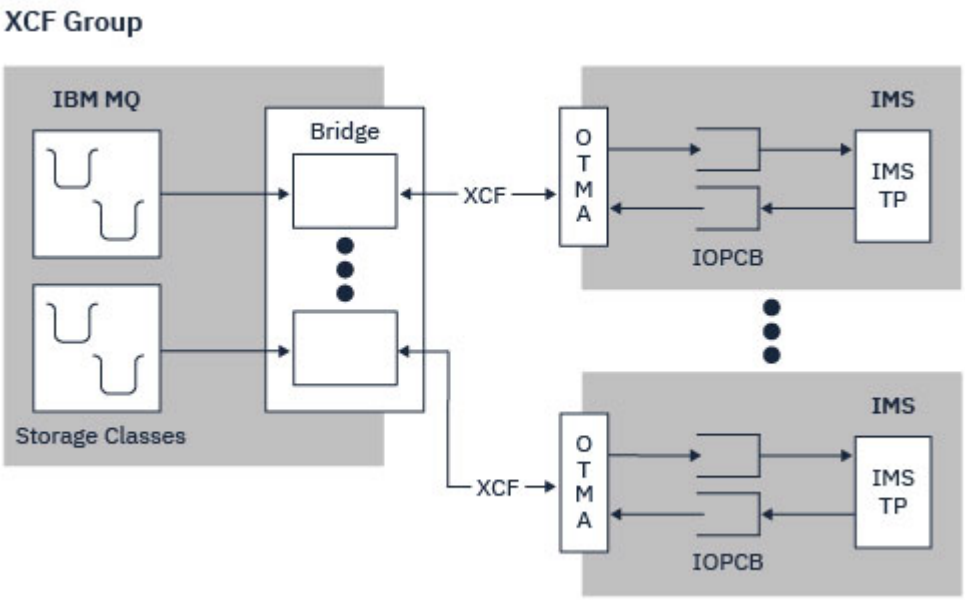


Figura 78. Il bridge IBM MQ - IMS

Un gestore code può connettersi a uno o più sistemi IMS e più gestori code possono connettersi a un sistema IMS . L'unica limitazione è che devono appartenere tutti allo stesso gruppo XCF e devono essere tutti nello stesso sysplex.

Consultare Impostazione del bridge IMS per informazioni sull'impostazione di un bridge IMS e sull'aggiunta di una connessione IMS aggiuntiva allo stesso gestore code.

Cos' è OTMA?

La funzione OTMA di IMS è un protocollo client / server senza connessione basato su transazione che viene eseguito su IMS. Funziona come interfaccia per i server di comunicazioni basati su host che accedono alle applicazioni IMS TM tramite z/OS XCF (Cross Systems coupling facility).

OTMA consente ai client di collegarsi a IMS per fornire prestazioni elevate per le interazioni tra client e IMS per una rete di grandi dimensioni o per un numero elevato di sessioni. OTMA è implementato in un ambiente sysplex z/OS . Pertanto, il dominio di OTMA è limitato al dominio di XCF.

Monitoraggio risorse OTMA

Il supporto per i messaggi del protocollo OTMA x '3C', disponibili in IMS v10 o versioni successive, è incluso nel bridge IBM MQ - IMS in IBM MQ for z/OS. Questi messaggi vengono inviati a client OTMA da IMS per notificare lo stato di integrità.

Se un partner IMS non è in grado di elaborare il volume di richieste di transazione inviate, notificherà a IBM MQ che si è verificata un'avvertenza di flusso abnorme. In risposta, IBM MQ rallenterà la frequenza con cui le richieste vengono inviate sul bridge.

Se IMS non è ancora in grado di elaborare le richieste di transazione e si verifica una condizione di piena piena, tutti i TPIP per il partner IMS vengono sospesi. Dopo la notifica dal partner IMS che la condizione di inondazione o di avvertenza è stata sollevata IBM MQ riprenderà tutti i TPIPE sospesi, se appropriato, e aumenterà gradualmente la velocità con cui le richieste di transazione vengono inviate fino a raggiungere la velocità massima. I messaggi della console vengono emessi da IBM MQ in seguito a una modifica dello stato dei partner IMS .

Se vengono utilizzati i partner IMS v10 , è necessario assicurarsi che la PTF UK45082 sia stata applicata.

Inoltro di transazioni IMS da IBM MQ

Per inoltrare una transazione IMS che utilizza il bridge, le applicazioni inserendo i messaggi su una coda IBM MQ come al solito. I messaggi contengono dati di transazione IMS ; possono avere un'intestazione IMS (la struttura MQIIH) o consentire al bridge IBM MQ - IMS di fare ipotesi sui dati nel messaggio.

IBM MQ inserisce quindi il messaggio in una coda IMS (viene prima accodato in IBM MQ per consentire l'utilizzo dei punti di sincronizzazione per garantire l'integrità dei dati). La classe di archiviazione della coda IBM MQ determina se la coda è una *coda OTMA* (ovvero, una coda utilizzata per trasmettere i messaggi al bridge IBM MQ - IMS) e il particolare partner IMS a cui vengono inviati i dati del messaggio.

I gestori code remoti possono anche avviare transazioni IMS scrivendo su queste code OTMA su IBM MQ for z/OS.

I dati restituiti dal sistema IMS vengono scritti direttamente nella coda di risposta IBM MQ specificata nella MQMD (message descriptor structure). (Questa potrebbe essere una coda di trasmissione al gestore code specificato nel campo **`ReplyToQMgr`** di MQMD.)

Concetti correlati

Applicazioni bridge IMS e IMS su IBM MQ for z/OS

Attività correlate

Personalizzazione del bridge IMS

Riferimenti correlati

“IBM MQ e IMS” a pagina 291

Utilizzare questo argomento per comprendere come IBM MQ funziona con IMS. L'adattatore IMS consente di collegare il gestore code a IMSe consente alle applicazioni IMS di utilizzare MQI.

z/OS IBM MQ e gli adattatori z/OS Batch, TSO e RRS

Utilizzare questo argomento per comprendere come IBM MQ funziona con gli adattatori z/OS Batch, TSO e RRS.

Introduzione agli adattatori batch

Gli adattatori Batch/TSO sono l'interfaccia tra i programmi applicativi IBM MQ e z/OS in esecuzione in JES, TSO o z/OS UNIX System Services. Questi adattatori consentono ai programmi di applicazione z/OS di utilizzare MQI.

Gli adattatori forniscono l'accesso alle risorse IBM MQ per i programmi in esecuzione nei seguenti modi o stati:

- Modalità attività (TCB)

- Stato del problema o del supervisore
- Modalità non cross - memory
- Modalità registro di non accesso

Le connessioni tra i programmi applicativi e IBM MQ sono a livello di attività. Gli adattatori forniscono un thread di connessione da un TCB (task control block) dell'applicazione a IBM MQ.

L'adattatore Batch/TSO supporta un protocollo di commit a fase singola per le modifiche apportate alle risorse di proprietà di IBM MQ. Non supporta protocolli di commit a più fasi. L'adattatore RRS consente alle applicazioni IBM MQ di partecipare a protocolli di commit a due fasi con altri prodotti abilitati RRS, coordinati da z/OS RRS (Resource Recovery Services).

Gli adattatori utilizzano il servizio z/OS STIMERM per pianificare un evento asincrono ogni secondo. Questo evento esegue un blocco di richieste di interruzione (IRB) che non implica alcuna attesa da parte dell'attività dell'applicazione batch. Questo IRB controlla se la BCE di terminazione IBM MQ è stata pubblicata. Se la BCE di terminazione è stata inviata, l'IRB pubblica qualsiasi ECB dell'applicazione in attesa di un evento in IBM MQ (ad esempio, un segnale o un'attesa).

Adattatore Batch/TSO

L'adattatore IBM MQ Batch/TSO fornisce il IBM MQ supporto per le applicazioni z/OS Batch e TSO. Tutti i programmi applicativi eseguiti in z/OS Batch o TSO devono avere il programma stub API CSQBSTUB link - modificato con essi. Lo stub fornisce all'applicazione l'accesso a tutte le chiamate MQI. Si utilizza il commit e il backout a fase singola per le applicazioni emettendo le chiamate MQI **MQCMIT** e **MQBACK**.

L'adattatore RRS

RRS (Resource Recovery Services) è un componente secondario di z/OS che fornisce un servizio a livello di sistema per il coordinamento del commit a due fasi tra i prodotti z/OS . L'adattatore IBM MQ Batch/TSO RRS (l'adattatore RRS) fornisce il supporto IBM MQ per le applicazioni z/OS Batch e TSO che desiderano utilizzare questi servizi. L'adattatore RRS consente a IBM MQ di diventare un partecipante completo nel coordinamento RRS. Le applicazioni possono partecipare all'elaborazione del commit in due fasi con altri prodotti che supportano RRS (ad esempio Db2).

L'adattatore RRS fornisce due stub; è necessario collegare i programmi di applicazione che desiderano utilizzare RRS con uno di questi stub.

CSQBRSTB
: Questo stub consente di utilizzare il commit e il backout a due fasi per applicazioni utilizzando i servizi di ripristino delle risorse richiamabili RRS invece delle chiamate MQI **MQCMIT** e **MQBACK**.

  È necessario anche collegare - modificare il modulo ATRSCSS dalla libreria SYS1.CSSLIB con l'applicazione. Se si utilizzano le chiamate MQI **MQCMIT** e **MQBACK**, si riceverà il codice di ritorno `MQRC_ENVIRONMENT_ERROR`.

CSQBRRSI
: Questo stub consente di utilizzare chiamate MQI **MQCMIT** e **MQBACK** ; IBM MQ implementa effettivamente queste chiamate come chiamate RRS **SRRCMIT** e **SRRBACK** .

Per informazioni sulla generazione di programmi applicativi che utilizzano l'adattatore RRS, consultare L'adattatore batch RRS.

Dove trovare ulteriori informazioni sugli adattatori z/OS Batch, TSO e RRS

È possibile trovare ulteriori informazioni sugli argomenti in questa sezione nelle seguenti fonti:

Tabella 25. Dove trovare ulteriori informazioni sull'utilizzo di z/OS Batch con IBM MQ	
Argomento	**Dove cercare**
Impostazione degli adattatori batch	Attività 19: impostazione di adattatori Batch, TSO e RRS
Servizi di ripristino delle risorse richiamabili RRS	*Programmazione MVS: servizi richiamabili per lingue di alto livello*

z/OS IBM MQ for z/OS e WebSphere Application Server

Utilizzare questo argomento per comprendere l'utilizzo di IBM MQ for z/OS da parte di WebSphere Application Server.

Le applicazioni scritte in Java in esecuzione in WebSphere Application Server possono utilizzare la specifica Java Message Service (JMS) per eseguire la messaggistica. La messaggistica point - to - point in questo ambiente può essere fornita da un gestore code IBM MQ for z/OS .

Un vantaggio dell'utilizzo di un gestore code IBM MQ for z/OS per fornire la messaggistica è che la connessione delle applicazioni JMS può partecipare pienamente alla funzionalità di una rete IBM MQ . Ad esempio, è possibile utilizzare il bridge IMS o scambiare messaggi con gestori code in esecuzione su altre piattaforme.

Connessione tra WebSphere Application Server e un gestore code

Per ulteriori informazioni, consultare Utilizzo congiunto di IBM MQ e WebSphere Application Server .

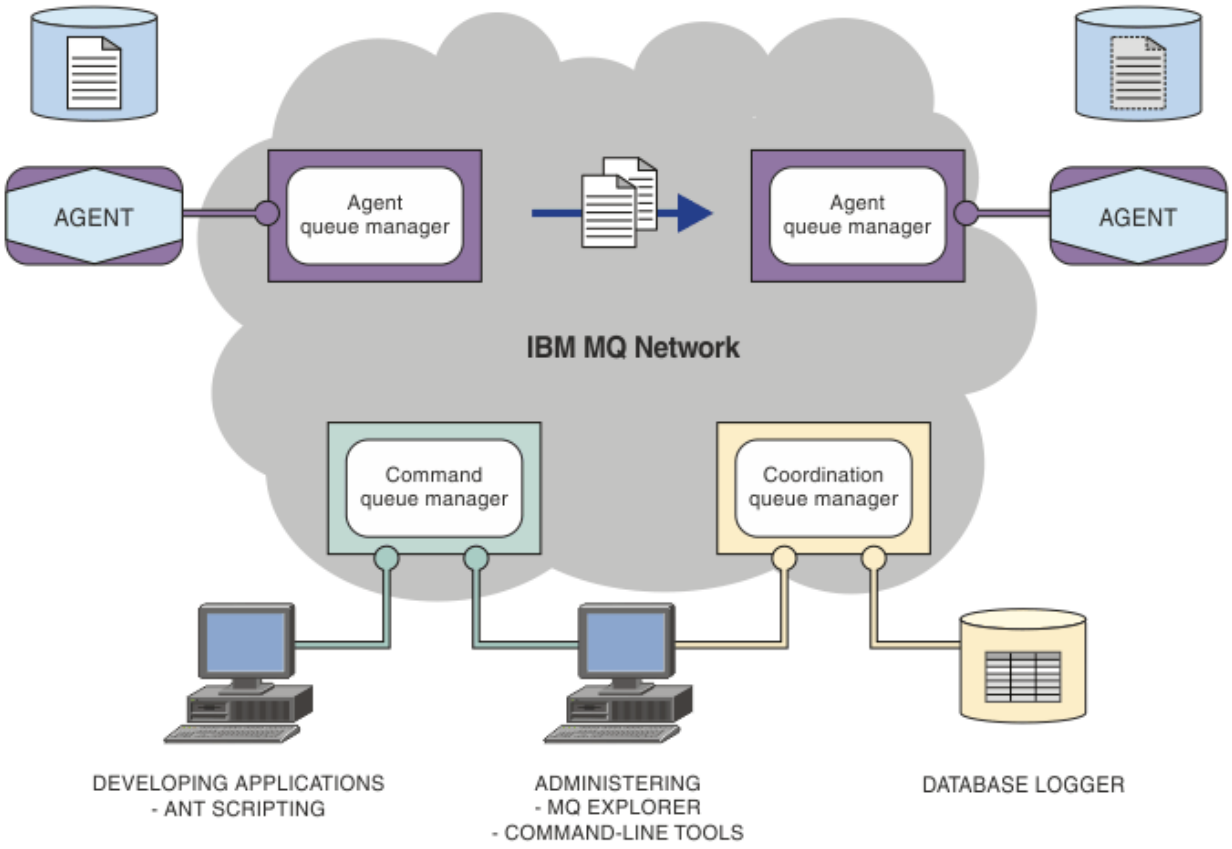
Utilizzo delle funzioni IBM MQ dalle applicazioni JMS

Per default, i messaggi JMS conservati nelle code IBM MQ utilizzano un'intestazione MQRFH2 per contenere alcune informazioni sull'intestazione del messaggio JMS . Molte applicazioni IBM MQ legacy non possono elaborare messaggi con queste intestazioni e richiedono intestazioni proprie, ad esempio MQCIH per CICS Bridge o MQWIH per IBM MQ Workflow. Per ulteriori dettagli su queste considerazioni speciali, consultare Mappatura dei messaggi JMS sui messaggi IBM MQ.

Managed File Transfer

Managed File Transfer trasferisce i file tra i sistemi in modo gestito e controllabile, indipendentemente dalla dimensione del file o dai sistemi operativi utilizzati.

È possibile utilizzare Managed File Transfer per creare una soluzione personalizzata, scalabile e automatizzata che consente di gestire, rendere attendibili e proteggere i trasferimenti file. Managed File Transfer elimina costose ridondanze, riduce i costi di manutenzione e massimizza gli investimenti IT esistenti.



Il diagramma mostra una topologia Managed File Transfer semplice. Ci sono due agenti, ognuno dei quali si connette al proprio gestore code dell'agente in una rete IBM MQ . Un file viene trasferito dall'agente su un lato del diagramma, tramite la rete IBM MQ , all'agente sull'altro lato. Inoltre, nella rete IBM MQ sono presenti il gestore code di coordinamento e un gestore code comandi. Le applicazioni e gli strumenti si collegano a questi gestori code per configurare, amministrare, operare e registrare l'attività Managed File Transfer nella rete IBM MQ .

Managed File Transfer può essere installato come quattro diverse opzioni, a seconda del proprio sistema operativo e della configurazione generale. Queste opzioni sono Managed File Transfer Agent, Managed File Transfer Logger, Managed File Transfer Serviceo Managed File Transfer Tools. Per ulteriori informazioni, consultare Opzioni del prodotto Managed File Transfer.

È possibile utilizzare Managed File Transfer per eseguire queste attività:

- Crea trasferimenti file gestiti
  - Linux Windows Creare nuovi trasferimenti file da IBM MQ Explorer su piattaforme Linux o Windows .
  - Creare nuovi trasferimenti file dalla riga comandi su tutte le piattaforme supportate.
  - Integrare la funzione di trasferimento file nello strumento Apache Ant .
  - Scrivere le applicazioni che controllano Managed File Transfer inserendo i messaggi nelle code comandi dell'agent.
  - Pianificare i trasferimenti di file da eseguire in un secondo momento. È anche possibile attivare trasferimenti di file pianificati in base a un intervallo di eventi del file system, ad esempio un nuovo file creato.
  - Monitorare continuamente una risorsa, ad esempio una directory, e quando il suo contenuto soddisfa alcune condizioni predefinite, avviare un'attività. Questa attività può essere un trasferimento file, uno script Ant o un lavoro JCL.

  - Trasferire i file da e verso le code IBM MQ .
  - Trasferire i file da e verso i server FTP, FTPS o SFTP.
  - Trasferire i file da e verso i nodi Connect:Direct .
  - Trasferire sia il testo che i file binari. I file di testo vengono convertiti automaticamente tra le codepage e le convenzioni di fine riga dei sistemi di origine e di destinazione.
  - I trasferimenti possono essere protetti, utilizzando gli standard industriali per le connessioni basate su SSL (Secure Socket Layer).
- Visualizzare i trasferimenti in corso e registrare le informazioni su tutti i trasferimenti nella rete
  - 

Visualizzare lo stato dei trasferimenti in corso da IBM MQ Explorer su piattaforme Linux o Windows .
  - 

Verificare lo stato dei trasferimenti completati utilizzando IBM MQ Explorer su piattaforme Linux o Windows .
  - Utilizzare la funzione del programma di registrazione database Managed File Transfer per salvare i messaggi di log in un database Db2 o Oracle .

Managed File Transfer è costruito su IBM MQ, che fornisce una consegna certa e unica dei messaggi tra le applicazioni. È possibile usufruire di varie funzionalità di IBM MQ. Ad esempio, è possibile utilizzare la compressione del canale per comprimere i dati che vengono inviati tra gli agent sui canali IBM MQ e utilizzare i canali SSL per proteggere i dati che vengono inviati tra gli agent. I file vengono trasferiti in modo affidabile e possono tollerare il malfunzionamento dell'infrastruttura su cui viene effettuato il trasferimento file. Se si verifica un'interruzione di rete, il trasferimento file viene riavviato dal punto in cui è stato lasciato quando è stata ripristinata la connettività.

Consolidando il trasferimento file con la tua rete IBM MQ esistente, puoi evitare di spendere le risorse necessarie per mantenere due infrastrutture separate. Se non sei già un cliente IBM MQ , creando una IBM MQ rete di supporto Managed File Transfer stai creando il backbone per una futura implementazione SOA. Se sei già un cliente IBM MQ , Managed File Transfer può trarre vantaggio dalla tua infrastruttura IBM MQ esistente, inclusi IBM MQ Internet Pass-Thru e IBM Integration Bus.

Puoi sfruttare le soluzioni di alta disponibilità IBM MQ per migliorare la resilienza della tua configurazione Managed File Transfer . Se gli agent utilizzano gestori code di dati replicati (RDQM), è necessario configurarli per utilizzare la funzione di indirizzo IP mobile. Ciò significa che gli agenti utilizzano lo stesso indirizzo IP per comunicare con una delle tre istanze RDQM attualmente in esecuzione e si riconnettono automaticamente in caso di failover (consultare Alta disponibilità RDQM e Creazione ed eliminazione di un indirizzo IP mobile). Se si utilizza la soluzione del gestore code a più istanze, le applicazioni utilizzano un indirizzo IP differente per comunicare con ciascuna istanza, gestita dalla riconnessione client in caso di failover (consultare Gestori code a più istanze e Riconnessione canale e client).

Managed File Transfer si integra con numerosi altri prodotti IBM :

IBM Integration Bus

Elaborare i file che sono stati trasferiti da Managed File Transfer come parte di un flusso IBM Integration Bus . Per ulteriori informazioni, consultare Utilizzo di MFT da IBM Integration Bus.

IBM Sterling Connect:Direct

Trasferire i file a e da una rete Connect:Direct esistente utilizzando il bridge Managed File Transfer Connect:Direct . Per ulteriori informazioni, consultare Il bridge Connect:Direct.

IBM Tivoli Composite Application Manager

IBM Tivoli Composite Application Manager fornisce un agent che è possibile utilizzare per monitorare le informazioni pubblicate sul gestore code di coordinamento.

Concetti correlati

Opzioni del prodotto Managed File Transfer

“Panoramica della topologia MFT” a pagina 299

Una panoramica su come gli agenti Managed File Transfer sono connessi al gestore code di coordinamento in una rete IBM MQ .

“Come funziona MFT con IBM MQ?” a pagina 299
Managed File Transfer interagisce in diversi modi con IBM MQ.

Come funziona MFT con IBM MQ?

Managed File Transfer interagisce in diversi modi con IBM MQ.

- Managed File Transfer trasferisce i file tra i processi dell'agent dividendo ciascun file in uno o più messaggi e trasmettendo i messaggi attraverso la rete IBM MQ .
- L'agent elabora lo spostamento dei dati del file utilizzando messaggi non persistenti per ridurre al minimo l'impatto sui log IBM MQ . Comunicando tra loro, i processi dell'agent regolano il flusso di messaggi contenenti i dati del file. Ciò impedisce la creazione di messaggi contenenti dati di file sulle code di trasmissione IBM MQ e garantisce che se uno qualsiasi dei messaggi non persistenti non viene consegnato, i dati di file vengono inviati di nuovo.
- Gli agent Managed File Transfer utilizzano un certo numero di code IBM MQ . Per ulteriori informazioni, consultare MFT code di sistema e l'argomento di sistema.
- Sebbene alcune di queste code siano strettamente per uso interno, un agent può accettare le richieste sotto forma di messaggi di comando specialmente formattati inviati a una coda specifica da cui l'agent legge. Sia i comandi della riga comandi che il plugin IBM MQ Explorer inviano IBM MQ messaggi all'agente per indicare all'agente di eseguire l'azione desiderata. È possibile scrivere le applicazioni IBM MQ che interagiscono con l'agent in questo modo. Per ulteriori informazioni, consultare Controllo MFT inserendo i messaggi nella coda comandi dell'agent.
- Gli agent Managed File Transfer inviano informazioni relative al loro stato, all'avanzamento e al risultato dei trasferimenti a un gestore code MQ designato come gestore code di coordinamento. Queste informazioni vengono pubblicate dal gestore code di coordinamento e possono essere sottoscritte dalle applicazioni che desiderano monitorare l'avanzamento del trasferimento o conservare i record dei trasferimenti che si sono verificati. Sia i comandi della riga comandi che il plug-in IBM MQ Explorer possono utilizzare le informazioni pubblicate. È possibile scrivere applicazioni IBM MQ che utilizzano queste informazioni. Per ulteriori informazioni sull'argomento in cui vengono pubblicate le informazioni, consultare SYSTEM.FTE ArgomentoFTE.
- I componenti chiave di Managed File Transfer sfruttano la funzione dei gestori code IBM MQ per memorizzare e inoltrare i messaggi. Ciò significa che se si verifica un'interruzione, le parti non interessate della propria infrastruttura possono continuare a trasferire i file. Ciò si estende al gestore code di coordinamento, dove una combinazione di sottoscrizioni di archiviazione e inoltro e durevoli consente al gestore code di coordinazione di tollerare che diventi non disponibile senza perdere le informazioni chiave sui trasferimenti file che hanno avuto luogo.

Panoramica della topologia MFT

Una panoramica su come gli agenti Managed File Transfer sono connessi al gestore code di coordinamento in una rete IBM MQ .

Gli agent Managed File Transfer inviano e ricevono i file trasferiti. Ogni agent ha la sua serie di code sul suo gestore code associato e l'agent è collegato al suo gestore code in modalità bind o client. Un agent può anche utilizzare il gestore code di coordinamento come proprio gestore code.

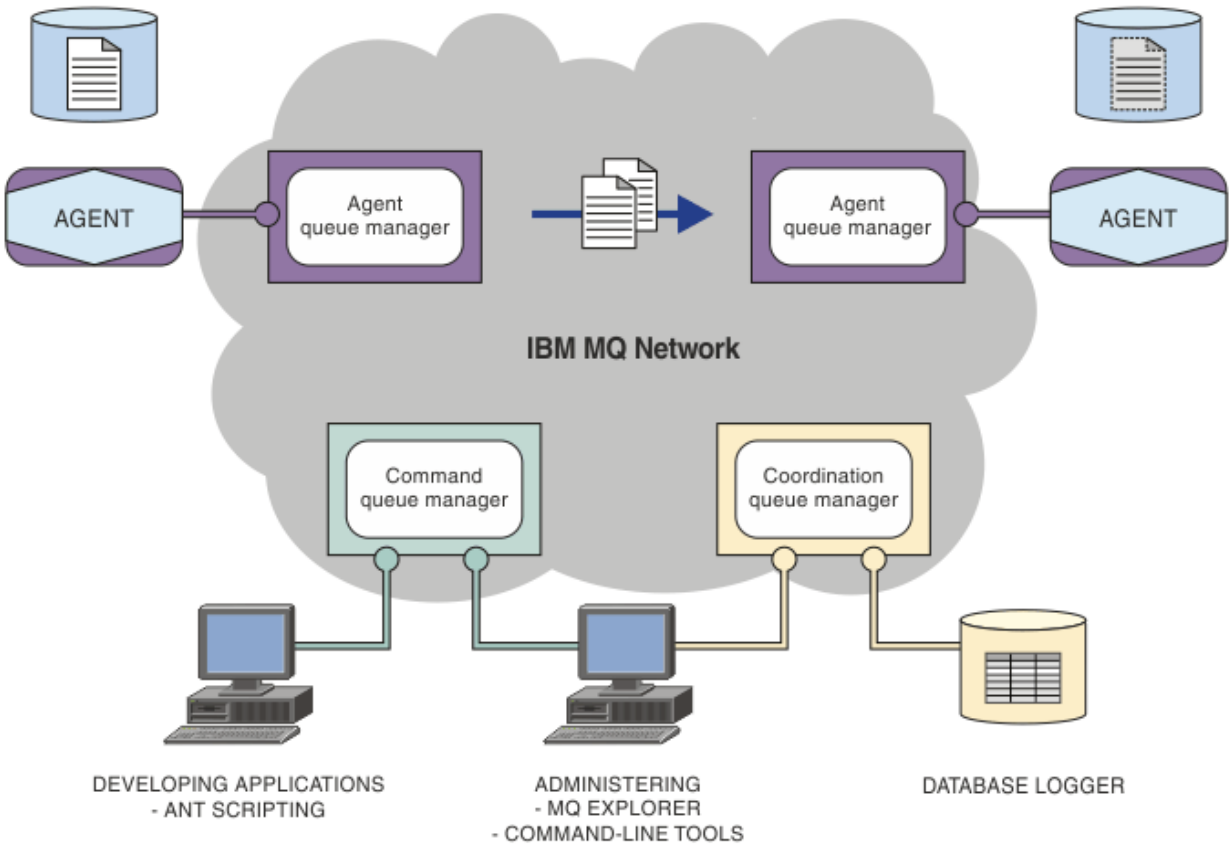
Il gestore code di coordinamento trasmette le informazioni di controllo e di trasferimento file. Il gestore code di coordinamento rappresenta un singolo punto per la raccolta di informazioni sull'agent, sullo stato del trasferimento e sul controllo del trasferimento. Il gestore code di coordinamento non deve essere disponibile per eseguire i trasferimenti. Se il gestore code di coordinamento diventa temporaneamente non disponibile, i trasferimenti continuano normalmente. I messaggi di controllo e stato vengono memorizzati nei gestori code dell'agent fino a quando il gestore code di coordinamento non diventa disponibile e possono essere elaborati normalmente.

Gli agent si registrano con il gestore code di coordinamento e pubblicano i loro dettagli su tale gestore code. Queste informazioni sull'agente vengono usate dal plugin Managed File Transfer per abilitare

l'inizio dei trasferimenti da IBM MQ Explorer. Le informazioni sull'agent raccolte sul gestore code di coordinamento vengono utilizzate dai comandi per visualizzare le informazioni sull'agent e lo stato dell'agent.

Lo stato del trasferimento e le informazioni di controllo del trasferimento vengono pubblicate sul gestore code di coordinamento. Lo stato del trasferimento e le informazioni di controllo del trasferimento vengono utilizzati dal plugin Managed File Transfer per monitorare l'avanzamento dei trasferimenti da IBM MQ Explorer. Le informazioni di controllo trasferimento memorizzate sul gestore code di coordinamento possono essere conservate per fornire la possibilità di controllo.

Il gestore code comandi viene utilizzato per connettersi alla rete IBM MQ ed è il gestore code a cui si è connessi quando si immettono i comandi Managed File Transfer .



Concetti correlati

“Managed File Transfer” a pagina 296
Managed File Transfer trasferisce i file tra i sistemi in modo gestito e controllabile, indipendentemente dalla dimensione del file o dai sistemi operativi utilizzati.

“Come funziona MFT con IBM MQ?” a pagina 299
Managed File Transfer interagisce in diversi modi con IBM MQ.

Managed File Transfer scenario

Panoramica su MFTREST API

REST API supporta alcuni comandi Managed File Transfer , incluso l'elencazione dei trasferimenti e i dettagli sugli agent di trasferimento file.

Da IBM MQ 9.1.0, REST API include le opzioni per elencare tutti i trasferimenti Managed File Transfer correnti e per interrogare lo stato degli agenti Managed File Transfer . Per ulteriori informazioni, vedi Introduzione a REST API MFT.

IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru (MQIPT) è un componente facoltativo di IBM MQ che può essere utilizzato per implementare soluzioni di messaggistica tra siti remoti su Internet.

Da IBM MQ 9.2.0, MQIPT è un componente facoltativo di IBM MQ. Per ottenere i file di installazione MQIPT per IBM MQ 9.3.x, andare all'indirizzo IBM Fix Central per IBM MQ. Prima di IBM MQ 9.2.0, MQIPT era disponibile come pacchetto di supporto.

Non è necessario eseguire IBM MQ 9.3 per utilizzare MQIPT in IBM MQ 9.3. È possibile utilizzare MQIPT per collegare qualsiasi versione supportata di IBM MQe non è necessario installare altri componenti IBM MQ alla stessa versione di MQIPT.

Se è stata acquistata la titolarità IBM MQ , è possibile installare tutte le copie richieste da MQIPT.
Le installazioni MQIPT non vengono conteggiate rispetto alla titolarità IBM MQ acquistata. Per ulteriori informazioni sulla licenza IBM MQ , consultare IBM MQ informazioni sulla licenza.

Nota: Questa documentazione è relativa a MQIPT in IBM MQ 9.3. Per la documentazione relativa al pacchetto di supporto MQIPT (versione 2.1) in IBM Documentation, consultare MQIPT (SupportPac MS81) nella documentazione IBM MQ 9.0 .

Nota: Se si utilizza MQIPT 2.1 o precedente, si consiglia di eseguire l'aggiornamento a MQIPT per IBM MQ 9.3, poiché la data di fine del supporto per il pacchetto di supporto MQIPT era il 30th settembre 2020.

IBM MQ Internet Pass-Thru viene eseguito come un servizio autonomo che può ricevere e inoltrare flussi di messaggi IBM MQ , tra due gestori code IBM MQ o tra un client IBM MQ e un gestore code IBM MQ .

MQIPT abilita questa connessione quando client e server non si trovano sulla stessa rete fisica.

Una o più istanze di MQIPT possono essere collocate nel percorso di comunicazione tra due gestori code IBM MQ o tra un client IBM MQ e un gestore code IBM MQ . Le istanze di MQIPT consentono ai due sistemi IBM MQ di scambiare messaggi senza la necessità di una connessione TCP/IP diretta tra i due sistemi. Ciò è utile se la configurazione del firewall non consente una connessione TCP/IP diretta tra i due sistemi.

MQIPT è in ascolto su una o più porte TCP/IP per le connessioni in entrata, che possono trasportare i normali messaggi IBM MQ , IBM MQ i messaggi con tunneling all'interno di HTTPo i messaggi codificati utilizzando TLS (Transport Layer Security) o SSL (Secure Sockets Layer). MQIPT può gestire più connessioni simultanee.

Si fa riferimento al canale IBM MQ che effettua la richiesta di connessione TCP/IP iniziale come *chiamante*, al canale a cui sta tentando di connettersi come *risponditore* e al gestore code che sta tentando di contattare come *gestore code di destinazione*.

MQIPT conserva i dati in memoria mentre li inoltra dall'origine alla destinazione. Nessun dato viene salvato su disco (tranne che per la memoria impaginata su disco dal sistema operativo). L'unico momento in cui MQIPT accede esplicitamente al disco è la lettura del relativo file di configurazione e la scrittura dei record di traccia e di log di connessione.

La gamma completa di tipi di canale IBM MQ può essere effettuata attraverso una o più istanze di MQIPT. La presenza di MQIPT in un percorso di comunicazione non ha alcun effetto sulle caratteristiche funzionali dei componenti IBM MQ connessi, ma potrebbe avere qualche effetto sulle prestazioni del trasferimento del messaggio.

MQIPT può essere utilizzato insieme a IBM MQ e IBM Integration Bus, come descritto in “Possibili configurazioni di MQIPT” a pagina 305.

Per installare MQIPT, consultare Installazione MQIPT.

Attività correlate

Configurazione di IBM MQ Internet Pass-Thru

Amministrazione e configurazione di IBM MQ Internet Pass-Thru

Riferimenti correlati

IBM MQ Internet Pass-Thru Riferimento di configurazione

Utilizzi di MQIPT

Esistono diversi utilizzi potenziali per IBM MQ Internet Pass-Thru (MQIPT).

MQIPT può essere utilizzato come concentratore di canali

Utilizzando MQIPT in questo modo, i canali verso o da più host separati possono apparire a un firewall come se fossero tutti verso o dall'host MQIPT . Ciò semplifica la definizione e la gestione delle regole di filtro del firewall.

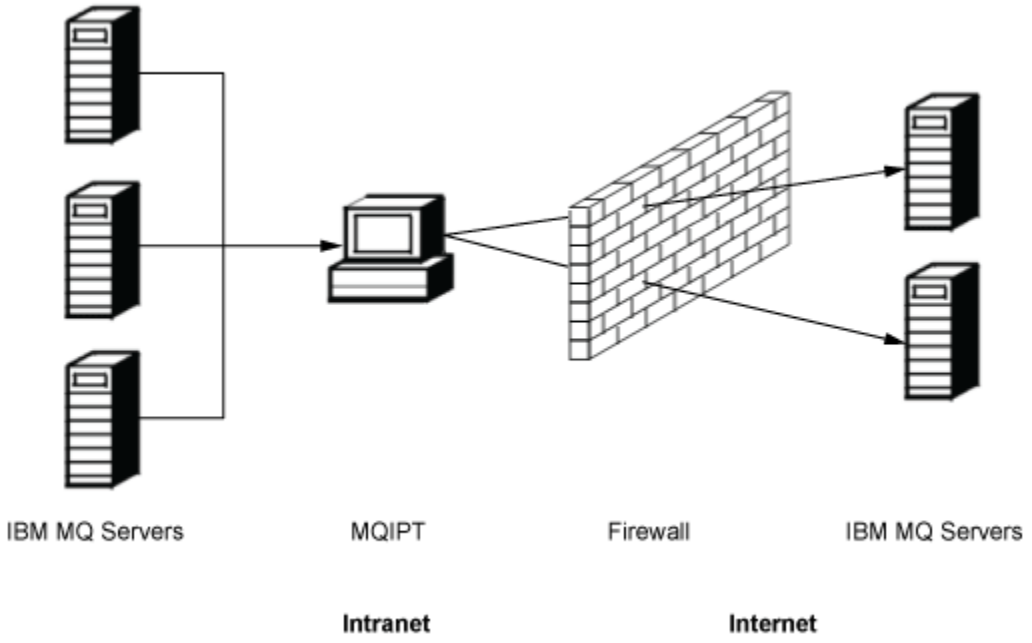


Figura 79. Esempio di MQIPT come concentratore di canale

MQIPT può essere inserito in una DMZ per fornire un singolo punto di accesso

Se MQIPT si trova all'interno di un firewall DMZ (una configurazione del firewall per la protezione delle reti locali), su un computer con un indirizzo IP (Internet Protocol) noto e attendibile, MQIPT può essere utilizzato per ascoltare le connessioni del canale IBM MQ in entrata che può quindi inoltrare all'intranet attendibile; il firewall interno deve consentire a questo computer attendibile di effettuare connessioni in entrata. In questa configurazione, MQIPT impedisce alle richieste esterne di accesso di ricevere i veri indirizzi IP dei computer nell'intranet attendibile. In questo modo, MQIPT fornisce un singolo punto di accesso. Se richiesto, MQIPT può essere configurato per accettare connessioni TLS e inoltrare i dati alla destinazione utilizzando una connessione TLS separata, terminando quindi la sessione TLS nella DMZ.

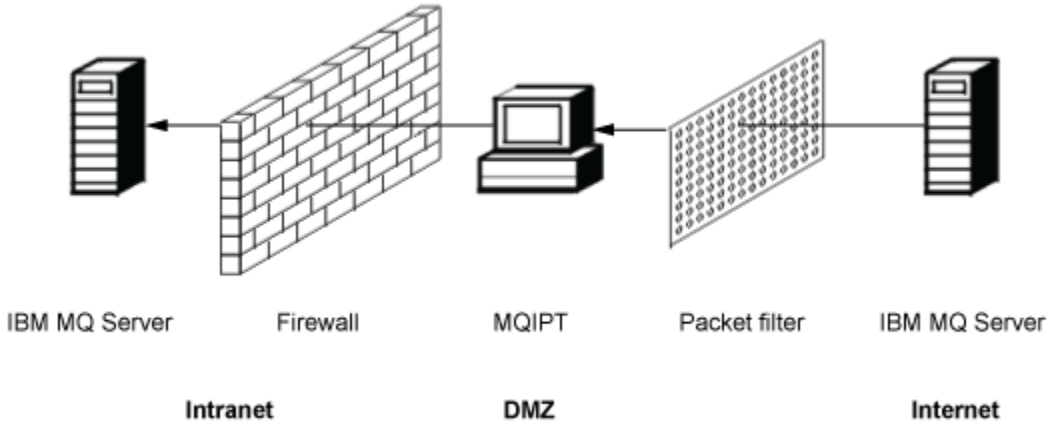


Figura 80. Esempio di MQIPT in un firewall DMZ

MQIPT può comunicare mediante il tunneling HTTP

Se due istanze di MQIPT sono distribuite in linea, possono comunicare utilizzando HTTP. La funzione di tunneling HTTP abilita la trasmissione delle richieste attraverso i firewall, utilizzando i proxy HTTP esistenti. Il primo MQIPT inserisce il protocollo IBM MQ in HTTP e il secondo estrae il protocollo IBM MQ dal relativo wrapper HTTP e lo inoltra al gestore code di destinazione.

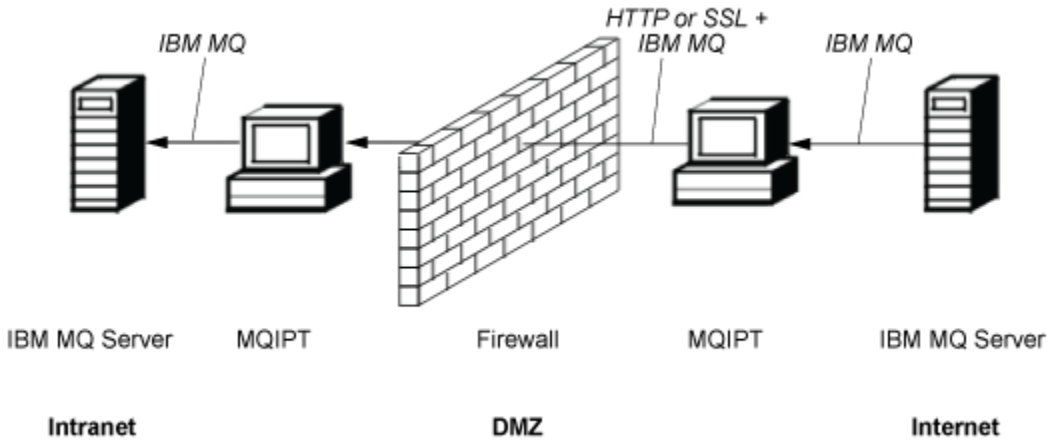


Figura 81. Esempio di tunneling MQIPT e HTTP

MQIPT può codificare i messaggi

Se MQIPT è configurato come nell'esempio precedente, le richieste possono essere codificate prima della trasmissione attraverso i firewall. Il primo MQIPT codifica i dati e il secondo li decodifica utilizzando SSL/TLS prima di inviarli al gestore code di destinazione.

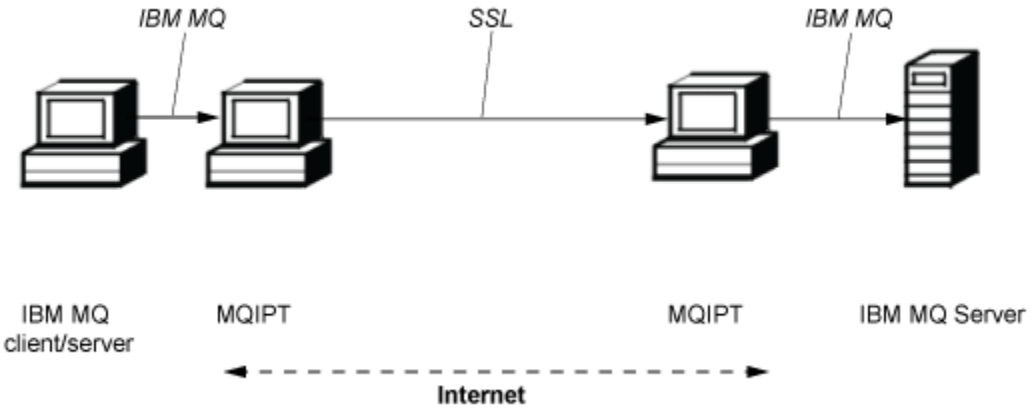


Figura 82. Esempio di MQIPT e SSL/TLS

Come funziona MQIPT

Nella sua configurazione più semplice, MQIPT agisce come un programma di inoltro del protocollo IBM MQ . È in ascolto su una porta TCP/IP e accetta richieste di connessione da canali IBM MQ .

Se viene ricevuta una richiesta con formato corretto, MQIPT stabilisce un'altra connessione TCP/IP tra se stessa e il gestore code IBM MQ di destinazione. Quindi, passa tutti i pacchetti di protocollo che riceve dalla connessione in entrata al gestore code di destinazione e restituisce i pacchetti di protocollo dal gestore code di destinazione alla connessione in entrata originale.

Non è implicata alcuna modifica al protocollo IBM MQ (client/server o gestore code su gestore code) poiché nessuna delle due estremità è direttamente consapevole della presenza dell'intermediario. Le nuove versioni del codice client o server IBM MQ non sono richieste.

Per utilizzare MQIPT, il canale chiamante deve essere configurato in modo da utilizzare il nome host e la porta MQIPT , non il nome host e la porta del gestore code di destinazione. Questo è definito con la proprietà **CONNAME** del canale IBM MQ . MQIPT legge i dati in entrata e li trasmette semplicemente al gestore code di destinazione. Altri campi di configurazione, come l'ID utente e la password in un canale client/server, vengono passati in modo simile al gestore code di destinazione.

Più gestori code

MQIPT può essere utilizzato per consentire l'accesso a più di un gestore code di destinazione. Perché ciò funzioni, deve esistere un meccanismo per indicare a MQIPT a quale gestore code connettersi, quindi MQIPT utilizza il numero di porta TCP/IP in entrata per stabilire a quale gestore code connettersi.

È quindi possibile configurare MQIPT in modo che sia in ascolto su più porte TCP/IP. Ogni porta in ascolto viene associata a un gestore code di destinazione tramite un MQIPT *instradamento*. È possibile definire fino a 100 instradamenti di questo tipo, che associano una porta TCP/IP in ascolto al nome host e alla porta del gestore code di destinazione. Ciò significa che il nome host (indirizzo IP) del gestore code di destinazione non è mai visibile al canale di origine. Ogni instradamento può gestire più connessioni tra la sua porta di ascolto e la destinazione, ogni connessione agisce in modo indipendente.

MQIPT file di configurazione

MQIPT utilizza un file di configurazione denominato `mqipt.conf`. Questo file contiene le definizioni di tutti gli instradamenti e le relative proprietà associate. Consultare Amministrazione e configurazione di IBM MQ Internet Pass-Thru per ulteriori informazioni su `mqipt.conf`.

Quando MQIPT viene avviato, avvia ogni instradamento elencato nel file di configurazione. I messaggi vengono scritti sulla console di sistema mostrando lo stato di ciascun instradamento. Quando viene

visualizzato il messaggio MQCPI078 per un instradamento, tale instradamento è pronto ad accettare richieste di connessione.

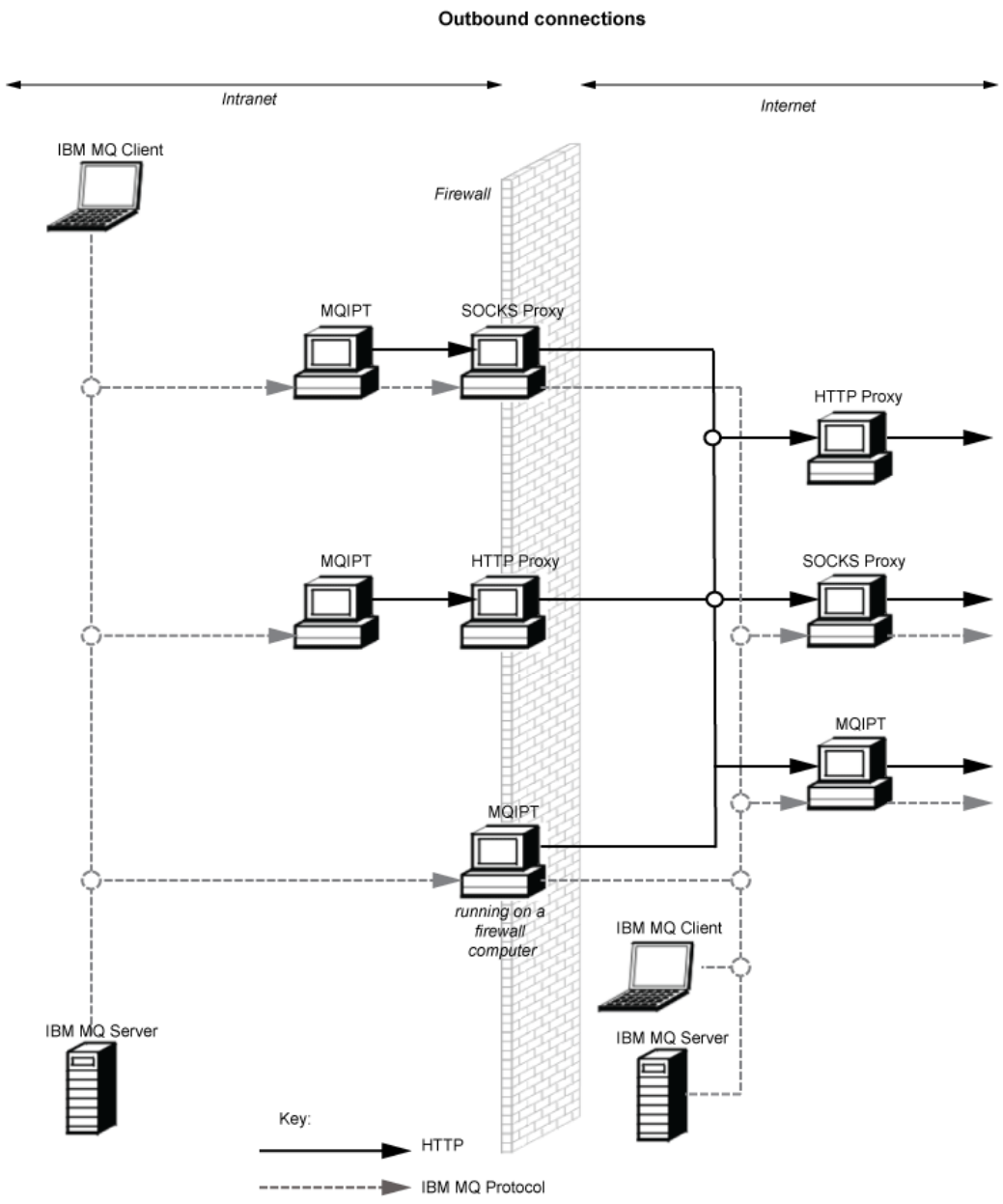
Possibili configurazioni di MQIPT

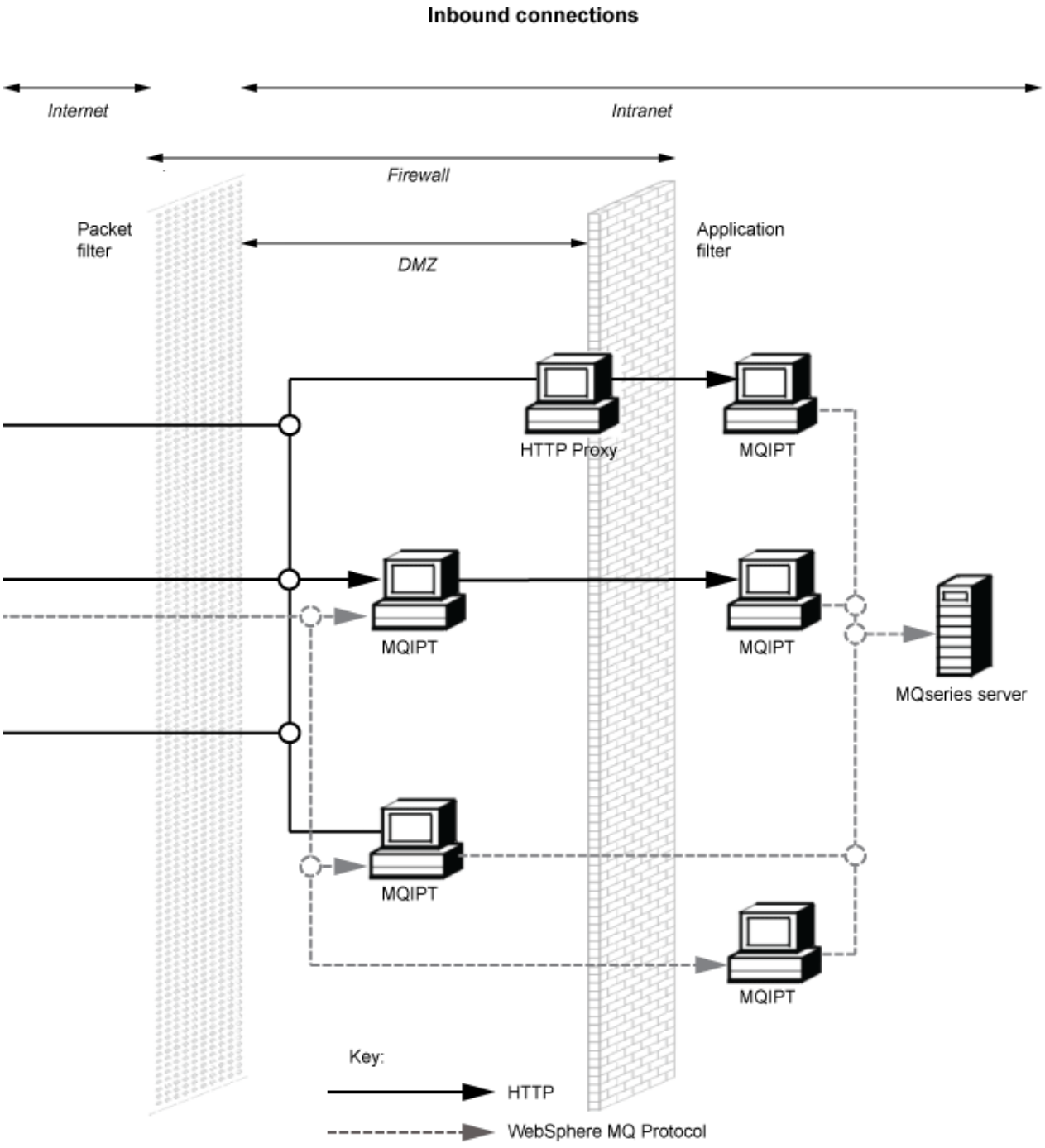
MQIPT può essere utilizzato insieme a IBM MQ e IBM Integration Bus.

La seguente figura a più parti mostra molte delle possibili configurazioni per MQIPT in una topologia IBM MQ . Illustra diversi modi in cui MQIPT può inviare messaggi. Mostra i client e i server su una intranet, all'interno di un firewall e su Internet all'esterno del firewall, passando i messaggi al proxy MQIPT, HTTP o SOCKS, che li inoltra.

I messaggi vengono ricevuti da un proxy MQIPT o da un proxy HTTP in una DMZ prima di passare il messaggio attraverso il firewall in ingresso a un server.

Si noti che il proxy HTTP , il proxy SOCKS e i computer MQIPT sul lato intranet del firewall rappresentano la possibilità di più computer concatenati su Internet. Ad esempio, un computer MQIPT potrebbe comunicare tramite uno o più computer proxy SOCKS o HTTP o ulteriori computer MQIPT , prima di raggiungere la destinazione.


Outbound connections
Intranet
Internet
IBM MQ Client
Firewall
MQIPT
SOCKS Proxy
HTTP Proxy
MQIPT
HTTP Proxy
SOCKS Proxy
MQIPT
MQIPT
running on a firewall computer
IBM MQ Client
IBM MQ Server
IBM MQ Server
Key:
HTTP
IBM MQ Protocol



Configurazioni compatibili

Scenari di connessione compatibili in cui un client IBM MQ o un gestore code comunica con MQIPT. Lo stesso instradamento o un secondo instradamento MQIPT viene utilizzato per comunicare con un gestore code di destinazione.

Configurazioni compatibili con un singolo instradamento MQIPT

Puoi utilizzare una singola rotta MQIPT per comunicare con IBM MQ.

Le colonne in Tabella 26 a pagina 308 contengono le seguenti informazioni:

1. Il protocollo utilizzato tra IBM MQ e l'instradamento MQIPT . La connessione può essere creata da un client IBM MQ o da un gestore code e può utilizzare FAP (IBM MQ Formats and Protocols) o un protocollo SSL/TLS.
2. La modalità in cui opera l'instradamento MQIPT . Il formato della comunicazione su Internet tra MQIPT e IBM MQ, è determinato dalla configurazione dell'instradamento MQIPT . Nota che dove la tabella cita SSL, puoi anche utilizzare TLS.
3. Il protocollo utilizzato tra l'instradamento MQIPT e il gestore code di destinazione.

Tabella 26. Configurazioni valide con una singola istanza di MQIPT

1. IBM MQ Protocollo di origine	**2. Modalità dell' MQIPT instradamento**	**3. IBM MQ Protocollo di destinazione**
FAP	FAP - proxy (predefinito)	FAP
	Server FAP e client SSL	SSL/TLS
SSL/TLS	Proxy SSL	SSL/TLS
	Server SSL e client FAP	FAP
	Server SSL e client SSL	SSL/TLS

Configurazioni compatibili con più di un instradamento MQIPT

Puoi scegliere di utilizzare più di una rotta, su una o più istanze di MQIPT, per comunicare con IBM MQ.

Le colonne in Tabella 27 a pagina 309 contengono le seguenti informazioni:

1. Il protocollo utilizzato tra IBM MQ e il primo instradamento MQIPT . La connessione può essere creata da un client IBM MQ o da un gestore code e può utilizzare FAP (IBM MQ Formats and Protocols) o un protocollo SSL/TLS.
2. La modalità in cui opera il primo instradamento MQIPT . Il formato della comunicazione su Internet tra MQIPT e IBM MQ, è determinato dalla configurazione dell'instradamento MQIPT . Nota che dove la tabella cita SSL, puoi anche utilizzare TLS.
3. La modalità in cui opera il secondo instradamento MQIPT .
4. Il protocollo utilizzato tra il secondo instradamento MQIPT e il gestore code di destinazione.

Tabella 27. Configurazioni valide con più istanze di MQIPT			
1. IBM MQ Protocollo di origine	**2. Modalità del primo MQIPT instradamento**	**3. Modalità del secondo MQIPT instradamento**	**4. IBM MQ Protocollo di destinazione**
FAP (predefinito)	FAP - proxy (predefinito)	FAP - proxy (predefinito)	FAP
	Server FAP e client SSL	Proxy SSL	SSL/TLS
		Server SSL e client FAP	FAP
		Server SSL e client SSL	SSL/TLS
	HTTP-client	HTTP-server e SSL - client	SSL/TLS
	HTTPS-client	HTTPS-server e client SSL	SSL/TLS
	HTTP-client	server-http	FAP
	HTTPS-client	HTTPS-server	FAP
SSL/TLS	Proxy SSL	Proxy SSL	SSL/TLS
		Server SSL e client FAP	FAP
		Server SSL e client SSL	SSL/TLS
	HTTP-client	server-http	FAP
	HTTPS-client	HTTPS-server	SSL/TLS
	HTTP-client	HTTP-server e SSL - client	FAP
	HTTPS-client	HTTPS-server e client SSL	SSL/TLS

Configurazioni canale supportate

Tutti i tipi di canale IBM MQ sono supportati, ma la configurazione è limitata alle connessioni TCP/IP. Per un client o gestore code IBM MQ , MQIPT viene visualizzato come se fosse il gestore code di destinazione. Laddove la configurazione del canale richiede un host di destinazione e un numero di porta, vengono specificati il numero di porta del listener e il nome host MQIPT .

Canali client/server

MQIPT ascolta le richieste di connessione client in entrata e le inoltra utilizzando il tunneling HTTP , SSL/TLS o come pacchetti di protocollo IBM MQ standard. Se MQIPT sta utilizzando il tunneling HTTP o SSL/TLS, li inoltra su una connessione a un secondo MQIPT. Se non utilizza il tunneling HTTP , li inoltra su una connessione a quello che vede come il gestore code di destinazione (anche se a sua volta potrebbe essere un ulteriore MQIPT). Quando il gestore code di destinazione ha accettato la connessione client, i pacchetti vengono inoltrati tra client e server.

Canali mittente / ricevente cluster

Se MQIPT riceve una richiesta in entrata da un canale mittente del cluster, presuppone che il gestore code sia stato abilitato a SOCKS e che il vero indirizzo di destinazione venga ottenuto durante il processo di handshake SOCKS. Inoltra la richiesta al successivo MQIPT o al gestore code di destinazione esattamente nello stesso modo dei canali di connessione client. Include anche i canali mittenti del cluster definiti automaticamente.

Mittente / destinatario

Se MQIPT riceve una richiesta in entrata da un canale mittente, la inoltra al successivo MQIPT o al gestore code di destinazione esattamente come per i canali di connessione client. Il gestore code di destinazione convalida la richiesta in entrata e avvia il canale ricevente, se appropriato. Tutte le comunicazioni tra il canale mittente e destinatario (inclusi i flussi di sicurezza) vengono inoltrate.

Richiedente / server
Questa combinazione viene gestita nello stesso modo delle precedenti configurazioni. La convalida della richiesta di connessione viene effettuata dal canale server sul gestore code di destinazione.

Richiedente / mittente
La configurazione di "callback" potrebbe essere utile se ai due gestori code non è consentito stabilire connessioni dirette tra loro, ma è consentito connettersi a MQIPT e accettare connessioni da esso.

Server / richiedente e server / ricevitore
Questi sono gestiti da MQIPT nello stesso modo in cui gestisce la configurazione `Sender/Receiver` .

Condizioni di terminazione e di errore del canale

Quando MQIPT rileva la chiusura (normale o anomala) di un canale IBM MQ , propaga la chiusura del canale. Se si chiude un instradamento utilizzando MQIPT, tutti i canali che attraversano tale instradamento vengono chiusi.

MQIPT fornisce una funzione di timeout di inattività facoltativa. Se MQIPT rileva che un canale è stato inattivo per un periodo di tempo superiore al timeout, esegue una chiusura immediata sulle due connessioni in questione.

I sistemi IBM MQ a entrambe le estremità del canale osservano queste condizioni di arresto anomale come errori di rete o come terminazione del canale da parte del partner. Il canale è quindi in grado di riavviare e ripristinare (se l'errore si verifica durante un periodo di dubbio del protocollo) come se MQIPT non fosse utilizzato.

Sicurezza dei messaggi

La gestione delle code distribuite IBM MQ garantisce che i messaggi vengano consegnati correttamente. Questo è ancora il caso quando MQIPT è presente tra le due estremità del canale. MQIPT non memorizza i dati del messaggio o non partecipa alla procedura del punto di sincronizzazione che garantisce la corretta consegna del messaggio.

Quando si utilizzano messaggi IBM MQ veloci, non persistenti, se l'instradamento MQIPT non riesce o viene riavviato quando un messaggio IBM MQ è in transito, il messaggio potrebbe essere perso. Prima di riavviare la rotta, assicurarsi che tutti i canali di IBM MQ che utilizzano la rotta MQIPT siano inattivi.

Per ulteriori informazioni sulla sicurezza dei messaggi in IBM MQ, consultare Sicurezza dei messaggi.

Gestori code a più istanze e alta disponibilità

MQIPT può essere utilizzato con gestori code a più istanze in ambienti ad alta disponibilità.

MQIPT non ha uno stato persistente e quindi non è possibile eseguire il failover MQIPT su un altro sistema. Disporre invece di più istanze di MQIPT con file di configurazione `mqipt.conf` identici in esecuzione su sistemi differenti. Monitorare la disponibilità di ciascuna istanza di MQIPT e riavviarla (sullo stesso sistema) se necessario. Ciò fornisce una serie di istanze MQIPT identiche che possono essere utilizzate per instradare connessioni. È necessario quindi assicurarsi che IBM MQ possa instradare le connessioni a MQIPT e che MQIPT possa inoltrare tali connessioni al gestore code di destinazione.

I canali IBM MQ in uscita possono essere indirizzati a un'istanza MQIPT disponibile in diversi modi, ad esempio:

- Utilizzare un programma di bilanciamento del carico o un router ad alta disponibilità, ad esempio IBM Network Dispatcher dal prodotto WebSphere Edge Components.
- Specificare più nomi di connessione nella definizione del canale IBM MQ utilizzando un elenco separato da virgole. IBM MQ tenta quindi di connettersi a ciascun indirizzo di MQIPT finché non trova un'istanza MQIPT disponibile.

È inoltre necessario indirizzare le connessioni da MQIPT al gestore code di destinazione. Se la configurazione ad alta disponibilità garantisce il failover dell'indirizzo IP con il gestore code di destinazione, non è richiesta alcuna configurazione MQIPT speciale: specificare l'indirizzo IP di

destinazione nella proprietà di instradamento **Destination** e consentire all'operazione di failover di spostare l'indirizzo IP con il gestore code.

Tuttavia, se l'indirizzo IP del gestore code cambia dopo un failover, è necessario fare in modo che MQIPT inoltra la connessione alla destinazione corretta. Ciò può essere fatto in uno dei modi seguenti:

- Scrivere un'uscita di instradamento che controlli quale indirizzo IP e numero di porta sono accessibili e quindi sovrascrivere la destinazione di instradamento per ogni connessione. Alcune uscite di instradamento di esempio sono fornite con MQIPT; possono essere adattate a questo scopo.
- Utilizzare un programma di bilanciamento del carico ad alta disponibilità per reindirizzare la connessione.
- Definire più instradamenti MQIPT , uno per ogni indirizzo IP e porta in cui il gestore code potrebbe essere in esecuzione. Quindi indirizza le connessioni IBM MQ ai vari instradamenti MQIPT , ad esempio elencando tutti gli indirizzi IP di instradamento e i numeri di porta in un elenco separato da virgole nel nome della connessione del canale in uscita.

È anche importante ottimizzare tutti i componenti end-to-end sul percorso di rete:

1. I tentativi di connessione ai sistemi non disponibili devono avere esito negativo immediatamente in modo che i tentativi di riconnessione possano passare alla prima destinazione disponibile.

   Per gli instradamenti MQIPT SSL, ottimizzare la proprietà di instradamento **SSLClientConnectTimeout** per assicurare un errore di connessione di prompt per destinazioni non disponibili. Fare riferimento alla documentazione IBM MQ per dettagli sui parametri di ottimizzazione IBM MQ . Inoltre, consultare la documentazione del proprio sistema operativo per dettagli sull'ottimizzazione TCP/IP per il sistema operativo. In tutti i casi, i tentativi di connessione non riusciti dovrebbero restituire rapidamente un errore di rete (ad esempio, un pacchetto di ripristino TCP) o dovrebbero andare in timeout senza ritardi ingiustificati.
2. Le connessioni attive a un sistema non riuscito devono essere interrotte prontamente in modo da poter stabilire nuove connessioni.

   È inoltre necessario considerare l'impatto di un failover in un momento in cui le connessioni utilizzano attivamente MQIPT. È probabile che le connessioni di rete vengano interrotte durante un failover. Per le applicazioni client, è possibile utilizzare la funzione di riconnessione client automatica IBM MQ per ristabilire le connessioni interrotte. Per i canali dei messaggi, è possibile specificare un breve intervallo di tentativi in modo che il canale si riconnetta prontamente. Consultare la documentazione di IBM MQ per ulteriori informazioni sulla riconnessione automatica del client e sulla configurazione dei tentativi del canale dei messaggi.

V 9.3.5 IBM MQ Console e REST API

È possibile utilizzare IBM MQ Console e REST API per gestire IBM MQed eseguire operazioni di messaggistica, utilizzando HTTP.

- È possibile utilizzare IBM MQ Console per eseguire attività di gestione di base da un browser web. Per ulteriori informazioni, consultare Amministrazione utilizzando IBM MQ Console.
- È possibile utilizzare administrative REST API per amministrare gli oggetti IBM MQ , come i gestori code e le code, gli agent e i trasferimenti Managed File Transfer . Per ulteriori informazioni, consultare Amministrazione utilizzando REST API.
- È possibile utilizzare messaging REST API per eseguire semplici operazioni point-to-point e pubblicare la messaggistica. Per ulteriori informazioni, consultare Messaggistica utilizzando REST API.

Opzioni di installazione

IBM MQ Console e REST API vengono eseguiti in un server WebSphere Liberty , denominato mqweb. Da IBM MQ 9.3.5, è possibile installare il server mqweb come componente facoltativo in un'installazione IBM MQ o come installazione IBM MQ Web Server autonoma.

V 9.3.5 Linux Installazione autonoma di IBM MQ Web Server

Da IBM MQ 9.3.5, il server mqweb può essere eseguito in una installazione autonoma di IBM MQ Web Server. Un'installazione autonoma di IBM MQ Web Server consente di installare ed eseguire il server mqweb su sistemi separati dalle installazioni di IBM MQ . L'installazione di un IBM MQ Web Server autonomo offre una maggiore flessibilità per quanto riguarda i sistemi e il numero di sistemi su cui si sceglie di eseguire i server mqweb. Se necessario, è possibile eseguire diverse istanze del server mqweb su macchine differenti per fornire la scalabilità e la disponibilità necessarie.

Se è stata acquistata la titolarità IBM MQ , è possibile installare tutte le copie richieste del IBM MQ Web Serverautonomo. Le installazioni IBM MQ Web Server non vengono conteggiate rispetto alla titolarità IBM MQ acquistata. Per ulteriori informazioni sulla licenza IBM MQ , consultare IBM MQ informazioni sulla licenza.

Le seguenti limitazioni si applicano in una installazione IBM MQ Web Server autonoma:

- IBM MQ Console può essere utilizzato per gestire solo i gestori code remoti.
- messaging REST API può essere utilizzato solo con gestori code remoti.
- administrative REST API non è disponibile.

Il IBM MQ Web Server autonomo è supportato solo su piattaforme Linux .

Per ulteriori informazioni sull'installazione del IBM MQ Web Serverautonomo, consultare Installazione del IBM MQ Web Serverautonomo.

Componente facoltativo di un'installazione IBM MQ

È possibile scegliere di installare il componente IBM MQ Console e REST API come parte di un'installazione IBM MQ .

Tutte le funzioni IBM MQ Console e REST API sono disponibili quando il server mqweb viene eseguito in un'installazione di IBM MQ .

- IBM MQ Console può essere utilizzato per gestire gestori code locali e remoti.
- messaging REST API può essere utilizzato con gestori code locali e remoti.
- administrative REST API può essere utilizzato per gestire gestori code locali e remoti.

Per utilizzare il componente IBM MQ Console e REST API , installare il seguente componente come parte dell'installazione di IBM MQ :

- AIX Su AIX, installare il fileset `mqm.web.rte` .
- IBM i Su IBM i, installare il componente WEB.
- Linux Su Linux, installare il componente `MQSeriesWeb` .
- Windows Su Windows, installare la funzione `Web Administration` .
- z/OS Su z/OS, installare la funzione `IBM MQ for z/OS UNIX System Services Web Components` .

Informazioni particolari

Queste informazioni sono state sviluppate per prodotti e servizi offerti negli Stati Uniti.

IBM potrebbe non offrire i prodotti, i servizi o le funzioni descritti in questo documento in altri paesi. Consultare il rappresentante IBM locale per informazioni sui prodotti e sui servizi disponibili nel proprio paese. Ogni riferimento relativo a prodotti, programmi o servizi IBM non implica che solo quei prodotti, programmi o servizi IBM possano essere utilizzati. In sostituzione a quelli forniti da IBM possono essere usati prodotti, programmi o servizi funzionalmente equivalenti che non comportino la violazione dei diritti di proprietà intellettuale o di altri diritti dell'IBM. Tuttavia, è responsabilità dell'utente valutare e verificare il funzionamento di qualsiasi prodotto, programma o servizio non IBM.

IBM potrebbe disporre di applicazioni di brevetti o brevetti in corso relativi all'argomento descritto in questo documento. La fornitura di tale documento non concede alcuna licenza a tali brevetti. Chi desiderasse ricevere informazioni relative a licenze può rivolgersi per iscritto a:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Per richieste di licenze relative ad informazioni double-byte (DBCS), contattare il Dipartimento di Proprietà Intellettuale IBM nel proprio paese o inviare richieste per iscritto a:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

Il seguente paragrafo non si applica al Regno Unito o a qualunque altro paese in cui tali dichiarazioni sono incompatibili con le norme locali: INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE LA PRESENTE PUBBLICAZIONE "NELLO STATO IN CUI SI TROVA" SENZA GARANZIE DI ALCUN TIPO, ESPRESSE O IMPLICITE, IVI INCLUSE, A TITOLO DI ESEMPIO, GARANZIE IMPLICITE DI NON VIOLAZIONE, DI COMMERCIABILITÀ E DI IDONEITÀ PER UNO SCOPO PARTICOLARE. Alcuni stati non consentono la rinuncia a garanzie esplicite o implicite in determinate transazioni; quindi la presente dichiarazione potrebbe non essere applicabile.

Questa pubblicazione potrebbe contenere imprecisioni tecniche o errori tipografici. Le informazioni incluse in questo documento vengono modificate su base periodica; tali modifiche vengono incorporate nelle nuove edizioni della pubblicazione. IBM si riserva il diritto di apportare miglioramenti o modifiche al prodotto/i e/o al programma/i descritti nella pubblicazione in qualsiasi momento e senza preavviso.

Qualsiasi riferimento a siti Web non IBM contenuto nelle presenti informazioni è fornito per consultazione e non vuole in alcun modo promuovere i suddetti siti Web. I materiali presenti in tali siti Web non sono parte dei materiali per questo prodotto IBM e l'utilizzo di tali siti Web è a proprio rischio.

Tutti i commenti e i suggerimenti inviati potranno essere utilizzati liberamente da IBM e diventeranno esclusiva della stessa.

Coloro che detengono la licenza su questo programma e desiderano avere informazioni su di esso allo scopo di consentire (i) uno scambio di informazioni tra programmi indipendenti ed altri (compreso questo) e (ii) l'uso reciproco di tali informazioni, dovrebbero rivolgersi a:

IBM Corporation
Coordinatore interoperabilità software, Dipartimento 49XA
Autostrada 3605 52 N

Rochester, MN 55901
U.S.A.

Queste informazioni possono essere rese disponibili secondo condizioni contrattuali appropriate, compreso, in alcuni casi, il pagamento di un addebito.

Il programma su licenza descritto in queste informazioni e tutto il materiale su licenza disponibile per esso sono forniti da IBM in base ai termini dell' IBM Customer Agreement, IBM International Program License Agreement o qualsiasi altro accordo equivalente tra le parti.

Tutti i dati relativi alle prestazioni contenuti in questo documento sono stati determinati in un ambiente controllato. Pertanto, i risultati ottenuti in altri ambienti operativi possono variare in modo significativo. Alcune misurazioni potrebbero essere state fatte su sistemi a livello di sviluppo e non vi è alcuna garanzia che queste misurazioni saranno le stesse sui sistemi generalmente disponibili. Inoltre, alcune misurazioni potrebbero essere state stimate mediante estrapolazione. I risultati quindi possono variare. Gli utenti di questo documento dovrebbero verificare i dati applicabili per il loro ambiente specifico.

Le informazioni relative a prodotti non IBM provengono dai fornitori di tali prodotti, dagli annunci pubblicati o da altre fonti pubblicamente disponibili. IBM non ha verificato tali prodotti e, pertanto, non può garantirne l'accuratezza delle prestazioni. Eventuali commenti relativi alle prestazioni dei prodotti non IBM devono essere indirizzati ai fornitori di tali prodotti.

Tutte le dichiarazioni riguardanti la direzione o l'intento futuro di IBM sono soggette a modifica o ritiro senza preavviso e rappresentano solo scopi e obiettivi.

Questa pubblicazione contiene esempi di dati e prospetti utilizzati quotidianamente nelle operazioni aziendali, Per poterli illustrare nel modo più completo possibile, gli esempi riportano nomi di persone, società, marchi e prodotti. Tutti questi nomi sono fittizi e qualsiasi somiglianza con nomi ed indirizzi adoperati da imprese realmente esistenti sono una mera coincidenza.

LICENZA SUL COPYRIGHT:

Queste informazioni contengono programmi applicativi di esempio in lingua originale, che illustrano le tecniche di programmazione su diverse piattaforme operative. È possibile copiare, modificare e distribuire questi programmi di esempio sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in conformità alle API (application programming interface) a seconda della piattaforma operativa per cui i programmi di esempio sono stati scritti. Questi esempi non sono stati testati approfonditamente tenendo conto di tutte le condizioni possibili. IBM, quindi, non può garantire o sottintendere l'affidabilità, l'utilità o il funzionamento di questi programmi.

Se si sta visualizzando queste informazioni in formato elettronico, le fotografie e le illustrazioni a colori potrebbero non apparire.

Informazioni sull'interfaccia di programmazione

Le informazioni sull'interfaccia di programmazione, se fornite, consentono di creare software applicativo da utilizzare con questo programma.

Questo manuale contiene informazioni sulle interfacce di programmazione che consentono al cliente di scrivere programmi per ottenere i servizi di WebSphere MQ.

Queste informazioni, tuttavia, possono contenere diagnosi, modifica e regolazione delle informazioni. La diagnosi, la modifica e la regolazione delle informazioni vengono fornite per consentire il debug del software applicativo.

Importante: Non utilizzare queste informazioni di diagnosi, modifica e ottimizzazione come interfaccia di programmazione poiché sono soggette a modifica.

Marchi

IBM, il logo IBM , ibm.com, sono marchi di IBM Corporation, registrati in molte giurisdizioni nel mondo. Un elenco aggiornato dei marchi IBM è disponibile sul web in "Copyright and trademark

information"www.ibm.com/legal/copytrade.shtml. Altri nomi di prodotti e servizi potrebbero essere marchi di IBM o altre società.

Microsoft e Windows sono marchi di Microsoft Corporation negli Stati Uniti, in altri paesi o entrambi.

UNIX è un marchio registrato di The Open Group negli Stati Uniti e/o in altri paesi.

Linux è un marchi registrato di Linus Torvalds negli Stati Uniti e/o in altri paesi.

Questo prodotto include il software sviluppato da Eclipse Project (https://www.eclipse.org/).

Java e tutti i marchi e i logo Java sono marchi registrati di Oracle e/o di società affiliate.

Numero parte:

(1P) P/N: