9.4

Pianificazione per IBM MQ

Nota

Prima di utilizzare queste informazioni e il prodotto che supportano, leggere le informazioni in “Informazioni particolari” a pagina 211.

Questa edizione si applica alla versione 9 release 4 di IBM® MQ e a tutte le successive release e modifiche se non diversamente indicato nelle nuove edizioni.

Indice

Pianificazione di un'architettura IBM MQ

Quando si pianifica l'ambiente IBM MQ , considerare il supporto fornito da IBM MQ per le architetture di gestori code singoli e multipli e per gli stili di messaggistica point-to-point e di pubblicazione / sottoscrizione. Inoltre, pianificare i requisiti delle risorse e l'utilizzo delle funzioni di registrazione e backup.

Informazioni su questa attività

Prima di pianificare la propria architettura IBM MQ , acquisire familiarità con i concetti di IBM MQ di base. Vedere IBM MQ Panoramica tecnica.

Le architetture IBM MQ vanno da semplici architetture che utilizzano un singolo gestore code a reti più complesse di gestori code interconnessi. Più gestori code sono connessi insieme utilizzando tecniche di accodamento distribuito. Per ulteriori informazioni sulla pianificazione di singole architetture di gestori code e di più architetture di gestori code, consultare i seguenti argomenti:

- “Architetture basate su un singolo gestore code” a pagina 18
- “Architetture basate su più gestori code” a pagina 19
  - “Pianificazione delle code e dei cluster distribuiti” a pagina 20
  - “Pianificazione della rete di pubblicazione / sottoscrizione distribuita” a pagina 72

z/OS Su IBM MQ for z/OS è possibile utilizzare le code condivise e i gruppi di condivisione code per implementare il bilanciamento del carico di lavoro e le applicazioni IBM MQ per essere scalabili e altamente disponibili. Per informazioni sulle code condivise e sui gruppi di condivisione code, consultare Code condivise e gruppi di condivisione code.

IBM MQ fornisce due diversi modelli di release:

- La release Long Term Support (LTS) è la più adatta per i sistemi che richiedono una distribuzione a lungo termine e la massima stabilità.
- La release Continuous Delivery (CD) è destinata ai sistemi che devono sfruttare rapidamente gli ultimi miglioramenti funzionali per IBM MQ.

Entrambi i tipi di release sono installati nello stesso modo, ma vi sono considerazioni relative al supporto e alla migrazione che è necessario comprendere. Per ulteriori informazioni, consultare Tipi di release e controllo delle versioni diIBM MQ.

Per informazioni sulla pianificazione di più installazioni, sui requisiti di memoria e prestazioni e sull'utilizzo dei client, consultare gli altri argomenti secondari.

Concetti correlati

Versioni e tipi di release IBM MQ

“Planning your IBM MQ environment on z/OS” a pagina 145
When planning your IBM MQ environment, you must consider the resource requirements for data sets, page sets, Db2, Coupling Facilities, and the need for logging, and backup facilities. Use this topic to plan the environment where IBM MQ runs.

Disponibilità, ripristino e riavvio

Attività correlate

Verifica dei requisiti

Verifica che i messaggi non vadano persi (registrazione)

tipi di release IBM MQ : considerazioni sulla pianificazione

I due tipi di release principali per IBM MQ sono: Long Term Support (LTS) e Continuous Delivery (CD). Per ogni piattaforma supportata, il tipo di rilascio scelto influenza l'ordine, l'installazione, la manutenzione e la migrazione.

Per informazioni dettagliate sui tipi di release, consultare IBM MQ tipi di release e controllo delle versioni.

Considerazioni per IBM MQ for Multiplatforms

Come ordinare

All'interno di Passport Advantage ci sono due eAssemblies separati per IBM MQ 9.4. Uno contiene le immagini di installazione per la release di IBM MQ 9.4.0 Long Term Support e l'altro contiene le immagini di installazione per la release di IBM MQ 9.4.x Continuous Delivery . Scaricare le immagini di installazione da eAssembly in base alla propria scelta di release.

Tutte le versioni IBM MQ e per IBM MQ 9.4 sia le release LTS che le release CD , appartengono allo stesso ID prodotto.

La titolarità ad utilizzare IBM MQ si estende all'intero prodotto (PID), in base ai vincoli dei componenti con licenza e delle metriche dei prezzi. Ciò significa che è possibile scegliere liberamente tra le immagini di installazione della release LTS e CD per IBM MQ 9.4.

Installazione

Dopo aver scaricato un'immagine di installazione da Passport Advantage, è necessario selezionare per l'installazione solo i componenti per cui è stata acquistata la titolarità. Consultare IBM MQ informazioni sulla licenza per ulteriori informazioni sui componenti installabili inclusi per ciascun componente addebitabile.

È possibile installare IBM MQ 9.4.0 LTS release e IBM MQ 9.4.x CD release sulla stessa immagine del sistema operativo. Se si esegue questa operazione, i componenti vengono visualizzati come installazioni separate, come supportato dal supporto multi - versione IBM MQ . Ciascuna versione dispone di set distinti di gestori code associati a tale versione.

Ogni nuova release di CD viene fornita come immagine di installazione. La nuova release CD può essere installata insieme a una release esistente oppure una release CD precedente può essere aggiornata dal programma di installazione alla nuova release.

Le release CD contengono miglioramenti funzionali e l'ultima serie di correzioni di difetti e aggiornamenti di sicurezza. Ogni release CD è cumulativa e sostituisce completamente tutte le release precedenti per quella versione di IBM MQ. Quindi, è possibile ignorare una release specifica di CD se non contiene alcuna funzione rilevante per la propria azienda.

Manutenzione

La release LTS viene servita dall'applicazione dei fix pack, che forniscono correzioni di difetti e dagli aggiornamenti cumulativi della sicurezza (CSU), che forniscono patch di sicurezza. I fix pack e le CSU vengono resi disponibili periodicamente e sono cumulativi.

Per CD, le CSU vengono prodotte solo per l'ultima release CD , che potrebbe essere su una versione successiva.

Occasionalmente, il team di supporto IBM potrebbe richiedere l'applicazione di una fix temporanea. Le fix temporanee sono note anche come fix di emergenza o di test e vengono utilizzate per applicare aggiornamenti urgenti che non possono attendere la successiva distribuzione della manutenzione.

Migrazione tra release LTS e release CD

Esistono dei vincoli e delle limitazioni, ma, generalmente, è possibile migrare un singolo gestore code dall'utilizzo del codice release LTS al codice release CD o dall'utilizzo del codice release CD al codice release LTS , purché la release di destinazione sia superiore a quella utilizzata prima della migrazione.

Sono possibili due approcci:

- Installare la nuova versione del codice in modo che venga aggiornata un'installazione esistente di IBM MQ . Tutti i gestori code associati all'installazione utilizzano la nuova release del codice quando vengono avviati.
- Installare la nuova release del codice come nuova installazione, quindi spostare le singole istanze del gestore code nella nuova installazione utilizzando il comando setmq .

Quando un gestore code avvia l'esecuzione di una release CD del codice, il livello di comando del gestore code viene aggiornato per indicare il nuovo livello di release. Ciò significa che tutte le nuove funzioni fornite nella release sono abilitate e che non è più possibile riavviare il gestore code utilizzando una release del codice con un numero VRM inferiore.

Considerazioni per IBM MQ for z/OS

Come ordinare

Quando si ordina IBM MQ for z/OS 9.4, su ShopZvengono offerte due funzioni separate. Le funzioni corrispondono alla release LTS e alla release CD . Entrambe le funzioni sono applicabili allo stesso ID prodotto (PID). Si tratta dell'ID prodotto concesso in licenza, quindi dove una funzione è concessa in licenza, vi è la possibilità di utilizzare la funzione alternativa, se richiesto. Quando si ordina, si seleziona la funzione corrispondente alla release LTS o CD .

Se si stanno selezionando prodotti da includere in un ServerPac, non è possibile scegliere sia la release LTS che la release CD nello stesso ordine ServerPac , in quanto i prodotti non possono essere installati da SMP/E nella stessa zona di destinazione.

Installazione

Le release LTS e CD sono fornite in serie separate di FMID. Tenere presente che questi FMID non possono essere installati nella stessa zona di destinazione SMP/E. Se sono necessarie entrambe le release LTS e CD :

- Installare la release LTS e CD in zone di destinazione separate.
- Mantenere le librerie di destinazione e di distribuzione separate per i due release.

Se il gestore code si trova in un gruppo di condivisione code, quando si esegue l'aggiornamento all'ultima versione del CD è necessario aggiornare tutti i gestori code del gruppo.

Il livello di comando di un gestore code è il livello VRM a tre cifre. Un programma IBM MQ può chiamare `MQINQ`, passando il selettore `MQIA_COMMAND_LEVEL` , per richiamare il livello di comando del gestore code a cui è connesso.

Poiché le release utilizzano FMID differenti, non è possibile aggiornare una release CD con la manutenzione per una release LTS o viceversa. Allo stesso modo, non esiste alcun modo per passare da una versione del codice del prodotto da una release LTS a una release CD o viceversa. Tuttavia, è possibile commutare un gestore code tra i modelli di release. Consultare Migrazione tra release LTS e release CD.

Nota:

Le release di IBM MQ 9.0.x e IBM MQ 9.1.x CD avevano FMID dipendenti da versione e release separati. Pertanto, il passaggio da 9.0.x CD a 9.1.x CD richiede almeno un'installazione SMP/E completa.

Da IBM MQ for z/OS 9.2.0, la release di CD usa una serie di FMID che rimangono gli stessi per tutte le release di IBM MQ for z/OS con un numero di versione 9. Poiché ogni nuova versione di IBM MQ è disponibile sia come release CD che come release LTS , è possibile aggiornare i release CD applicando le PTF a una singola installazione SMP/E anche quando si supera un limite di versione principale. Ad esempio, è possibile passare da IBM MQ for z/OS 9.2.0 CD, a IBM MQ for z/OS 9.2.2 CD, a IBM MQ for z/OS 9.2.4 CD, a IBM MQ for z/OS 9.3.0 CD, applicando le PTF.

È possibile distinguere tra una release LTS e CD con lo stesso livello VRM cercando il messaggio CSQY000I nel log del lavoro del gestore code.

Manutenzione

IBM MQ for z/OS utilizza le PTF per la manutenzione.

LTS Le PTF sono specifiche di una particolare serie di librerie corrispondenti a un livello di release specifico. Per le funzioni di UNIX System Services (ovvero, JMS e UI WEB, Connector Pack e Managed File Transfer), le PTF z/OS sono allineate direttamente con i fix pack Multiplatforms e gli aggiornamenti di sicurezza cumulativi (CSU). Queste correzioni sono cumulative e sono rese disponibili contemporaneamente al fix pack o CSU Multiplatforms equivalente.

CD CD Le CSU di solito non vengono rese disponibili tra le release del CD, ma sono incluse nella release IBM MQ for z/OS CD successiva. Puoi anche contattare il supporto per richiedere un + + USERMOD.

Altre correzioni su IBM MQ for z/OS sono correzioni distinte su parti particolari. Queste correzioni risolvono problemi specifici, non sono cumulative e vengono rese disponibili man mano che vengono prodotte.

Migrazione tra release LTS e release CD

Esistono vincoli e limitazioni, ma generalmente è possibile migrare un singolo gestore code dall'utilizzo del codice release LTS al codice release CD o dall'utilizzo del codice release CD al codice release LTS a condizione che la release di destinazione sia superiore a quella in uso prima della migrazione.

Da IBM MQ for z/OS 9.2.0, è possibile migrare avanti e indietro tra le release CD e LTS con lo stesso VRM tutte le volte che è necessario e senza alcun impatto sulla capacità di migrare all'indietro. Ad esempio, un gestore code può essere avviato in IBM MQ for z/OS 9.3.0 LTS, quindi arrestato e avviato in IBM MQ for z/OS 9.3.0 CD, quindi arrestato e avviato in IBM MQ for z/OS 9.3.0 LTS.

IBM MQ for z/OS ha tradizionalmente fornito una funzionalità di fallback (migrazione all'indietro) in modo che dopo un periodo di esecuzione dopo una migrazione sia possibile eseguire il fallback alla release precedente. Questa funzione viene conservata per le release LTS e per quelle CD con un modificatore di `0` come 9.3.0 CD, ma non è possibile quando l'origine o la destinazione di una migrazione è una release CD con un numero di modificatore diverso da zero, ad esempio 9.2.5 o 9.3.1.

I seguenti sono scenari di migrazione validi e illustrano il funzionamento di questo principio:

Release di origine	Release di destinazione	Note
LTS 9.1.0	CD 9.4.0 LTS o 9.4.0	La migrazione all'indietro non è supportata poiché 9.1.0 LTS non è supportato standard.
LTS 9.2.0	CD 9.4.0 LTS o 9.4.0	Migrazione all'indietro supportata.
LTS 9.3.0	CD 9.4.0 LTS o 9.4.0	Migrazione all'indietro supportata.
CD 9.3.5	CD 9.4.0 LTS o 9.4.0	La migrazione all'indietro non è supportata poiché il release di origine è CD con un modificatore diverso da zero.
CD 9.4.0 LTS o 9.4.0	CD 9.4.1	Migrazione all'indietro non supportata poiché la release di destinazione è CD con un modificatore diverso da zero. `Write to operator with reply` CSQY041D viene emesso per confermare la migrazione.

Attività correlate

z/OS Applicazione e rimozione della manutenzione su z/OS

Informazioni correlate
Download di IBM MQ 9.4

Considerazioni su IBM MQ e IBM MQ Appliance on premise per la conformità al GDPR

Per i PID:

Distribuito

- IBM MQ/IBM MQ Advanced - 5724-H72
- IBM MQ for HPE NonStop - 5724-A39

z/OS

- IBM MQ for z/OS - 5655-MQ9
- IBM MQ for z/OS Value Unit Edition - 5655-VU9
- IBM MQ Advanced for z/OS - 5655-AV9
- IBM MQ Advanced for z/OS Value Unit Edition - 5655-AV1

IBM MQ Appliance

- IBM MQ Appliance M2003 - 5900 - ALJ
- IBM MQ Appliance M2002 - 5737-H47

Avviso:

Questo documento contiene informazioni che assistono l'utente nella preparazione per la conformità al GDPR. Fornisce informazioni sulle funzioni di IBM MQ configurate ed aspetti dell'utilizzo del prodotto da considerare per essere assistiti nell'organizzazione degli adempimenti del GDPR. Queste informazioni non sono un elenco esaustivo, a causa dei molti modi in cui i client possono scegliere e configurare le funzioni e della grande varietà di modi in cui il prodotto può essere utilizzato da solo e con applicazioni e sistemi di terze parti.

I clienti sono responsabili del rispetto delle normative di conformità, come European Union General Data Protection Regulation. I clienti sono gli unici responsabili di ottenere consigli legali competenti per identificare e interpretare leggi e norme pertinenti che possono influire sul business ed eventuali azioni da intraprendere per conformarsi a tali normative.

I prodotti, servizi e altre funzionalità qui descritti non sono adatti per tutte le situazioni dei clienti e potrebbero avere una disponibilità limitata. IBM non fornisce consulenza legale, contabile o di revisione contabile né dichiara o garantisce che i suoi servizi o prodotti garantiranno la conformità dei clienti a qualsiasi legge o regolamento.

Sommario

10. Capacità di limitare l'utilizzo dei dati personali
11. Gestione file

GDPR

Il Regolamento generale sulla protezione dei dati (GDPR) è stato adottato dall'Unione Europea ("UE") e si applica dal 25 maggio 2018.

Perché è importante il GDPR?

Il GDPR istituisce un quadro normativo più solido per la protezione dei dati, per il trattamento dei dati personali degli individui. Il GDPR apporta:

- Diritti nuovi e potenziati per gli individui
- Definizione ampliata dei dati personali
- Nuovi obblighi per i processori
- Potenziale per importanti sanzioni pecuniarie in caso di inosservanza
- Notifica di violazione di dati obbligatoria

Ulteriori informazioni su GDPR:

- Portale di informazioni sul GDPR dell'Unione Europea
- Sito webibm.com/GDPR

Configurazione del prodotto - Considerazioni per la conformità al GDPR

Le seguenti sezioni espongono le considerazioni per configurare IBM MQ per aiutare l'organizzazione ad utilizzare il GDPR.

Ciclo di vita dei dati

IBM MQ è un prodotto middleware transazionale orientato ai messaggi che consente alle applicazioni di scambiare in modo asincrono i dati forniti dalle applicazioni. IBM MQ supporta una gamma di API di messaggistica, protocolli e bridge allo scopo di collegare le applicazioni. Come tale, IBM MQ può essere utilizzato per scambiare molte forme di dati, alcuni dei quali potrebbero essere potenzialmente soggetti al GDPR. Esistono diversi prodotti di terze parti con cui IBM MQ potrebbe scambiare dati. Alcuni di questi sono di proprietà di IBM, ma molti altri sono forniti da altri fornitori di tecnologie. Il sito Web Software Product Compatibility Reports fornisce elenchi del software associato. Per considerazioni relative alla conformità al GDPR di un prodotto di terze parti, è necessario consultare la relativa documentazione. Gli amministratori IBM MQ controllano il modo in cui IBM MQ interagisce con i dati che vi passano attraverso, mediante la definizione di code, argomenti e sottoscrizioni.

Quali tipi di flusso di dati attraverso IBM MQ?

Poiché IBM MQ fornisce un servizio di messaggistica asincrona per i dati dell'applicazione, non vi è alcuna risposta definitiva a questa domanda perché i casi di utilizzo variano in base alla distribuzione dell'applicazione. I dati dei messaggi dell'applicazione vengono resi persistenti nei file della coda (serie di pagine o Coupling Facility su z/OS), nei log e negli archivi e il messaggio stesso può contenere dati regolati dal GDPR. I dati dei messaggi forniti dall'applicazione possono essere inclusi anche nei file raccolti per la determinazione dei problemi, ad esempio i log degli errori, i file di traccia e gli FFST. Sull'applicazione z/OS , i dati dei messaggi forniti possono essere inclusi anche nello spazio di indirizzo o nei dump della CF (Coupling Facility).

Di seguito sono riportati alcuni esempi tipici di dati personali che possono essere scambiati utilizzando IBM MQ:

- I dipendenti del cliente (ad esempio; IBM MQ potrebbe essere utilizzato per collegare il libro paga del cliente o i sistemi HR)

- I dati personali dei clienti del cliente (ad esempio; IBM MQ potrebbe essere utilizzato da un cliente per lo scambio di dati tra le applicazioni relative ai propri clienti, ad esempio l'acquisizione di lead di vendita e la memorizzazione di dati all'interno del proprio sistema CRM).
- I dati personali sensibili dei clienti del cliente (ad esempio; IBM MQ potrebbe essere utilizzato all'interno di contesti industriali che richiedono lo scambio di dati personali, come i record sanitari HL7-based quando si integrano le applicazioni cliniche).

Oltre ai dati dei messaggi forniti dall'applicazione, IBM MQ elabora i seguenti tipi di dati:

- Credenziali di autenticazione (come nome utente e password, chiavi API, ecc.)
- Informazioni personali tecnicamente identificabili (come ID dispositivo, identificativi basati sull'utilizzo, indirizzo IP, ecc.) - quando è collegato a un individuo)

Dati personali utilizzati per il contatto online con IBM

I clienti IBM MQ possono inoltrare commenti / feedback/richieste online per contattare IBM su IBM MQ argomenti in vari modi, principalmente:

- Area dei commenti pubblici sulle pagine nell'area IBM MQ su IBM Developer
- Area commenti pubblici sulle pagine di Informazioni sul prodotto IBM MQ in IBM Documentation
- Commenti pubblici in IBM Support Forums
- Commenti pubblici in IBM Integration Ideas

Generalmente, vengono utilizzati solo il nome del cliente e l'indirizzo email, per abilitare le risposte personali per l'oggetto del contatto e l'uso dei dati personali è conforme alla IBM Online Privacy Statement.

Raccolta dati

IBM MQ può essere utilizzato per raccogliere dati personali. Nel valutare il tuo utilizzo di IBM MQ e le tue esigenze di soddisfare le richieste del GDPR, dovresti considerare i tipi di dati personali che, nelle tue circostanze, stanno passando attraverso IBM MQ. È possibile considerare aspetti quali:

- In che modo i dati arrivano ai gestori code? (In quali protocolli? I dati sono crittografate? I dati sono firmati?)
- Come vengono inviati i dati dai gestori code? (In quali protocolli? I dati sono crittografate? I dati sono firmati?)
- Come vengono memorizzati i dati mentre passano attraverso un gestore code? (Qualsiasi applicazione di messaggistica ha il potenziale per scrivere i dati del messaggio su un supporto con stato, anche se un messaggio è non persistente. Si è consapevoli del modo in cui le funzioni di messaggistica potrebbero potenzialmente esporre aspetti dei dati dei messaggi dell'applicazione che passano attraverso il prodotto?)
- In che modo le credenziali vengono raccolte e archiviate dove necessario da IBM MQ per accedere alle applicazioni di terze parti?

IBM MQ potrebbe dover comunicare con altri sistemi e servizi che richiedono l'autenticazione, ad esempio LDAP. Dove necessario, i dati di autenticazione (ID utente, password) vengono configurati e memorizzati da IBM MQ per il loro utilizzo in tali comunicazioni. Laddove possibile, è necessario evitare di utilizzare le credenziali personali per l'autenticazione IBM MQ . Considerare la protezione della memoria utilizzata per i dati di autenticazione. (Consultare Data Storage di seguito.)

Archiviazione dati

Quando i dati del messaggio viaggiano attraverso i gestori code, IBM MQ persisterà (forse più copie di) tali dati direttamente su un supporto con stato. Gli utenti IBM MQ potrebbero considerare di proteggere i dati dei messaggi mentre sono inattivi.

I seguenti elementi evidenziano le aree in cui IBM MQ persiste i dati forniti dall'applicazione, che gli utenti possono considerare quando assicurano la conformità con il GDPR.

- Code messaggi applicazione:

  IBM MQ fornisce code di messaggi per consentire lo scambio di dati asincrono tra le applicazioni. I messaggi persistenti e non persistenti memorizzati su una coda vengono scritti su un supporto con stato.
- Code agent trasferimento file

  IBM MQ Managed File Transfer utilizza code di messaggi per coordinare il trasferimento affidabile dei dati dei file, i file che contengono dati personali e i record dei trasferimenti vengono memorizzati su queste code.
- Code di trasmissione

  Per trasferire i messaggi in modo affidabile tra gestori code, i messaggi vengono memorizzati temporaneamente nelle code di trasmissione.
- Code di messaggi non recapitabili:

  Esistono alcune circostanze in cui i messaggi non possono essere inseriti in una coda di destinazione e vengono memorizzati in una coda di messaggi non recapitabili, se tale coda è configurata sul gestore code.
- Code di backout:

  Le interfacce di messaggistica JMS e XMS forniscono una funzione che consente ai messaggi non elaborabili di essere spostati in una coda di backout dopo che si sono verificati diversi backout per consentire l'elaborazione di altri messaggi validi.
- Coda errori AMS:

  IBM MQ Advanced Message Security sposterà i messaggi non conformi a una politica di sicurezza nel SISTEMA SYSTEM.PROTECTION.ERROR.QUEUE è simile alla coda dei messaggi non instradabili.
- Pubblicazioni conservate:

  IBM MQ fornisce una funzione di pubblicazione conservata per consentire alle applicazioni di sottoscrizione di richiamare una pubblicazione precedente.
- Consegna rinviata:

  IBM MQ supporta la funzione di ritardo di consegna JMS 2.0 e Jakarta Messaging 3.0 che consente ai messaggi di essere consegnati a destinazione in un momento futuro. I messaggi che non sono stati ancora recapitati vengono memorizzati nel SISTEMA SYSTEM.DDELAY.LOCAL.QUEUE .

Ulteriori informazioni:

- Registrazione: verifica della perdita di messaggi
- Impostazioni coda agent MFT
- Utilizzo della coda di messaggi non recapitabili
- Gestione di messaggi non elaborabili nelle classi IBM MQ for JMS
- Gestione degli errori AMS
- Pubblicazioni conservate
- JMS 2.0 ritardo di consegna

I seguenti elementi evidenziano le aree in cui IBM MQ può conservare indirettamente i dati forniti dall'applicazione che gli utenti possono anche voler considerare quando assicurano la conformità con il GDPR.

- Messaggistica di instradamento traccia:

  IBM MQ fornisce funzioni di instradamento traccia, che registrano l'instradamento di un messaggio tra le applicazioni. I messaggi di evento generati possono includere informazioni personali tecnicamente identificabili come indirizzi IP.
- Traccia attività applicazione:

IBM MQ fornisce la traccia dell'attività dell'applicazione, che registra le attività API di messaggistica di applicazioni e canali, la traccia dell'attività dell'applicazione può registrare il contenuto dei dati dei messaggi forniti dall'applicazione ai messaggi di eventi.

- Traccia servizio:

  IBM MQ fornisce funzioni di traccia del servizio, che registrano i percorsi del codice interno attraverso i quali vengono trasmessi i dati del messaggio. Come parte di queste funzioni, IBM MQ può registrare il contenuto dei dati dei messaggi forniti dall'applicazione nei file di traccia memorizzati sul disco.

- Eventi gestore code:

  IBM MQ può generare messaggi di evento che possono includere dati personali, come eventi di configurazione, comandi e autorizzazioni.

Ulteriori informazioni:

- Messaggistica di indirizzamento traccia
- Utilizzo della traccia
- Monitoraggio eventi
- Eventi di gestori code

Per proteggere l'accesso alle copie dei dati del messaggio forniti dall'applicazione, considerare le azioni riportate di seguito:

- Limitare l'accesso utente privilegiato ai dati IBM MQ nel sistema di file, ad esempio limitando l'appartenenza dell'utente al gruppo 'mqm' sulle piattaforme UNIX and Linux® .
- Limitare l'accesso dell'applicazione ai dati IBM MQ tramite code dedicate e controllo accessi. Se necessario, evitare la condivisione non necessaria delle risorse, come le code tra le applicazioni e fornire un controllo di accesso granulare alle risorse di argomenti e code.
- Limitare l'accesso alle copie replicate dei dati IBM MQ in configurazioni HA (High Availability) o DR (Disaster Recovery) e proteggere le connessioni utilizzate per la replica.
- Utilizzare IBM MQ Advanced Message Security per fornire la firma end-to-end e / o la codifica dei dati del messaggio.
- Utilizzare la crittografia a livello di file o di volume per proteggere le directory o i file system che potrebbero contenere dati, traccia o log IBM MQ .
- Una volta caricata la traccia del servizio in IBM, è possibile eliminare i file di traccia del servizio e i dati FFST se si è preoccupati del contenuto potenzialmente contenente dati personali.

Ulteriori informazioni:

- Utenti privilegiati
- Pianificazione del supporto file system su Multiplatforms
- Codifica del filesystem su IBM MQ Appliance

Un amministratore IBM MQ può configurare un gestore code con credenziali (nome utente e password, chiavi API, ecc.) per servizi di terzi come LDAP. Questi dati vengono generalmente memorizzati nella directory dei dati del gestore code protetta tramite le autorizzazioni del filesystem.

Quando viene creato un gestore code IBM MQ , la directory dei dati viene impostata con il controllo accessi basato sul gruppo in modo che IBM MQ possa leggere i file di configurazione e utilizzare le credenziali per connettersi a tali sistemi. Gli amministratori IBM MQ sono considerati utenti privilegiati e sono membri di questo gruppo, quindi hanno accesso in lettura ai file. Alcuni file sono offuscati ma non sono crittografati. Per questo motivo, per proteggere completamente l'accesso alle credenziali, è necessario considerare le seguenti azioni:

- Limitare l'accesso utente privilegiato ai dati IBM MQ , ad esempio limitando l'appartenenza del gruppo 'mqm' sulle piattaforme UNIX and Linux .
- Utilizzare la codifica a livello di file o volume per proteggere il contenuto della directory dei dati del gestore code.

- Crittografare i backup della directory di configurazione di produzione e memorizzarli con i controlli di accesso appropriati.
- Considerare la possibilità di fornire tracce di controllo per errori di autenticazione, controllo degli accessi e modifiche di configurazione con eventi di sicurezza, comandi e configurazione.

Ulteriori informazioni:

- Protezione di IBM MQ

Accesso ai dati

È possibile accedere ai dati del gestore code IBM MQ tramite le seguenti interfacce del prodotto, alcune delle quali sono progettate per l'accesso tramite una connessione remota e altre per l'accesso mediante una connessione locale.

- IBM MQ Console [Solo remoto]
- API REST di gestione IBM MQ [Solo remota)
- IBM MQ Messaggistica REST API [solo remota]
- MQI [Locale e remoto]
- JMS [locale e remoto]
- XMS [Locale e Remoto]
- IBM MQ Telemetria (MQTT) [Solo remoto]
- IBM MQ Light (AMQP) [Solo remoto]
- Bridge IBM MQ IMS [Solo locale]
- Bridge IBM MQ CICS [Solo locale]
- IBM MQ MFT Protocol bridges [Solo remoto]
- IBM MQ Connect:Direct bridge [Solo remoto]
- IBM MQ MQAI [Locale e remoto]
- Comandi PCF IBM MQ [locale e remoto]
- Comandi IBM MQ MQSC [Locale e remoto]
- IBM MQ Explorer [Locale e Remoto]
- Uscite utente IBM MQ [solo locali]
- IBM MQ Internet Pass-Thru [Solo remoto]
- Metriche Red Hat® OpenShift® Monitoring (Prometheus) (le metriche sono dati numerici sulle statistiche del gestore code)
- IBM MQ Appliance Console seriale [Solo locale]
- IBM MQ Appliance SSH [Solo remoto]
- API REST IBM MQ Appliance [Solo remota]
- IBM MQ Appliance Web UI [Solo remota]
- 
IBM MQ Kafka Connectors (Kafka Connect) [Locale e Remoto]

Le interfacce sono progettate per consentire agli utenti di apportare modifiche a un gestore code IBM MQ e ai messaggi memorizzati su di esso. Le operazioni di amministrazione e di messaggistica sono protette in modo che vi siano tre fasi coinvolte quando viene effettuata una richiesta;

- Autenticazione
- Associazione ruolo
- Autorizzazione

Autenticazione:

Se il messaggio o l'operazione di gestione è stata richiesta da una connessione locale, l'origine di questa connessione è un processo in esecuzione sullo stesso sistema. L'utente che esegue il processo deve aver superato tutte le fasi di autenticazione fornite dal sistema operativo. Il nome utente del proprietario del processo da cui è stata effettuata la connessione viene dichiarato come identità. Ad esempio, questo potrebbe essere il nome dell'utente che esegue la shell da cui è stata avviata un'applicazione. Le possibili forme di autenticazione per le connessioni locali sono:

1. Nome utente asserito (SO locale)
2. Nome utente e password facoltativi (OS, LDAP o repository 3rd parti personalizzati)
3. Solo IBM MQ token di sicurezza (JWT)

Se l'azione di gestione è stata richiesta da una connessione remota, le comunicazioni con IBM MQ vengono effettuate tramite un'interfaccia di rete. Le seguenti forme di identità possono essere presentate per l'autenticazione tramite connessioni di rete;

1. Nome utente asserito (dal sistema operativo remoto)
2. Nome utente e password (SO, LDAP o repository 3rd parti personalizzati)
3. Indirizzo di rete di origine (come l'indirizzo IP)
4. X.509 Certificato digitale (autenticazione SSL/TLS reciproca)
5. Token di sicurezza (come il token LTPA2 o il token JWT)
6. Altra sicurezza personalizzata (funzionalità fornita dalle uscite di 3rd parti)
7. Chiavi SSH

L'integrazione di IBM MQcon IBM Cloud Pak for Integration aggiunge un nuovo tipo di autenticazione per il IBM MQ Console: Single Sign - On con il Cloud Pak. (soloCP4I)

Associazione ruoli:

Nella fase di associazione dei ruoli, le credenziali fornite nella fase di autenticazione possono essere associate a un identificativo utente alternativo. A condizione che l'identificativo utente associato sia autorizzato a procedere (ad esempio, gli utenti amministrativi possono essere bloccati dalle regole di autenticazione del canale), l'ID utente associato viene portato avanti nella fase finale quando si autorizzano le attività rispetto alle risorse IBM MQ .

Autorizzazione:

IBM MQ consente a utenti differenti di disporre di autorizzazioni differenti per le diverse risorse di messaggistica, come code, argomenti e altri oggetti gestore code.

Registrazione attività:

Alcuni utenti di IBM MQ potrebbero dover creare un record di controllo dell'accesso alle risorse MQ . Esempi di log di controllo desiderati potrebbero includere modifiche alla configurazione che contengono informazioni sulla modifica oltre a chi l'ha richiesto.

Le seguenti fonti di informazioni sono disponibili per implementare questo requisito:

1. Un gestore code IBM MQ può essere configurato per produrre eventi di comando quando un comando di gestione è stato eseguito correttamente.
2. Un gestore code IBM MQ può essere configurato per produrre eventi di configurazione quando una risorsa del gestore code viene creata, modificata o eliminata.
3. Un gestore code IBM MQ può essere configurato per produrre un evento di autorizzazione quando un controllo di autorizzazione non riesce per una risorsa.
4. Messaggi di errore che indicano che i controlli di autorizzazione non riusciti vengono scritti nei log degli errori del gestore code.
5. La console IBM MQ scriverà i messaggi di controllo nei relativi log quando l'autenticazione, i controlli di autorizzazione hanno esito negativo o quando i gestori code vengono creati, avviati, arrestati o eliminati.

6. IBM MQ Appliance scriverà i messaggi di controllo nei propri log per registrare i login utente e le modifiche di sistema.

Quando si considera questo tipo di soluzioni, gli utenti di IBM MQ potrebbero voler considerare i seguenti punti:

- I messaggi di eventi non sono persistenti, quindi quando un gestore code viene riavviato le informazioni vengono perse. I monitor di eventi devono essere configurati per consumare costantemente i messaggi disponibili e trasferire il contenuto ai supporti persistenti.
- Gli utenti con privilegi IBM MQ dispongono di privilegi sufficienti per disabilitare gli eventi, cancellare i log o eliminare i gestori code.

Per ulteriori informazioni sulla sicurezza dell'accesso ai dati IBM MQ e sulla fornitura di una traccia di controllo, fare riferimento ai seguenti argomenti:

- IBM MQ meccanismi di sicurezza
- Eventi di configurazione
- Eventi di comandi
- Utilizzo dei log degli errori

Elaborazione dati

Codifica mediante un'infrastruttura di chiavi pubbliche:

È possibile proteggere le connessioni di rete a IBM MQ specificando che le connessioni utilizzano TLS, che può anche fornire l'autenticazione reciproca del lato iniziale della connessione.

L'utilizzo delle funzioni di sicurezza PKI fornite dai meccanismi di trasporto è il primo passo per proteggere l'elaborazione dei dati con IBM MQ. Tuttavia, senza abilitare ulteriori funzioni di sicurezza, il comportamento di un'applicazione di consumo è quello di elaborare tutti i messaggi consegnati ad essa senza convalidare l'origine del messaggio o se è stato modificato durante il transito.

Gli utenti di IBM MQ che dispongono della licenza per utilizzare le funzionalità Advanced Message Security (AMS) possono controllare il modo in cui le applicazioni elaborano i dati personali contenuti nei messaggi, mediante la definizione e configurazione delle politiche di sicurezza. Le politiche di sicurezza consentono di applicare la firma digitale e / o la codifica ai dati dei messaggi tra le applicazioni.

È possibile utilizzare le politiche di sicurezza per richiedere e convalidare una firma digitale quando si utilizzano messaggi per garantire che i messaggi siano autentici. La crittografia AMS fornisce un metodo con cui i dati del messaggio vengono convertiti da un formato leggibile a una versione codificata che può essere decodificata solo da un'altra applicazione se è il destinatario previsto o il messaggio e ha accesso alla chiave di decrittografia corretta.

Per ulteriori informazioni sull'utilizzo di SSL e dei certificati per proteggere le connessioni di rete, fare riferimento ai seguenti argomenti nella documentazione del prodotto IBM MQ :

- Configurazione della sicurezza TLS per IBM MQ
- Panoramica AMS

Eliminazione dei dati

IBM MQ fornisce comandi e azioni dell'interfaccia utente per eliminare i dati forniti al prodotto. Ciò significa che gli utenti di IBM MQ possono eliminare i dati relativi a particolari individui, se necessario.

- Aree di comportamento IBM MQ da considerare per la conformità con l'eliminazione dei dati del client GDPR
  - Eliminare i dati del messaggio memorizzati su una coda dell'applicazione:
    - Rimozione di singoli messaggi utilizzando l'API di messaggistica o gli strumenti o utilizzando la scadenza del messaggio.

    - Specificando che i messaggi sono non persistenti, conservati su una coda in cui la classe di messaggi non persistenti è normale e riavviando il gestore code.
    - Cancellare amministrativamente la coda.
    - Eliminazione della coda.
  - Eliminare i dati di pubblicazione conservati memorizzati su un argomento da:
    - Specificare che i messaggi sono non persistenti e riavviare il gestore code.
    - Sostituzione dei dati conservati con nuovi dati o utilizzando la scadenza del messaggio.
    - Cancellare amministrativamente la stringa di argomenti.
  - Eliminare i dati memorizzati su un gestore code eliminando l'intero gestore code e tutte le copie replicate per l'alta disponibilità o il ripristino di emergenza.
  - Eliminare i dati memorizzati dai comandi di traccia del servizio eliminando i file nella directory di traccia.
  - Eliminare i dati FFST memorizzati eliminando i file nella directory degli errori.
  - Eliminare lo spazio di indirizzi e i dump della CF (Coupling Facility) (su z/OS).
  - Eliminare l'archivio, il backup o altre copie di tali dati.
- Aree di comportamento IBM MQ da considerare per la conformità con l'eliminazione dei dati dell'account GDPR
  - È possibile eliminare i dati di account e le preferenze memorizzate da IBM MQ per la connessione ai gestori code e ai servizi di 3rd eliminando (incluse le copie di archivio, di backup o in altro modo replicate):
    - Oggetti delle informazioni di autenticazione del gestore code che memorizzano credenziali.
    - Record di autorizzazione del gestore code che fanno riferimento agli identificatori utente.
    - Regole di autenticazione del canale del gestore code che associano o bloccano indirizzi IP specifici, DN certificato o identificatori utente.
    - File di credenziali utilizzati da IBM MQ Managed File Transfer Agent, Logger e MQ Explorer MFT Plugin per l'autenticazione con il gestore code e i file server.
    - I certificati digitali X.509 che rappresentano o contengono informazioni su un individuo dai keystore che possono essere utilizzati dalle connessioni SSL/TLS o IBM MQ Advanced Message Security (AMS).
    - Singoli account utente da IBM MQ Appliance, incluso il riferimento a tali account nei file di log del sistema.
    - Metadati dello spazio di lavoro IBM MQ Explorer e impostazioni Eclipse .
    - IBM MQ Explorer come specificato in Preferenze password.
    - IBM MQ Console e file di configurazione del server mqweb.
    - File di configurazione e keystore IBM MQ Internet Pass-Thru .

Ulteriori informazioni:

- MFT e IBM MQ autenticazione della connessione
- Associazione delle credenziali per un server di file utilizzando il file ProtocolBridgeCredentials.xml
- Configurazione di utenti e ruoli IBM MQ Console

Monitoraggio dei dati

IBM MQ fornisce una serie di funzioni di controllo che gli utenti possono utilizzare per comprendere meglio le prestazioni delle applicazioni e dei gestori code.

IBM MQ fornisce inoltre una serie di funzioni che consentono di gestire i log degli errori del gestore code.

Ulteriori informazioni:

- Monitoraggio della rete IBM MQ
- Servizi di messaggi diagnostici
- Servizio QMErrorLog
- IBM MQ Appliance monitoraggio e reporting

Capacità di limitare l'uso dei dati personali

Utilizzando le funzioni riepiloga in questo documento, IBM MQ consente all'utente finale di limitare l'utilizzo dei propri dati personali.

Le code di messaggi IBM MQ non devono essere utilizzate come archivio dati permanente allo stesso modo di un database, il che è particolarmente vero quando si gestiscono i dati dell'applicazione soggetti al GDPR.

A differenza di un database in cui i dati possono essere trovati attraverso una query di ricerca, può essere difficile trovare i dati del messaggio a meno che non si conosca la coda, il messaggio e gli identificatori di correlazione di un messaggio.

I messaggi contenenti i dati di un individuo possono essere prontamente identificati e localizzati, è possibile utilizzare le funzioni di messaggistica IBM MQ standard per accedere o modificare i dati del messaggio.

Gestione file

1. IBM MQ Managed File Transfer non esegue la scansione del malware sui file trasferiti. I file vengono trasferiti così come sono e viene eseguito un controllo di integrità per garantire che i dati file non vengano modificati durante il trasferimento. I checksum di origine e di destinazione vengono pubblicati come parte della pubblicazione dello stato di trasferimento. Si consiglia agli utenti finali di implementare la scansione del malware come appropriato per il loro ambiente prima che MFT trasferisca il file e dopo che MFT distribuisca un file a un endpoint remoto.
2. IBM MQ Managed File Transfer non esegue azioni in base al tipo MIME o all'estensione file. MFT legge i file e trasferisce i byte esattamente come letti dal file di input.

Architetture basate su un singolo gestore code

Le architetture IBM MQ più semplici implicano la configurazione e l'utilizzo di un singolo gestore code.

Prima di pianificare la propria architettura IBM MQ , acquisire familiarità con i concetti di IBM MQ di base. Vedere IBM MQ Panoramica tecnica.

Nelle seguenti sezioni sono descritte diverse possibili architetture che utilizzano un singolo gestore code:

- “Singolo gestore code con applicazioni locali che accedono a un servizio” a pagina 18
- “Singolo gestore code con applicazioni remote che accedono a un servizio come client” a pagina 19
- “Singolo gestore code con una configurazione di pubblicazione / sottoscrizione” a pagina 19

Singolo gestore code con applicazioni locali che accedono a un servizio

La prima architettura basata su un singolo gestore code è quella in cui le applicazioni che accedono a un servizio sono in esecuzione sullo stesso sistema delle applicazioni che forniscono il servizio. Un gestore code IBM MQ fornisce l'intercomunicazione asincrona tra le applicazioni che richiedono il servizio e quelle che forniscono il servizio. Ciò significa che la comunicazione tra le applicazioni può continuare anche se una delle applicazioni è offline per un lungo periodo di tempo.

Singolo gestore code con applicazioni remote che accedono a un servizio come client

La seconda architettura basata su un singolo gestore code ha le applicazioni in esecuzione in remoto dalle applicazioni che forniscono il servizio. Le applicazioni remote sono in esecuzione su sistemi differenti per i servizi. Le applicazioni si connettono come client al singolo gestore code. Questo significa che l'accesso a un servizio può essere fornito a più sistemi tramite un singolo gestore code.

Una limitazione di questa architettura è che una connessione di rete deve essere disponibile per il funzionamento di un'applicazione. L'interazione tra l'applicazione e il gestore code sulla connessione di rete è sincrona.

Singolo gestore code con una configurazione di pubblicazione / sottoscrizione

Un'architettura alternativa che utilizza un singolo gestore code è quella di utilizzare una configurazione di pubblicazione / sottoscrizione. Nella messaggistica di pubblicazione / sottoscrizione, è possibile disaccoppiare il provider di informazioni dai consumatori di tali informazioni. Ciò differisce dagli stili di messaggistica punto a punto nelle architetture precedentemente descritte, in cui le applicazioni devono conoscere le informazioni sull'applicazione di destinazione, ad esempio il nome della coda su cui inserire i messaggi. Utilizzando la pubblicazione / sottoscrizione IBM MQ , l'applicazione di invio pubblica un messaggio con un argomento specificato in base all'oggetto delle informazioni. IBM MQ gestisce la distribuzione del messaggio alle applicazioni che hanno registrato un interesse in tale soggetto tramite una sottoscrizione. Inoltre, le applicazioni riceventi non hanno bisogno di conoscere l'origine dei messaggi per riceverli. Per ulteriori informazioni, consultare Messaggistica di pubblicazione / sottoscrizione e Esempio di una singola configurazione di pubblicazione / sottoscrizione del gestore code.

Concetti correlati

Introduzione a IBM MQ

Attività correlate

“Pianificazione di un'architettura IBM MQ” a pagina 5
Quando si pianifica l'ambiente IBM MQ , considerare il supporto fornito da IBM MQ per le architetture di gestori code singoli e multipli e per gli stili di messaggistica point-to-point e di pubblicazione / sottoscrizione. Inoltre, pianificare i requisiti delle risorse e l'utilizzo delle funzioni di registrazione e backup.

Creazione e gestione di gestori code su più piattaforme

Architetture basate su più gestori code

È possibile utilizzare le tecniche di accodamento dei messaggi distribuiti per creare un'architettura IBM MQ che implica la configurazione e l'utilizzo di più gestori code.

Prima di pianificare la propria architettura IBM MQ , acquisire familiarità con i concetti di IBM MQ di base. Vedere IBM MQ Panoramica tecnica.

Un'architettura IBM MQ può essere modificata, senza alcuna modifica alle applicazioni che forniscono servizi, aggiungendo ulteriori gestori code.

Le applicazioni possono essere ospitate sulla stessa macchina di un gestore code e quindi ottenere una comunicazione asincrona con un servizio ospitato su un altro gestore code su un altro sistema. In alternativa, le applicazioni che accedono a un servizio possono connettersi come client a un gestore code che fornisce l'accesso asincrono al servizio su un altro gestore code.

Gli instradamenti che connettono gestori code differenti e le relative code vengono definiti utilizzando tecniche di accodamento distribuito. I gestori code nell'architettura sono connessi utilizzando i canali. I canali vengono utilizzati per spostare automaticamente i messaggi da un gestore code a un altro in una direzione a seconda della configurazione dei gestori code.

Per una panoramica di alto livello della pianificazione di una rete IBM MQ , consultare “Progettazione di reti di gestori code distribuiti” a pagina 21.

Per informazioni su come pianificare i canali per la tua architettura IBM MQ , vedi Tecniche di accodamento distribuito IBM MQ.

La gestione delle code distribuite consente di creare e monitorare la comunicazione tra gestori code. Per ulteriori informazioni sulla gestione delle code distribuite, consultare Introduzione alla gestione delle code distribuite.

Attività correlate

“Pianificazione di un'architettura IBM MQ” a pagina 5
Quando si pianifica l'ambiente IBM MQ , considerare il supporto fornito da IBM MQ per le architetture di gestori code singoli e multipli e per gli stili di messaggistica point-to-point e di pubblicazione / sottoscrizione. Inoltre, pianificare i requisiti delle risorse e l'utilizzo delle funzioni di registrazione e backup.

Creazione e gestione di gestori code su più piattaforme

Pianificazione delle code e dei cluster distribuiti

È possibile connettere manualmente le code che si trovano sui gestori code distribuiti oppure è possibile creare un cluster di gestori code e consentire al prodotto di connettere i gestori code. Per scegliere una topologia adatta per la rete di messaggistica distribuita, è necessario considerare i requisiti per il controllo manuale, la dimensione della rete, la frequenza di modifica, la disponibilità e la scalabilità.

Prima di iniziare

Questa attività presuppone che l'utente comprenda quali sono le reti di messaggistica distribuite e come funzionano. Per una panoramica tecnica, consultare Accodamento distribuito e cluster.

Informazioni su questa attività

Per creare una rete di messaggistica distribuita, è possibile configurare manualmente i canali per connettere le code ospitate su diversi gestori code oppure è possibile creare un cluster di gestori code. Il clustering consente ai gestori code di comunicare tra loro senza la necessità di impostare definizioni di canali supplementari o definizioni di code remote, semplificandone la configurazione e la gestione.

Per scegliere una topologia adatta per la propria rete di pubblicazione / sottoscrizione distribuita, è necessario considerare le seguenti domande generali:

- Di quale controllo manuale hai bisogno per le connessioni nella tua rete?
- Quanto sarà grande la tua rete?
- Quanto sarà dinamico?
- Quali sono i requisiti di disponibilità e scalabilità?

Procedura

- Considerare quanto controllo manuale è necessario sulle connessioni nella rete.

  Se sono necessarie solo poche connessioni o se le singole connessioni devono essere definite in modo molto preciso, è necessario creare la rete manualmente.

  Se sono necessari più gestori code che sono logicamente correlati e che devono condividere dati e applicazioni, è consigliabile raggrupparli insieme in un cluster di gestori code.
- Stimare la dimensione della rete.
  a) Stimare quanti gestori code sono necessari. Tenere presente che le code possono essere ospitate su più di un gestore code.
  b) Se si sta considerando l'utilizzo di un cluster, aggiungere due ulteriori gestori code che fungano da repository completi.

  Per reti più grandi, la configurazione manuale e la manutenzione delle connessioni possono richiedere molto tempo e si consiglia di utilizzare un cluster.

- Considerare la dinamica dell'attività di rete.

  Pianificare le code occupate da ospitare sui gestori code con prestazioni.

  Se si prevede che le code vengano create ed eliminate di frequente, considerare l'uso di un cluster.
- Considera i tuoi requisiti di disponibilità e scalabilità.
  a) Decidere se è necessario garantire l'alta disponibilità dei gestori code. In tal caso, stimare il numero di gestori code a cui si applica questo requisito.
  b) Considerare se alcuni dei gestori code sono meno capaci di altri.
  c) Considerare se i collegamenti di comunicazione ad alcuni dei gestori code sono più fragili rispetto ad altri.
  d) Considerare la possibilità di ospitare code su più gestori code.

  Le reti e i cluster configurati manualmente possono essere entrambi configurati per essere altamente disponibili e scalabili. Se si utilizza un cluster, è necessario definire due ulteriori gestori code come repository completi. Disporre di due repository completi garantisce che il cluster continui a funzionare se uno dei repository completi diventa non disponibile. Assicurarsi che i gestori code del repository completo siano solidi, performanti e abbiano una buona connettività di rete. Non pianificare l'utilizzo dei gestori code del repository completo per qualsiasi altro lavoro.
- In base a questi calcoli, utilizzare i link forniti per decidere se configurare manualmente le connessioni tra i gestori code o utilizzare un cluster.

Operazioni successive

È ora possibile configurare la rete di messaggistica distribuita.

Attività correlate

Configurazione dell'accodamento distribuito

Configurazione di un cluster di gestore code

Progettazione di reti di gestori code distribuiti

IBM MQ invia e riceve i dati tra le applicazioni e sulle reti utilizzando gestori code e canali. La pianificazione della rete implica la definizione dei requisiti per creare un framework per la connessione di questi sistemi su una rete.

I canali possono essere creati tra il sistema e qualsiasi altro sistema con cui è necessario disporre di comunicazioni. È possibile creare canali multi - hop per connettersi a sistemi in cui non si dispone di connessioni dirette. Le connessioni del canale dei messaggi descritte negli scenari vengono mostrati come diagramma di rete in Figura 1 a pagina 22.

Se è necessario creare canali tra sistemi su reti fisiche differenti o canali che comunicano attraverso un firewall, l'utilizzo di IBM MQ Internet Pass-Thru potrebbe semplificare la configurazione. Per ulteriori informazioni, consultare IBM MQ Internet Pass-Thru.

Nomi di canali e code di trasmissione

È possibile assegnare qualsiasi nome alle code di trasmissione. Ma per evitare confusione, è possibile assegnare loro gli stessi nomi dei nomi dei gestori code di destinazione o dei nomi degli alias dei gestori code, a seconda dei casi. Ciò associa la coda di trasmissione all'instradamento che utilizzano, fornendo una chiara panoramica degli instradamenti paralleli creati tramite gestori code intermedi (con più passaggi).

Non è così chiaro per i nomi dei canali. I nomi dei canali in Figura 1 a pagina 22 per QM2, ad esempio, devono essere diversi per i canali in entrata e in uscita. Tutti i nomi canale possono ancora contenere i propri nomi di coda di trasmissione, ma devono essere qualificati per renderli univoci.

Ad esempio, in QM2, esiste un canale QM3 proveniente da QM1e un canale QM3 che passa a QM3. Per rendere univoci i nomi, il primo potrebbe essere denominato QM3_from_QM1e il secondo potrebbe essere denominato QM3_from_QM2. In questo modo, i nomi dei canali mostrano il nome della coda di

trasmissione nella prima parte del nome. La direzione e il nome del gestore code adiacente vengono mostrati nella seconda parte del nome.

Una tabella di nomi di canali suggeriti per Figura 1 a pagina 22 viene fornita in Tabella 1 a pagina 22.

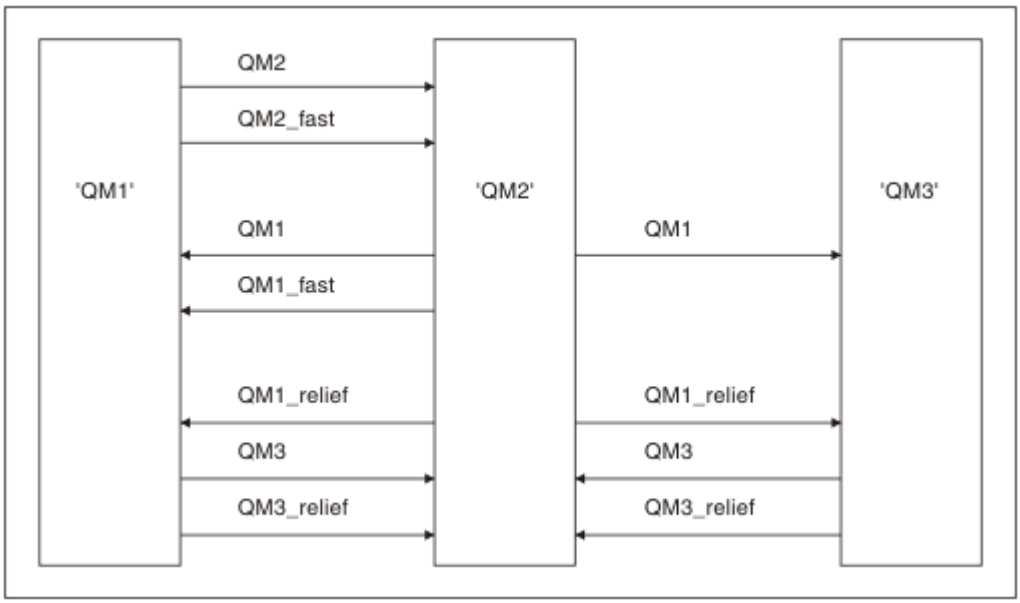


Figura 1. Diagramma di rete che mostra tutti i canali

Tabella 1. Esempio di nomi di canali

Nome instradamento	**Gestori code che ospitano il canale**	**Nome coda di trasmissione**	**Nome canale suggerito**
QM1	QM1 & QM2	QM1 (in QM2)	QM1.from.QM2
QM1	QM2 & QM3	QM1 (in QM3)	QM1.from.QM3
QM1_fast	QM1 & QM2	QM1_fast (in QM2)	QM1_fast.from.QM2
QM1_relief	QM1 & QM2	QM1_relief (alle QM2)	QM1_relief.from.QM2
QM1_relief	QM2 & QM3	QM1_relief (in QM3)	QM1_relief.from.QM3
QM2	QM1 & QM2	QM2 (in QM1)	QM2.from.QM1
QM2_fast	QM1 & QM2	QM2_fast (all'indirizzo QM1)	QM2_fast.from.QM1
QM3	QM1 & QM2	QM3 (in QM1)	QM3.from.QM1
QM3	QM2 & QM3	QM3 (in QM2)	QM3.from.QM2
QM3_relief	QM1 & QM2	QM3_relief (in QM1)	QM3_relief.from.QM1
QM3_relief	QM2 & QM3	QM3_relief (in QM2)	QM3_relief.from.QM2

Nota:

1. z/OS Su IBM MQ for z/OS, i nomi dei gestori code sono limitati a quattro caratteri.
2. Denominare in modo univoco tutti i canali nella rete. Come mostrato nella sezione Tabella 1 a pagina 22, includere i nomi dei gestori code di origine e di destinazione nel nome del canale è un buon modo per farlo.

Pianificatore di rete

La creazione di una rete presuppone l'esistenza di un'altra funzione di livello superiore di *pianificatore di rete* i cui piani sono implementati dagli altri membri del team.

Per le applicazioni ampiamente utilizzate, è più economico pensare in termini di siti di accesso locale per la concentrazione del traffico dei messaggi, utilizzando i collegamenti a banda larga tra i siti di accesso locale, come mostrato in Figura 2 a pagina 23.

In questo esempio ci sono due sistemi principali e un certo numero di sistemi satellitari. La configurazione effettiva dipende da considerazioni di business. Ci sono due gestori code concentratori situati in centri convenienti. Ogni concentratore QM dispone di canali di messaggi per i gestori code locali:

- QM - concentrator 1 dispone di canali di messaggi per ciascuno dei tre gestori code locali, QM1, QM2e QM3. Le applicazioni che utilizzano questi gestori code possono comunicare tra loro tramite i concentratori QM.
- QM - concentrator 2 dispone di canali di messaggi per ognuno dei tre gestori code locali, QM4, QM5e QM6. Le applicazioni che utilizzano questi gestori code possono comunicare tra loro tramite i concentratori QM.
- I concentratori QM hanno canali di messaggi tra loro, consentendo così a qualsiasi applicazione in un gestore code di scambiare messaggi con qualsiasi altra applicazione in un altro gestore code.

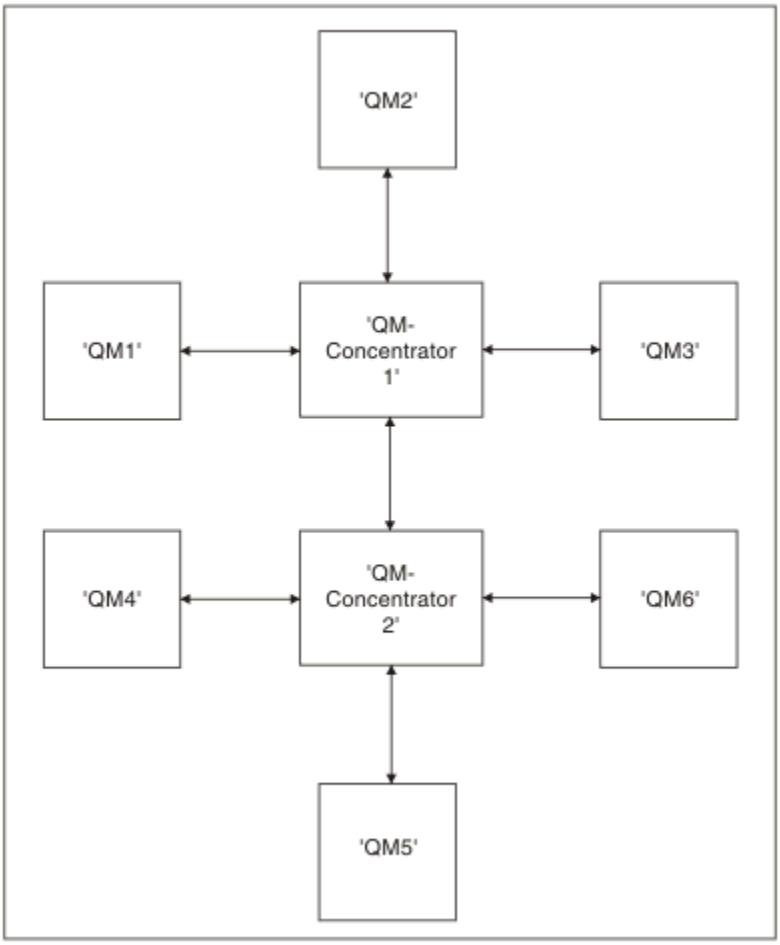


Figura 2. Diagramma di rete che mostra i concentratori QM

Progettazione di cluster

I cluster forniscono un meccanismo per l'interconnessione dei gestori code in modo da semplificare sia la configurazione iniziale che la gestione in corso. I cluster devono essere attentamente progettati per garantire che funzionino correttamente e che raggiungano i livelli richiesti di disponibilità e reattività.

Prima di iniziare

Per un'introduzione ai concetti di clustering, consultare i seguenti argomenti:

- Accodamento distribuito e cluster
- “Confronto tra cluster e accodamento distribuito” a pagina 29
- Componenti di un cluster

Quando si progetta il cluster del gestore code, è necessario prendere alcune decisioni. È necessario prima decidere quali gestori code nel cluster devono contenere i repository completi delle informazioni cluster. Qualsiasi gestore code creato può essere utilizzato in un cluster. È possibile scegliere un qualsiasi numero di gestori code per questo scopo, ma il numero ideale è due. Per informazioni sulla selezione dei gestori code per contenere i repository completi, consultare “Come scegliere i gestori code del cluster per conservare i repository completi” a pagina 32.

Per ulteriori informazioni sulla progettazione del cluster, consultare i seguenti argomenti:

- “Cluster di esempio” a pagina 38
- “Organizzazione di un cluster” a pagina 33
- “Convenzioni di denominazione cluster” a pagina 33
- z/OS “Queue sharing groups and clusters” a pagina 35
- “Cluster sovrapposti” a pagina 35

Operazioni successive

Consultare i seguenti argomenti per ulteriori informazioni sulla configurazione e l'utilizzo dei cluster:

- Come stabilire la comunicazione in un cluster
- Configurazione di un cluster di gestori code
- Instradamento dei messaggi verso e dai cluster
- Utilizzo dei cluster per la gestione del workload

Per ulteriori informazioni che ti aiutano a configurare il tuo cluster, vedi“Suggerimenti per il clustering” a pagina 36.

Pianificazione della modalità di utilizzo di più code di trasmissione cluster

È possibile definire esplicitamente le code di trasmissione o fare in modo che il sistema crei le code di trasmissione. Se si definiscono le code di trasmissione, si ha un maggiore controllo sulle definizioni delle code. z/OS Su z/OS, si ha anche un maggiore controllo sulla serie di pagine in cui sono contenuti i messaggi.

Definizione delle code di trasmissione

Esistono due metodi per definire le code di trasmissione:

- Automaticamente, utilizzando l'attributo DEFCLXQ del gestore code, come segue:

  ```
  ALTER QMGR DEFCLXQ(SCTQ | CHANNEL)
  ```

  DEFCLXQ (SCTQ) indica che la coda di trasmissione predefinita per tutti i canali mittente del cluster è SYSTEM.CLUSTER.TRANSMIT.QUEUE. Questo è il valore predefinito.

  DEFCLXQ (CHANNEL) indica che per impostazione predefinita ogni canale mittente del cluster utilizza una coda di trasmissione separata denominata SYSTEM.CLUSTER.TRANSMIT.*nome canale*. Ogni coda di trasmissione viene definita automaticamente dal gestore code. Per ulteriori informazioni, consultare “Code di trasmissione cluster definite automaticamente” a pagina 26.
- Manualmente, definendo una coda di trasmissione con un valore specificato per l'attributo CLCHNAME. L'attributo CLCHNAME indica quali canali mittente del cluster devono utilizzare la coda di trasmissione.

z/OS Se stai definendo manualmente una coda di trasmissione suz/OS , Vedere“Pianificazione di code di trasmissione cluster definite manualmente” a pagina 27 per maggiori informazioni.

Di quale sicurezza ho bisogno?

Per avviare uno switch, automaticamente o manualmente, è necessaria l'autorità per avviare un canale.

Per definire la coda utilizzata come coda di trasmissione, è necessaria l'autorizzazione IBM MQ standard per definire la coda.

Quando è il momento adatto per implementare la modifica?

Quando si modifica la coda di trasmissione utilizzata dai canali mittente del cluster, è necessario assegnare un tempo in cui effettuare l'aggiornamento, considerando i seguenti punti:

- Il tempo richiesto per un canale per commutare la coda di trasmissione dipende dal numero totale di messaggi sulla vecchia coda di trasmissione, dal numero di messaggi da spostare e dalla dimensione dei messaggi.
- Le applicazioni possono continuare a inserire i messaggi nella coda di trasmissione mentre si verifica la modifica. Ciò potrebbe portare ad un aumento del tempo di transizione.
- È possibile modificare il parametro CLCHNAME di qualsiasi coda di trasmissione o DEFCLXQ in qualsiasi momento, preferibilmente quando il carico di lavoro è basso.

  Si noti che nulla accade immediatamente.
- Le modifiche si verificano solo quando un canale viene avviato o riavviato. Quando un canale viene avviato, controlla la configurazione corrente e, se necessario, passa a una nuova coda di trasmissione.
- Esistono diverse modifiche che potrebbero modificare l'associazione di un canale mittente del cluster con una coda di trasmissione:
  - Modifica del valore dell'attributo CLCHNAME di una coda di trasmissione, rendendo CLCHNAME meno specifico o vuoto.
  - Modifica del valore dell'attributo CLCHNAME di una coda di trasmissione, rendendo CLCHNAME più specifico.
  - Eliminazione di una coda con CLCHNAME specificato.
  - Modifica dell'attributo del gestore code DEFCLXQ.

Quanto tempo ci vorrà per l'interruttore?

Durante il periodo di transizione, tutti i messaggi per il canale vengono spostati da una coda di trasmissione ad un'altra. Il tempo richiesto per un canale per commutare la coda di trasmissione dipende dal numero totale di messaggi sulla vecchia coda di trasmissione e dal numero di messaggi che devono essere spostati.

Per le code contenenti alcune migliaia di messaggi, lo spostamento dei messaggi dovrebbe richiedere meno di un secondo. Il tempo effettivo dipende dal numero e dalla dimensione dei messaggi. Il gestore code deve essere in grado di spostare i messaggi a molti megabyte al secondo.

Le applicazioni possono continuare a inserire i messaggi nella coda di trasmissione mentre si verifica la modifica. Ciò potrebbe portare ad un aumento del tempo di transizione.

Ogni canale mittente del cluster interessato deve essere riavviato per rendere effettiva la modifica. Pertanto, si consiglia di modificare la configurazione della coda di trasmissione quando il gestore code non è occupato e pochi messaggi sono memorizzati nelle code di trasmissione del cluster.

IL**runswchl** comando z/OS o il Comando SWITCH CHANNEL(*) STATO in CSQUTIL SUz/OS può essere utilizzato per interrogare lo stato dei canali del mittente del cluster e quali modifiche in sospeso sono in sospeso nella configurazione della coda di trasmissione.

Come implementare il cambiamento

Consultare Implementazione del sistema utilizzando più code di trasmissione cluster per dettagli su come apportare la modifica a più code di trasmissione cluster, automaticamente o manualmente.

Annullamento della modifica

z/OS

Consultare Annullamento di una modifica ad una coda di trasmissione su z/OS per i dettagli su come eseguire il backout delle modifiche in caso di problemi.

Code di trasmissione cluster definite automaticamente
È possibile che il sistema generi le code di trasmissione per conto dell'utente.

Prima di iniziare

z/OS Per impostare manualmente le code di trasmissione cluster su z/OS, consultare “Pianificazione di code di trasmissione cluster definite manualmente” a pagina 27.

Informazioni su questa attività

Se un canale non dispone di una coda di trasmissione del cluster definita manualmente associata e si specifica DEFCLXQ (CHANNEL), quando il canale avvia il gestore code definisce automaticamente una coda dinamica permanente per il canale mittente del cluster. Modello coda SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE viene utilizzato per definire automaticamente la coda di trasmissione del cluster dinamico permanente con il nome SYSTEM.CLUSTER.TRANSMIT.`ChannelName`.

Importante: z/OS In IBM MQ 8.0, il gestore code non ha `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE`. Non è possibile migrare direttamente da IBM MQ 8.0 a questa versione. Per informazioni sull'aggiunta di `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` in un gestore code migrato da IBM MQ 8.0, consultare questo argomento nella documentazione per la versione provvisoria utilizzata per migrare il gestore code.

Procedura

1. Utilizzare l'attributo del gestore code *DEFCLXQ* .

   Per ulteriori informazioni su questo attributo, consultare ALTER QMGR.

   Ci sono due opzioni:

   Ctcs
   Questa opzione è quella predefinita e indica che si utilizza il singolo SYSTEM.CLUSTER.TRANSMIT.QUEUE.

   CHANNEL
   Indica che si utilizzano più code di trasmissione cluster.
2. Per passare alla nuova associazione:
   - Arrestare e riavviare il canale.
   - Il canale utilizza la nuova definizione della coda di trasmissione.
   - I messaggi vengono trasferiti da un processo di commutazione di transizione dalla vecchia coda alla nuova coda di trasmissione.

   Tenere presente che tutti i messaggi dell'applicazione vengono inseriti nella vecchia definizione.

   Quando il numero di messaggi nella vecchia coda raggiunge lo zero, i nuovi messaggi vengono inseriti direttamente nella nuova coda di trasmissione.
3. Per controllare quando termina il processo di commutazione:

a) Uno switch della coda di trasmissione avviato da un canale viene eseguito in background e l'amministratore può monitorare il log dei lavori del gestore code per determinare quando è stato completato.

b) Monitorare i messaggi nella registrazione lavoro per visualizzare l'avanzamento dello switch.

c) Per assicurarsi che solo i canali desiderati stiano utilizzando questa coda di trasmissione, immettere il comando DIS CLUSQMGR (*) dove, ad esempio, la proprietà della coda di trasmissione che definisce la coda di trasmissione è `APPQMGR.CLUSTER1.XMITQ`.

d) z/OS
Utilizzare il comando SWITCH CHANNEL (*) STATUS in CSQUTIL.

Questa opzione indica quali modifiche in sospeso sono in sospeso e quanti messaggi devono essere spostati tra le code di trasmissione.

Risultati

Sono state impostate le code o la coda di trasmissione del cluster.

Attività correlate

“Pianificazione di code di trasmissione cluster definite manualmente” a pagina 27
SUIBM MQ for z/OS , se definisci tu stesso le code di trasmissione, hai un maggiore controllo sulle definizioni e sul set di pagine in cui vengono conservati i messaggi.

Riferimenti correlati

Gestore code ALTER
VISUALIZZA CLUSQMGR

z/OS *Pianificazione di code di trasmissione cluster definite manualmente*
SUIBM MQ for z/OS , se definisci tu stesso le code di trasmissione, hai un maggiore controllo sulle definizioni e sul set di pagine in cui vengono conservati i messaggi.

Prima di iniziare

Per impostare automaticamente le code di trasmissione del cluster, vedere“Code di trasmissione cluster definite automaticamente” a pagina 26 .

Informazioni su questa attività

L'amministratore definisce manualmente una coda di trasmissione e utilizza l'attributo della coda CLCHNAME per definire quale canale o canali del mittente del cluster utilizzeranno questa coda come coda di trasmissione.

Tenere presente che CLCHNAME può includere un carattere jolly all'inizio o alla fine, per consentire l'utilizzo di una singola coda per più canali.

Procedura

1. Ad esempio, immettere quanto segue:

```
DEFINE QLOCAL(APPQMGR.CLUSTER1.XMITQ)
CLCHNAME(CLUSTER1.TO.APPQMGR)
USAGE(XMITQ) STGCLASS(STG1)
INDXTYPE( CORRELID ) SHARE

DEFINE STGCLASS(STG1) PSID(3)
DEFINE PSID(3) BUFFERPOOL(4)
```

Suggerimento: È necessario pianificare la serie di pagine (e il pool di buffer) da utilizzare per le code di trasmissione. È possibile avere diversi set di pagine per code diverse e fornire isolamento tra di loro,

in modo che il riempimento di un set di pagine non influisca sulle code di trasmissione in altri set di pagine.

Consultare Gestione delle code di trasmissione del cluster e dei canali mittenti del cluster per informazioni su come ciascun canale seleziona la coda appropriata.

Quando il canale viene avviato, passa la sua associazione alla nuova coda di trasmissione. Per garantire che nessun messaggio vada perso, il gestore code trasferisce automaticamente i messaggi dalla vecchia coda di trasmissione del cluster alla nuova coda di trasmissione in ordine.

2. Utilizzare la funzione CSQUTIL SWITCH per passare alla nuova associazione.

   Per ulteriori informazioni, consultare Commutare la coda di trasmissione associata ai canali mittente del cluster (SWITCH) .

   a) Arrestare il canale, o i canali, la cui coda di trasmissione deve essere modificata, in modo che si trovino nello stato ARRESTATO.

      Ad esempio:

      ```
      STOP CHANNEL(CLUSTER1.TO.APPQMGR)
      ```

   b) Modificare l'attributo `CLCHNAME(XXXX)` nella coda di trasmissione.

   c) Utilizzare la funzione SWITCH per commutare i messaggi o monitorare ciò che sta accadendo.

      Utilizzare il comando

      ```
      SWITCH CHANNEL(*) MOVEMSGS(YES)
      ```

      per spostare i messaggi senza avviare il canale.

   d) Avviare il canale, o i canali, e verificare se il canale sta utilizzando le code corrette.

      Ad esempio:

      ```
      DIS CHS(CLUSTER1.TO.APPQMGR)
      DIS CHS(*) where(XMITQ eq APPQMGR.CLUSTER1.XMITQ)
      ```

   Suggerimento: Il seguente processo utilizza la funzione CSQUTIL SWITCH. Per ulteriori informazioni, vedere Cambia la coda di trasmissione associata ai canali mittente del cluster (SWITCH) .

   Non è necessario utilizzare questa funzione, ma l'utilizzo di questa funzione fornisce più opzioni:

   - L'utilizzo di SWITCH CHANNEL (*) STATUS fornisce un metodo semplice per identificare lo stato di commutazione dei canali mittente del cluster. Consente all'amministratore di visualizzare quali canali sono attualmente in fase di commutazione e i canali con uno switch in sospeso che diventano effettivi al successivo avvio di tali canali.

     Senza questa funzione, l'amministratore deve utilizzare più comandi DISPLAY, quindi elaborare l'output risultante per verificare queste informazioni. L'amministratore può anche confermare che una modifica della configurazione ha il risultato richiesto.
   - Se CSQUTIL viene utilizzato per avviare lo switch, CSQUTIL continua a monitorare l'avanzamento di questa operazione e termina solo quando lo switch è stato completato.

     Ciò può rendere molto più semplice l'esecuzione di queste operazioni in batch. Inoltre, se CSQUTIL viene eseguito per commutare più canali, CSQUTIL esegue queste azioni in modo sequenziale; ciò può avere un impatto minore per l'azienda rispetto a più switch in esecuzione in parallelo.

Risultati

Hai impostato la coda o le code di trasmissione del cluster suz/OS .

Controllo accessi e code di trasmissione di più cluster

Scegliere tra tre modalità di controllo quando un'applicazione inserisce i messaggi nelle code cluster remote. Le modalità sono la verifica in remoto rispetto alla coda del cluster, la verifica in locale rispetto

a SYSTEM.CLUSTER.TRANSMIT.QUEUEo la verifica rispetto ai profili locali per la coda del cluster o il gestore code del cluster.

IBM MQ consente di verificare localmente, o localmente e in remoto, che un utente disponga dell'autorizzazione per inserire un messaggio in una coda remota. Un'applicazione tipica IBM MQ utilizza solo il controllo locale e si basa sul gestore code remoto che considera attendibili i controlli di accesso effettuati sul gestore code locale. Se non viene utilizzato il controllo remoto, il messaggio viene inserito nella coda di destinazione con l'autorizzazione del processo del canale dei messaggi remoto. Per utilizzare il controllo remoto, è necessario impostare l'autorizzazione di inserimento del canale di ricezione sulla sicurezza del contesto.

I controlli locali vengono eseguiti rispetto alla coda aperta dall'applicazione. Nell'accodamento distribuito, l'applicazione di solito apre una definizione di coda remota e vengono effettuati controlli di accesso rispetto alla definizione di coda remota. Se il messaggio viene inserito con un'intestazione di instradamento completa, i controlli vengono eseguiti sulla coda di trasmissione. Se un'applicazione apre una coda cluster che non è sul gestore code locale, non vi è alcun oggetto locale da controllare. I controlli del controllo accessi vengono effettuati rispetto alla coda di trasmissione del cluster, SYSTEM.CLUSTER.TRANSMIT.QUEUE. Anche con più code di trasmissione del cluster, i controlli del controllo dell'accesso locale per le code del cluster remoto vengono eseguiti su SYSTEM.CLUSTER.TRANSMIT.QUEUE.

La scelta del controllo locale o remoto è una scelta tra due estremi. La verifica in remoto è dettagliata. Ogni utente deve disporre di un profilo di controllo accessi su ogni gestore code nel cluster per poter essere inserito in qualsiasi coda cluster. La verifica locale è generica. Ogni utente necessita di un solo profilo di controllo accessi per la coda di trasmissione del cluster sul gestore code a cui è connesso. Con tale profilo, possono inserire un messaggio in qualsiasi coda del cluster su qualsiasi gestore code in qualsiasi cluster.

Gli amministratori hanno un altro modo per impostare il controllo accessi per code cluster. È possibile creare un profilo di sicurezza per una coda del cluster su qualsiasi gestore code nel cluster utilizzando il comando **setmqaut** . Il profilo ha effetto se si apre localmente una coda del cluster remoto, specificando solo il nome della coda. È inoltre possibile impostare un profilo per un gestore code remoto. In questo caso, il gestore code può controllare il profilo di un utente che apre una coda cluster fornendo un nome completo.

I nuovi profili funzionano solo se si modifica la stanza del gestore code, **ClusterQueueAccessControl** in RQMName. Il valore predefinito è Xmitq. È necessario creare profili per tutte le applicazioni esistenti delle code cluster che utilizzano code cluster. Se si modifica la stanza in RQMName senza creare profili, è probabile che le applicazioni abbiano esito negativo.

Suggerimento: Il controllo dell'accesso alla coda del cluster non è valido per l'accodamento remoto. I controlli di accesso vengono ancora eseguiti rispetto alle definizioni locali. Le modifiche indicano che è possibile seguire lo stesso approccio per configurare il controllo accessi sulle code cluster e sugli argomenti cluster. z/OS Le modifiche allineano inoltre l'approccio di controllo dell'accesso per le code cluster più strettamente con z/OS. I comandi per impostare il controllo accessi su z/OS sono diversi, ma entrambi controllano l'accesso rispetto a un profilo piuttosto che all'oggetto stesso.

Concetti correlati

“Clustering: isolamento dell'applicazione utilizzando più code di trasmissione cluster” a pagina 47
È possibile isolare i flussi di messaggi tra gestori code in un cluster. È possibile inserire i messaggi trasportati da canali mittenti del cluster differenti in code di trasmissione cluster differenti. È possibile utilizzare l'approccio in un singolo cluster o con cluster sovrapposti. L'argomento fornisce esempi e alcune procedure ottimali per guidare l'utente nella scelta di un approccio da utilizzare.

Attività correlate

Impostando ClusterQueueAccessControl

Confronto tra cluster e accodamento distribuito

Confrontare i componenti che devono essere definiti per connettere i gestori code utilizzando l'accodamento distribuito e il cluster.

Se non si utilizzano i cluster, i gestori code sono indipendenti e comunicano utilizzando l'accodamento distribuito. Se un gestore code deve inviare messaggi a un altro gestore code, è necessario definire:

- Una coda di trasmissione
- Un canale per il gestore code remoto

Figura 3 a pagina 30 mostra i componenti richiesti per l'accodamento distribuito.

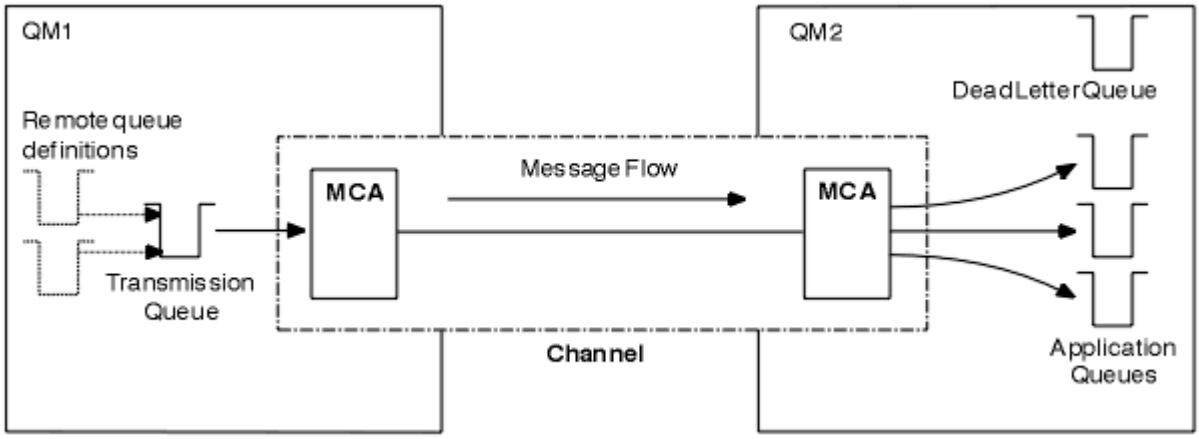


Figura 3. accodamento distribuito

Se si raggruppano i gestori code in un cluster, le code su qualsiasi gestore code sono disponibili per qualsiasi altro gestore code nel cluster. Qualsiasi gestore code può inviare un messaggio a qualsiasi altro gestore code nello stesso cluster senza definizioni esplicite. Non vengono fornite definizioni di canale, definizioni di coda remota o code di trasmissione per ciascuna destinazione. Ogni gestore code in un cluster ha una singola coda di trasmissione da cui può trasmettere messaggi a qualsiasi altro gestore code nel cluster. Ogni gestore code in un cluster deve definire solo:

- Un canale ricevente del cluster su cui ricevere i messaggi
- Un canale mittente del cluster con cui si introduce e impara a conoscere il cluster

Definizioni per impostare un cluster rispetto all'accodamento distribuito

Consultare Figura 4 a pagina 30, che mostra quattro gestori code ciascuno con due code. Considerare quante definizioni sono necessarie per connettere questi gestori code utilizzando l'accodamento distribuito. Confrontare quante definizioni sono necessarie per impostare la stessa rete di un cluster.

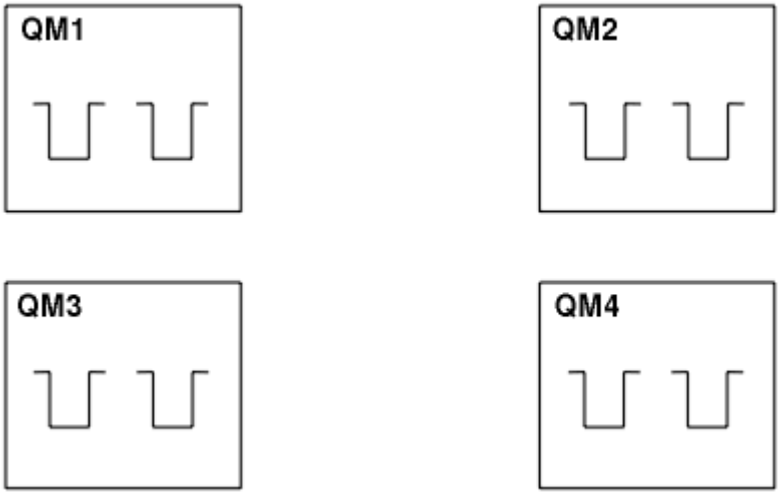


Figura 4. Una rete di quattro gestori code

Definizioni per impostare una rete utilizzando l'accodamento distribuito

Per impostare la rete mostrata in Figura 3 a pagina 30 utilizzando l'accodamento distribuito, è possibile disporre delle seguenti definizioni:

Tabella 2. Definizioni per l'accodamento distribuito

Descrizione	Numero per gestore code	Numero totale
Una definizione di canale mittente per un canale su cui inviare messaggi a ogni altro gestore code	3	12
Una definizione di canale ricevente per un canale su cui ricevere i messaggi da ogni altro gestore code	3	12
Una definizione della coda di trasmissione per una coda di trasmissione per ogni altro gestore code	3	12
Una definizione di coda locale per ciascuna coda locale	2	8
Una definizione di coda remota per ciascuna coda remota in cui questo gestore code desidera inserire i messaggi	6	24

È possibile ridurre questo numero di definizioni utilizzando definizioni generiche di canale ricevente. Il numero massimo di definizioni può essere pari a 17 su ciascun gestore code, che è un totale di 68 per questa rete.

Definizioni per configurare una rete utilizzando i cluster

Per configurare la rete mostrata in Figura 3 a pagina 30 utilizzando cluster sono necessarie le seguenti definizioni:

Tabella 3. Definizioni per il clustering

Descrizione	Numero per gestore code	Numero totale
Una definizione di canale mittente del cluster per un canale su cui inviare i messaggi a un gestore code del repository	1	4
Una definizione di canale ricevente del cluster per un canale su cui ricevere messaggi da altri gestori code nel cluster	1	4
Una definizione di coda locale per ciascuna coda locale	2	8

Per impostare questo cluster di gestori code (con due repository completi), sono necessarie quattro definizioni su ciascun gestore code, per un totale di sedici definizioni. È inoltre necessario modificare le definizioni dei gestori code per due gestori code, per renderli gestori code con repository completo per il cluster.

Sono richieste solo una definizione di canale CLUSSDR e una CLUSRCVR . Una volta definito il cluster, è possibile aggiungere o rimuovere i gestori code (diversi dai gestori code del repository) senza alcuna interruzione per gli altri gestori code.

L'utilizzo di un cluster riduce il numero di definizioni richieste per configurare una rete contenente molti gestori code.

Con meno definizioni da fare c'è meno rischio di errore:

- I nomi degli oggetti corrispondono sempre, ad esempio il nome del canale in una coppia mittente - destinatario.

- Il nome della coda di trasmissione specificato in una definizione di canale corrisponde sempre alla definizione della coda di trasmissione corretta o al nome della coda di trasmissione specificato in una definizione di coda remota.
- Una definizione QREMOTE punta sempre alla coda corretta sul gestore code remoto.

Una volta impostato un cluster, è possibile spostare le code del cluster da un gestore code a un altro all'interno del cluster senza dover eseguire alcuna attività di gestione del sistema su un altro gestore code. Non è possibile dimenticare di eliminare o modificare le definizioni di canale, coda remota o coda di trasmissione. È possibile aggiungere nuovi gestori code a un cluster senza alcuna interruzione della rete esistente.

Come scegliere i gestori code del cluster per conservare i repository completi

In ogni cluster è necessario scegliere almeno uno e preferibilmente due gestori code per contenere repository completi. Due archivi completi sono sufficienti per tutte le circostanze, tranne le più eccezionali. Se possibile, scegliere i gestori code ospitati su piattaforme robuste e connesse in modo permanente, che non hanno interruzioni coincidenti e che si trovano in una posizione centrale geograficamente. Considerare inoltre la possibilità di dedicare i sistemi come host di repository completi e di non utilizzare tali sistemi per altre attività.

I *repository completi* sono gestori code che mantengono un quadro completo dello stato del cluster. Per condividere queste informazioni, ogni repository completo è connesso dai canali CLUSSDR (e dalle corrispondenti definizioni CLUSRCVR) a ogni altro repository completo nel cluster. È necessario definire manualmente questi canali.

Figura 5. Due repository completi collegati.

Ogni altro gestore code nel cluster conserva un'immagine dello stato del cluster in un *repository parziale*. Questi gestori code pubblicano informazioni su se stessi e richiedono informazioni su altri gestori code, utilizzando due repository completi disponibili. Se un repository completo scelto non è disponibile, ne viene utilizzato un altro. Quando il repository completo scelto diventa di nuovo disponibile, raccoglie le informazioni nuove e modificate più recenti dagli altri in modo che mantengano il passo. Se tutti i repository completi non sono più in servizio, gli altri gestori code utilizzano le informazioni di cui dispongono nei repository parziali. Tuttavia, sono limitati all'utilizzo delle informazioni di cui dispongono; non è possibile elaborare nuove informazioni e richieste di aggiornamenti. Quando i repository completi si riconnettono alla rete, i messaggi vengono scambiati per aggiornare tutti i repository (completi e parziali).

Quando si pianifica l'assegnazione di repository completi, includere le seguenti considerazioni:

- I gestori code scelti per contenere repository completi devono essere affidabili e gestiti. Scegli i gestori code ospitati su una piattaforma solida e connessa in modo permanente.
- Considera le interruzioni pianificate per i sistemi che ospitano i tuoi repository completi e assicurati che non abbiano interruzioni coincidenti.
- Considerare le prestazioni di rete: scegliere i gestori code che si trovano in una posizione centrale geograficamente o che condividono lo stesso sistema di altri gestori code nel cluster.
- Considerare se un gestore code è un membro di più di un cluster. Può essere amministrativamente conveniente utilizzare lo stesso gestore code per ospitare i repository completi per diversi cluster, purché questo vantaggio sia bilanciato rispetto a quanto si prevede che il gestore code sia occupato.
- Considerare la possibilità di dedicare alcuni sistemi per contenere solo repository completi e non utilizzare questi sistemi per altre attività. Ciò garantisce che tali sistemi richiedano solo la manutenzione per la configurazione del gestore code e non vengano rimossi dal servizio per la manutenzione di altre applicazioni aziendali. Inoltre, garantisce che l'attività di gestione del repository non sia in competizione con le applicazioni per le risorse di sistema. Ciò può essere particolarmente utile nei cluster di grandi

dimensioni (ad esempio, i cluster di più di un migliaio di gestori code), in cui i repository completi hanno un carico di lavoro molto più elevato nel mantenere lo stato del cluster.

Avere più di due repository completi è possibile, ma raramente consigliato. Sebbene le definizioni degli oggetti (ossia code, argomenti e canali) fluiscano in tutti i repository completi disponibili, le richieste passano solo da un repository parziale a un massimo di due repository completi. Ciò significa che, quando vengono definiti più di due repository completi e due repository completi diventano non disponibili, alcuni repository parziali potrebbero non ricevere gli aggiornamenti previsti. Consultare ClusterMQ : perché solo due repository completi?

Una situazione in cui potrebbe essere utile definire più di due repository completi è quando si migrano i repository completi esistenti su un nuovo hardware o su nuovi gestori code. In questo caso, è necessario introdurre i repository completi di sostituzione e confermare che siano completamente popolati, prima di rimuovere i repository completi precedenti. Ogni volta che si aggiunge un repository completo, è necessario connetterlo direttamente a ogni altro repository completo con i canali CLUSSDR .

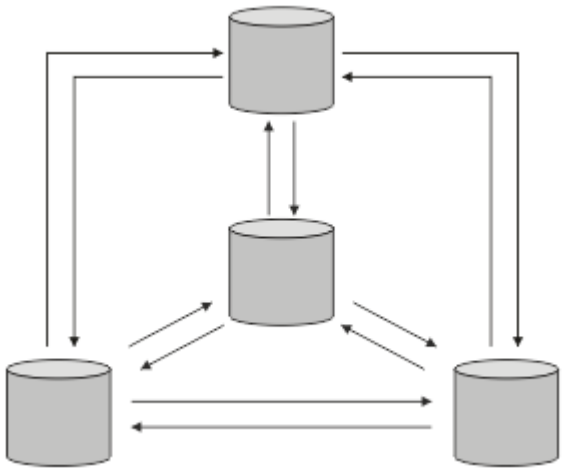


Figura 6. Più di due repository completi connessi

Informazioni correlate

Cluster MQ : perché solo due repository completi?

Quanto può essere grande un cluster MQ ?

Organizzazione di un cluster

Selezionare quali gestori code collegare e a quale repository completo. Considerare l'effetto delle prestazioni, la versione del gestore code e se sono desiderabili più canali CLUSSDR .

Dopo aver selezionato i gestori code per conservare i repository completi, è necessario decidere quali gestori code collegare a quale repository completo. La definizione del Canale CLUSSDR collega un gestore code a un repository completo da cui rileva gli altri repository completi nel cluster. Da quel momento in poi, il gestore code invia messaggi a due repository completi. Tenta sempre di utilizzare prima quello per cui ha una definizione di canale CLUSSDR . È possibile scegliere di collegare un gestore code a un repository completo. Nella scelta, considerare la topologia della propria configurazione e la posizione fisica o geografica dei gestori code.

Poiché tutte le informazioni sul cluster vengono inviate a due repository completi, potrebbero verificarsi situazioni in cui si desidera creare una seconda definizione di canale CLUSSDR . È possibile definire un secondo canale CLUSSDR in un cluster con molti repository completi distribuiti su una vasta area. È quindi possibile controllare a quali due repository completi vengono inviate le informazioni.

Convenzioni di denominazione cluster

Considerare la denominazione dei gestori code nello stesso cluster utilizzando una convenzione di denominazione che identifica il cluster a cui appartiene il gestore code. Utilizzare una convenzione di denominazione simile per i nomi canale ed estenderla per descrivere le caratteristiche del canale.

Procedure ottimali per la denominazione di cluster MQ

Sebbene i nomi cluster possano avere una lunghezza massima di 48 caratteri, i nomi cluster relativamente brevi sono utili quando si applicano le convenzioni di denominazione ad altri oggetti. Consultare “Procedure ottimali per la scelta dei nomi dei canali cluster” a pagina 34.

Quando si sceglie un nome cluster, di solito è utile rappresentare lo 'scopo' del cluster (che è probabile sia di lunga durata) piuttosto che il 'contenuto'. Ad esempio 'B2BPROD' o 'ACTTEST' invece di 'QM1_QM2_QM3_CLUS'.

Procedure ottimali nella scelta dei nomi dei gestori code del cluster

Se si sta creando un nuovo cluster e i relativi membri da zero, considerare una convenzione di denominazione per i gestori code che rifletta l'utilizzo del cluster. Ogni gestore code deve avere un nome differente. Tuttavia, è possibile fornire ai gestori code in un cluster una serie di nomi simili, per facilitare l'identificazione e la memorizzazione dei raggruppamenti logici (ad esempio, 'ACTTQM1, ACTTQM2).

I nomi dei gestori code relativamente brevi (ad esempio, meno di 8 caratteri) aiutano se si sceglie di utilizzare la convenzione descritta nella sezione successiva, o qualcosa di simile, per i nomi dei canali.

Procedure ottimali per la scelta dei nomi dei canali cluster

Poiché i gestori code e i cluster possono avere nomi fino a 48 caratteri e un nome canale è limitato a 20 caratteri, prestare attenzione quando si denominano gli oggetti per evitare di dover modificare la convenzione di denominazione a metà di un progetto (vedere la sezione precedente).

Quando si definiscono i canali, tenere presente che i canali mittenti del cluster creati automaticamente su qualsiasi gestore code del cluster prendono il loro nome dal canale ricevente del cluster corrispondente configurato sul gestore code ricevente nel cluster e devono pertanto essere univoci e avere senso *sui gestori code remoti nel cluster.*

Un approccio comune consiste nell'utilizzare il nome gestore code preceduto dal nome cluster. Ad esempio, se il nome del cluster è CLUSTER1 e i gestori code sono QM1, QM2, i canali riceventi del cluster sono CLUSTER1.QM1, CLUSTER1.QM2.

È possibile estendere questa convenzione se i canali hanno priorità differenti o utilizzano protocolli differenti. Ad esempio:

- CLUSTER1.QM1.S1
- CLUSTER1.QM1.N3
- CLUSTER1.QM1.T4

In questo esempio, S1 potrebbe essere il primo canale SNA, N3 potrebbe essere il canale NetBIOS con una priorità di rete di tre e T4 potrebbe essere un IP TCP utilizzando una rete IPV4 .

Denominazione delle definizioni di canale condiviso

Una singola definizione di canale può essere condivisa tra più cluster, nel qual caso le convenzioni di denominazione qui suggerite richiedono modifiche. Tuttavia, come descritto in Gestione delle definizioni di canale , di solito è preferibile definire canali discreti per ciascun cluster in ogni caso.

Convenzioni di denominazione dei canali meno recenti

Al di fuori degli ambienti cluster, è stato storicamente comune utilizzare una convenzione di denominazione 'FROMQM.TO.TARGETQM', quindi è possibile che i cluster esistenti abbiano utilizzato qualcosa di simile (come CLUSTER.TO.TARGET). Ciò non è consigliato come parte di un nuovo schema di denominazione del cluster in quanto riduce ulteriormente i caratteri disponibili per trasmettere informazioni 'utili' all'interno del nome del canale.

z/OS Nomi canale su IBM MQ for z/OS

È possibile definire le risorse generiche VTAM o i nomi generici DDNS (*Dynamic Domain Name Server*). È possibile definire i nomi di connessione utilizzando nomi generici. Tuttavia, quando si crea una definizione ricevente del cluster, non utilizzare un nome di connessione generico.

Il problema con l'utilizzo di nomi di connessione generici per le definizioni di ricevente del cluster è il seguente: se si definisce un CLUSRCVR con un CONNAME generico, non è possibile garantire che i canali CLUSSDR puntino ai gestori code desiderati. Il CLUSSDR iniziale potrebbe finire per puntare a qualsiasi gestore code nel gruppo di condivisione code, non necessariamente uno che ospita un repository completo. Se un canale inizia a tentare nuovamente una connessione, potrebbe riconnettersi a un gestore code differente con lo stesso nome generico, interrompendo il flusso di messaggi.

z/OS *Queue sharing groups and clusters*

Shared queues can be cluster queues and queue managers in a queue sharing group can also be cluster queue managers.

On IBM MQ for z/OS you can group queue managers into queue sharing groups. A queue manager in a queue sharing group can define a local queue that is to be shared by up to 32 queue managers.

Shared queues can also be cluster queues. Furthermore, the queue managers in a queue sharing group can also be in one or more clusters.

È possibile definire le risorse generiche VTAM o i nomi generici DDNS (*Dynamic Domain Name Server*). È possibile definire i nomi di connessione utilizzando nomi generici. Tuttavia, quando si crea una definizione ricevente del cluster, non utilizzare un nome di connessione generico.

Il problema con l'utilizzo di nomi di connessione generici per le definizioni di ricevente del cluster è il seguente: se si definisce un CLUSRCVR con un CONNAME generico, non è possibile garantire che i canali CLUSSDR puntino ai gestori code desiderati. Il CLUSSDR iniziale potrebbe finire per puntare a qualsiasi gestore code nel gruppo di condivisione code, non necessariamente uno che ospita un repository completo. Se un canale inizia a tentare nuovamente una connessione, potrebbe riconnettersi a un gestore code differente con lo stesso nome generico, interrompendo il flusso di messaggi.

A CLUSRCVR channel that uses the group listener port can not be started because, if this were the case, it would not be possible to tell which queue manager the CLUSRCVR would connect to each time. The cluster system queues on which information is kept about the cluster are not shared. Each queue manager has its own.

Cluster channels are used not only to transfer application messages but internal system messages about the setup of the cluster. Each queue manager in the cluster must receive these internal system messages to participate properly in clustering, so needs its own unique CLUSRCVR channel on which to receive them.

A shared CLUSRCVR could start on any queue manager in the queue sharing group (QSG) and so lead to an inconsistent supply of the internal system messages to the QSG queue managers, meaning none can properly participate in the cluster. To ensure no shared CLUSRCVR channels can be used, any attempt fails with the CSQX502E message.

Cluster sovrapposti

I cluster sovrapposti forniscono ulteriori funzioni di gestione. Utilizzare gli elenchi nomi per ridurre il numero di comandi necessari per gestire i cluster che si sovrappongono.

È possibile creare cluster che si sovrappongono. Ci sono una serie di motivi per cui è possibile definire i cluster che si sovrappongono; ad esempio:

- Per consentire alle diverse organizzazioni di avere la propria amministrazione.
- Per consentire la gestione separata delle applicazioni indipendenti.
- Per creare classi di servizio.

In Figura 7 a pagina 36, il gestore code STF2 è membro di entrambi i cluster. Quando un gestore code è membro di più di un cluster, è possibile sfruttare gli elenchi dei nomi per ridurre il numero di definizioni necessarie. Gli elenchi nomi contengono un elenco di nomi, ad esempio, nomi cluster. È possibile creare un elenco nomi che denomina i cluster. Specificare l'elenco nomi nel comando ALTER QMGR per STF2 per renderlo un gestore code del repository completo per entrambi i cluster.

Se hai più di un cluster nella tua rete, devi fornire loro nomi diversi. Se due cluster con lo stesso nome vengono uniti, non è possibile separarli di nuovo. È anche una buona idea dare ai cluster e ai canali nomi diversi. Sono più facilmente distinguibili quando si guarda l'output dei comandi DISPLAY . I nomi dei gestori code devono essere univoci all'interno di un cluster per poter funzionare correttamente.

Definizione di classi di servizio

Immagina un'università che abbia un gestore code per ogni membro del personale e ogni studente. I messaggi tra membri del personale devono viaggiare su canali con una priorità elevata e una larghezza di banda elevata. I messaggi tra gli studenti devono viaggiare su canali più economici e lenti. È possibile impostare questa rete utilizzando tecniche di accodamento distribuite tradizionali. IBM MQ seleziona quali canali utilizzare esaminando il nome della coda di destinazione e il nome del gestore code.

Per distinguere chiaramente tra il personale e gli studenti, è possibile raggruppare i relativi gestori code in due cluster, come mostrato in Figura 7 a pagina 36. IBM MQ sposta i messaggi nella coda delle riunioni nel cluster del personale solo sui canali definiti in quel cluster. I messaggi per la coda di gossip nel cluster di studenti passano attraverso i canali definiti in tale cluster e ricevono la classe di servizio appropriata.

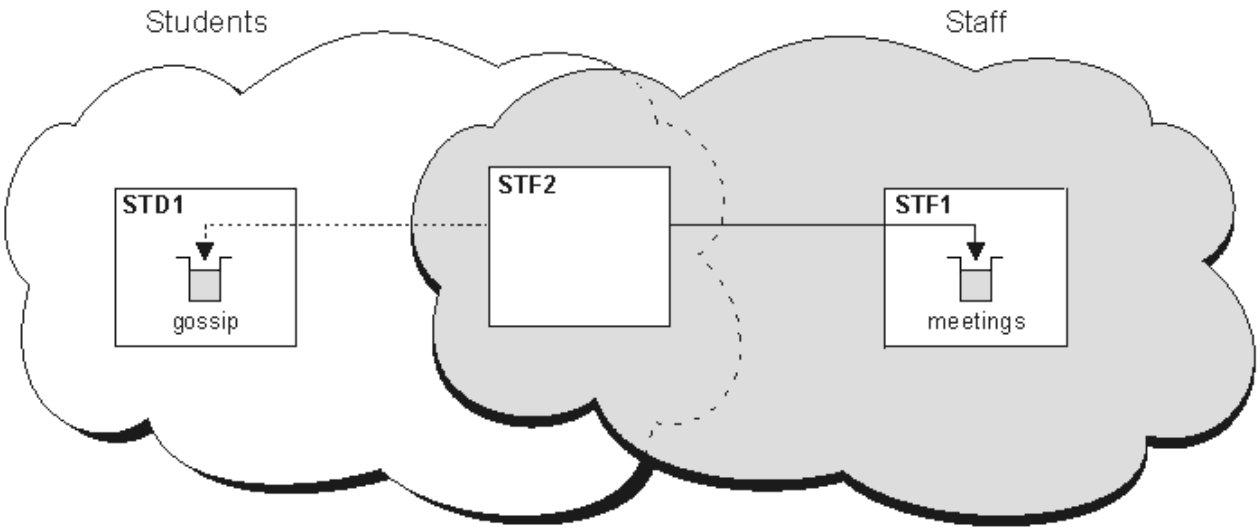


Figura 7. Classi di servizio

Suggerimenti per il clustering

Potrebbe essere necessario apportare alcune modifiche ai propri sistemi o applicazioni prima di utilizzare il clustering. Ci sono sia somiglianze che differenze dal comportamento dell'accodamento distribuito.

- È necessario aggiungere definizioni di configurazione manuali ai gestori code all'esterno di un cluster per consentire loro di accedere alle code cluster.
- Se si uniscono due cluster con lo stesso nome, non è possibile separarli di nuovo. Pertanto, è consigliabile assegnare a tutti i cluster un nome univoco.
- Se un messaggio arriva a un gestore code ma non vi è alcuna coda per riceverlo, il messaggio viene inserito nella coda di messaggi non recapitabili. Se non è presente una coda di messaggi non recapitabili, il canale ha esito negativo e riprova. L'utilizzo della coda di messaggi non recapitabili è uguale a quello della coda distribuita.
- L'integrità dei messaggi persistenti viene mantenuta. I messaggi non vengono duplicati o persi come risultato dell'utilizzo dei cluster.

- L'utilizzo di cluster riduce la gestione del sistema. I cluster semplificano la connessione di reti più grandi con molti più gestori code di quanti ne sarebbero in grado di considerare l'utilizzo dell'accodamento distribuito. Esiste il rischio che si consumino risorse di rete eccessive se si tenta di abilitare la comunicazione tra ogni gestore code in un cluster.
- Se si utilizza IBM MQ Explorer, che presenta i gestori code in una struttura ad albero, la vista per i cluster di grandi dimensioni potrebbe essere complessa.
- 

Lo scopo degli elenchi di distribuzione è utilizzare un singolo comando MQPUT per inviare lo stesso messaggio a più destinazioni. Gli elenchi di distribuzione sono supportati su IBM MQ for Multiplatforms. È possibile utilizzare elenchi di distribuzione con cluster di gestori code. In un cluster, tutti i messaggi vengono espansi all'ora MQPUT . Il vantaggio, in termini di traffico di rete, non è così grande come in un ambiente non cluster. Il vantaggio delle liste di distribuzione è che i numerosi canali e code di trasmissione non devono essere definiti manualmente.
- Se stai per utilizzare i cluster per bilanciare il tuo carico di lavoro, esamina le tue applicazioni. Verificare se i messaggi devono essere elaborati da un particolare gestore code o in una particolare sequenza. Si dice che tali applicazioni abbiano affinità di messaggi. Potrebbe essere necessario modificare le applicazioni prima di poterle utilizzare in cluster complessi.
- È possibile scegliere di utilizzare l'opzione MQOO_BIND_ON_OPEN su un MQOPEN per forzare l'invio di messaggi a una destinazione specifica. Se il gestore code di destinazione non è disponibile, i messaggi non vengono consegnati finché il gestore code non diventa nuovamente disponibile. I messaggi non vengono instradati a un altro gestore code a causa del rischio di duplicazione.
- Se un gestore code deve ospitare un repository del cluster, è necessario conoscerne il nome host o l'indirizzo IP. È necessario specificare queste informazioni nel parametro CONNAME quando si crea la definizione CLUSSDR su altri gestori code che si uniscono al cluster. Se si utilizza DHCP, l'indirizzo IP è soggetto a modifica in quanto DHCP può assegnare un nuovo indirizzo IP ogni volta che si riavvia un sistema. Pertanto, non è necessario specificare l'indirizzo IP nelle definizioni CLUSSDR . Anche se tutte le definizioni CLUSSDR specificano il nome host piuttosto che l'indirizzo IP, le definizioni non sarebbero comunque affidabili. DHCP non aggiorna necessariamente la voce della directory DNS per l'host con il nuovo indirizzo. Se è necessario denominare i gestori code come repository completi sui sistemi che utilizzano DHCP, installare il software che garantisce l'aggiornamento della directory DNS.
- Non utilizzare nomi generici, ad esempio risorse generiche VTAM o nomi generici DDNS (Dynamic Domain Name Server) come nomi di connessione per i propri canali. In tal caso, i canali potrebbero connettersi a un gestore code diverso da quello previsto.
- È possibile ottenere un messaggio solo da una coda cluster locale, ma è possibile inserire un messaggio in qualsiasi coda in un cluster. Se si apre una coda per utilizzare il comando MQGET , il gestore code apre la coda locale.
- Non è necessario modificare alcuna delle applicazioni se si imposta un cluster IBM MQ semplice. L'applicazione può denominare la coda di destinazione sulla chiamata MQOPEN e non è necessario conoscere l'ubicazione del gestore code. Se si configura un cluster per la gestione del carico di lavoro, è necessario esaminare le applicazioni e modificarle in base alle esigenze.
- È possibile visualizzare i dati di monitoraggio e di stato correnti per un canale o una coda utilizzando i comandi DISPLAY CHSTATUS e DISPLAY QSTATUS **runmqsc** . Le informazioni di controllo possono essere utilizzate per misurare le prestazioni e l'integrità del sistema. Il monitoraggio è controllato dagli attributi gestore code, coda e canale. Il monitoraggio dei canali mittenti del cluster definiti automaticamente è possibile con l'attributo del gestore code MONACLS .

Concetti correlati

Cluster

“Confronto tra cluster e accodamento distribuito” a pagina 29
Confrontare i componenti che devono essere definiti per connettere i gestori code utilizzando l'accodamento distribuito e il cluster.

Componenti di un cluster

Attività correlate

Configurazione di un cluster di gestore code

Per quanto tempo i repository dei gestori code conservano le informazioni?

I repository del gestore code conservano le informazioni per 30 giorni. Un processo automatico aggiorna in modo efficiente le informazioni utilizzate.

Quando un gestore code invia alcune informazioni su se stesso, i gestori code del repository completo e parziale memorizzano le informazioni per 30 giorni. Le informazioni vengono inviate, ad esempio, quando un gestore code annuncia la creazione di una nuova coda. Per impedire la scadenza di queste informazioni, i gestori code inviano automaticamente tutte le informazioni su se stessi dopo 27 giorni. Se un repository parziale invia una nuova richiesta di informazioni in parte per tutta la durata di 30 giorni, il tempo di scadenza rimane quello originale di 30 giorni.

Quando le informazioni scadono, non vengono rimosse immediatamente dal repository. Invece è tenuto per un periodo di grazia di 60 giorni. Se non viene ricevuto alcun aggiornamento entro il periodo di tolleranza, le informazioni vengono rimosse. Il periodo di dilazione consente il fatto che un gestore code potrebbe essere temporaneamente fuori servizio alla data di scadenza. Se un gestore code viene disconnesso da un cluster per più di 90 giorni, smette di far parte del cluster. Tuttavia, se si riconnette alla rete, diventa di nuovo parte del cluster. I repository completi non utilizzano le informazioni scadute per soddisfare le nuove richieste provenienti da altri gestori code.

Allo stesso modo, quando un gestore code invia una richiesta di informazioni aggiornate da un repository completo, la richiesta dura 30 giorni. Dopo 27 giorni IBM MQ controlla la richiesta. Se è stato fatto riferimento ad esso durante i 27 giorni, viene aggiornato automaticamente. In caso contrario, viene lasciato scadere e viene aggiornato dal gestore code se è di nuovo necessario. La scadenza delle richieste impedisce la creazione di richieste di informazioni dai gestori code inattivi.

Nota: È necessario scaricare e installare la PTF per APAR PH43191, che corregge gli errori di sistema nel calcolo della scadenza di una sottoscrizione. Questi errori possono causare la scadenza anticipata della sottoscrizione (con conseguente emissione del messaggio CSQX456I) o la scadenza dopo la scadenza dell'oggetto (con conseguente errore MQRC 2085 (MQRC_UNKNOWN_OBJECT)).

Per i cluster di grandi dimensioni, può essere disruttivo se molti gestori code inviano automaticamente tutte le informazioni su se stessi contemporaneamente. Consultare Refreshing in a large cluster can affect performance and availability of the cluster.

Concetti correlati

“Cluster: utilizzo delle procedure consigliate per REFRESH CLUSTER” a pagina 70
Utilizzare il comando **REFRESH CLUSTER** per eliminare tutte le informazioni contenute localmente su un cluster e ricreare tali informazioni dai repository completi nel cluster. Non è necessario utilizzare questo comando, tranne in circostanze eccezionali. Se hai bisogno di usarlo, ci sono considerazioni speciali su come usarlo. Queste informazioni sono una guida basata sul test e sul feedback dei clienti.

Cluster di esempio

Il primo esempio mostra il cluster più piccolo possibile di due gestori code. Il secondo e il terzo esempio mostrano due versioni di un cluster di gestori code.

Il cluster più piccolo possibile contiene solo due gestori code. In questo caso, entrambi i gestori code contengono repository completi. Sono necessarie solo poche definizioni per configurare il cluster, ma esiste un alto grado di autonomia in ogni gestore code.

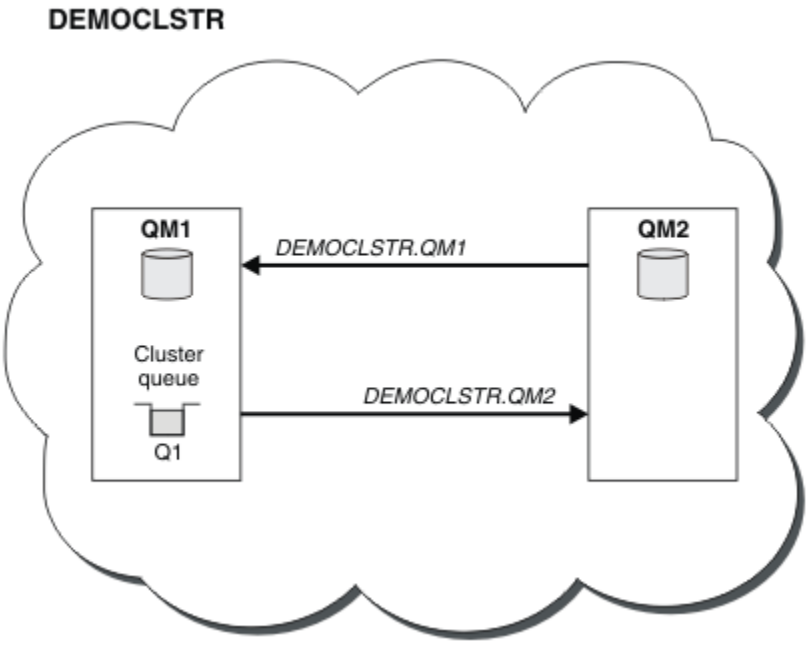


Figura 8. Un piccolo cluster di due gestori code

- I gestori code possono avere nomi lunghi come LONDON e NEWYORK. z/OS Su IBM MQ for z/OS, i nomi dei gestori code sono limitati a quattro caratteri.
- Ogni gestore code è generalmente configurato su una macchina separata. Tuttavia, è possibile avere più gestori code sulla stessa macchina.

Per istruzioni sulla configurazione di un cluster di esempio simile, consultare Configurazione di un nuovo cluster.

Figura 9 a pagina 40 mostra i componenti di un cluster denominato CLSTR1.

- In questo cluster, sono presenti tre gestori code, QM1, QM2e QM3.
- QM1 e QM2 ospitano repository di informazioni su tutti i gestori code e gli oggetti correlati al cluster nel cluster. Ad essi si fa riferimento come *gestori code repository completi*. I repository sono rappresentati nel diagramma dai cilindri ombreggiati.
- QM2 e QM3 ospitano alcune code accessibili a qualsiasi altro gestore code nel cluster. Le code accessibili a qualsiasi altro gestore code nel cluster vengono denominate *code cluster*. Le code cluster sono rappresentate nel diagramma dalle code ombreggiate. Le code cluster sono accessibili da qualsiasi punto del cluster. Il codice cluster IBM MQ garantisce che le definizioni di coda remota per le code cluster vengano create su qualsiasi gestore code a cui fanno riferimento.

  Come con l'accodamento distribuito, un'applicazione utilizza la chiamata MQPUT per inserire un messaggio su una coda del cluster in qualsiasi gestore code del cluster. Un'applicazione utilizza la chiamata MQGET per richiamare i messaggi da una coda cluster solo sul gestore code in cui risiede la coda.
- Ogni gestore code ha una definizione creata manualmente per l'estremità di ricezione di un canale denominato *cluster_name. queue_manager_name* su cui può ricevere messaggi. Sul gestore code di ricezione, *cluster_name. queue_manager_name* è un canale ricevente del cluster. Un canale ricevente del cluster è simile a un canale ricevente utilizzato nell'accodamento distribuito; riceve i messaggi per il gestore code. Inoltre, riceve anche informazioni sul cluster.

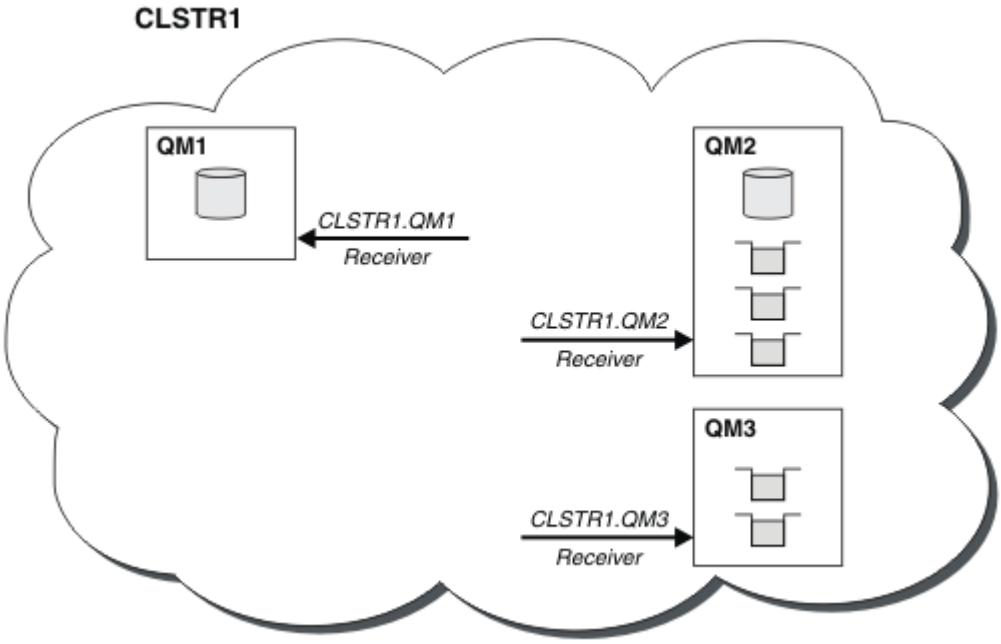


Figura 9. Un cluster di gestori code

- In Figura 10 a pagina 41 ogni gestore code ha anche una definizione per l'estremità di invio di un canale. Si collega al canale ricevente del cluster di uno dei gestori code del repository completo. Sul gestore code di invio, *cluster_name. queue_manager_name* è un canale mittente del cluster. QM1 e QM3 hanno canali mittenti del cluster che si collegano a CLSTR1.QM2, consultare la linea tratteggiata "2".

  QM2 ha un canale mittente del cluster che si connette a CLSTR1.QM1, vedi la riga tratteggiata "3". Un canale mittente del cluster è simile a un canale mittente utilizzato nell'accodamento distribuito; invia messaggi al gestore code di ricezione. Inoltre, invia anche informazioni sul cluster.

  Una volta definite sia l'estremità del ricevente del cluster che l'estremità del mittente del cluster di un canale, il canale viene avviato automaticamente.

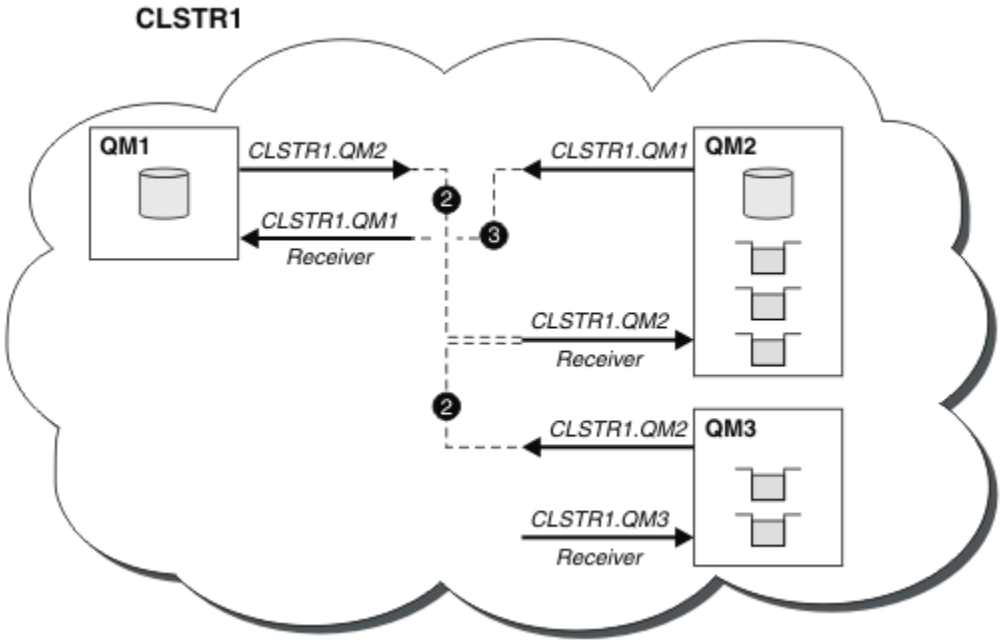


Figura 10. Un cluster di gestori code con canali mittente

La definizione di un canale mittente del cluster sul gestore code locale introduce tale gestore code a un gestore code del repository completo. Il gestore code del repository completo aggiorna le informazioni nel repository completo di conseguenza. Quindi crea automaticamente un canale mittente del cluster al gestore code originale e invia tali informazioni sul cluster. Pertanto, un gestore code apprende informazioni su un cluster e un cluster apprende informazioni su un gestore code.

Esaminare nuovamente Figura 9 a pagina 40. Si supponga che un'applicazione connessa al gestore code QM3 desideri inviare alcuni messaggi alle code in QM2. La prima volta che QM3 deve accedere a tali code, le rileva consultando un repository completo. Il repository completo in questo caso è QM2, a cui si accede utilizzando il canale mittente CLSTR1.QM2. Con le informazioni del repository, è possibile creare automaticamente definizioni remote per tali code. Se le code si trovano su QM1, questo meccanismo funziona ancora, poiché QM2 è un repository completo. Un repository completo ha un record completo di tutti gli oggetti nel cluster. In questo ultimo caso, QM3 crea automaticamente anche un canale mittente del cluster corrispondente al canale ricevente del cluster su QM1, consentendo la comunicazione diretta tra i due.

La Figura 11 a pagina 42 mostra lo stesso cluster, con i due canali mittente del cluster che sono stati creati automaticamente. I canali mittenti del cluster sono rappresentati dalle due linee tratteggiate che si uniscono con il canale ricevente del cluster CLSTR1.QM3. Mostra anche la coda di trasmissione del cluster, SYSTEM.CLUSTER.TRANSMIT.QUEUE, che QM1 utilizza per inviare i propri messaggi. Tutti i gestori code nel cluster hanno una coda di trasmissione cluster, da cui possono inviare messaggi a qualsiasi altro gestore code nello stesso cluster.

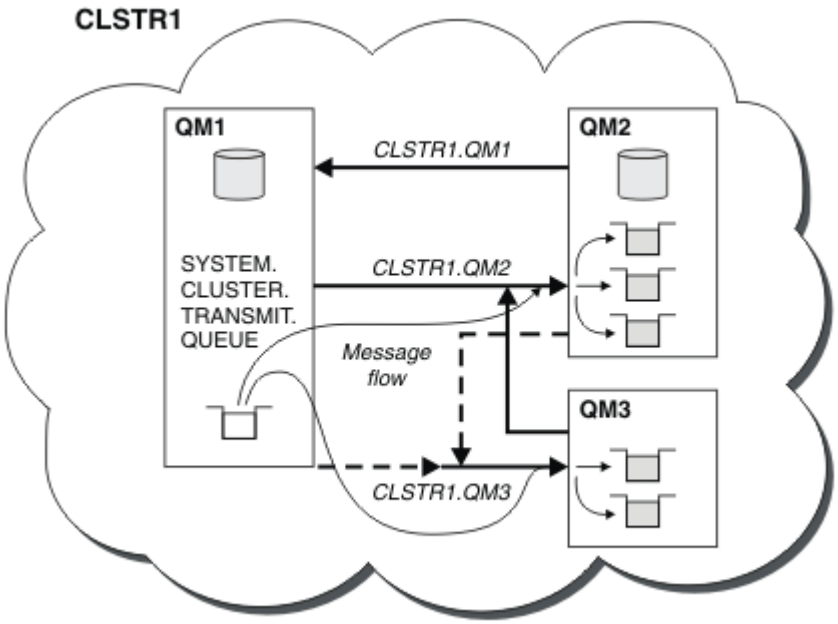


Figura 11. Un cluster di gestori code, che mostra canali definiti automaticamente

Nota: Altri diagrammi mostrano solo le estremità di ricezione dei canali per cui si effettuano definizioni manuali. Le estremità di invio vengono omesse perché sono per lo più definite automaticamente quando necessario. La definizione automatica della maggior parte dei canali mittente del cluster è fondamentale per la funzione e l'efficienza dei cluster.

Concetti correlati

“Confronto tra cluster e accodamento distribuito” a pagina 29
Confrontare i componenti che devono essere definiti per connettere i gestori code utilizzando l'accodamento distribuito e il cluster.

Componenti di un cluster

Attività correlate

Configurazione di un cluster di gestore code

Configurazione di un nuovo cluster

Clustering: procedure ottimali

I cluster forniscono un meccanismo per l'interconnessione dei gestori code. Le migliori pratiche descritte in questa sezione si basano su test e feedback da parte dei clienti.

Una corretta configurazione del cluster dipende da una buona pianificazione e da una conoscenza approfondita dei fondamentali di IBM MQ , come una buona gestione dell'applicazione e la progettazione della rete. Accertarsi di conoscere le informazioni contenute negli argomenti correlati prima di continuare.

Concetti correlati

Accodamento distribuito e cluster

Cluster

Attività correlate

“Progettazione di cluster” a pagina 23
I cluster forniscono un meccanismo per l'interconnessione dei gestori code in modo da semplificare sia la configurazione iniziale che la gestione in corso. I cluster devono essere attentamente progettati per garantire che funzionino correttamente e che raggiungano i livelli richiesti di disponibilità e reattività.

Monitoraggio dei cluster

Clustering: considerazioni speciali per i cluster che si sovrappongono
Questo argomento fornisce una guida per la pianificazione e la gestione dei cluster IBM MQ . Queste informazioni sono una guida basata sul test e sul feedback dei clienti.

Proprietà del cluster

Acquisire dimestichezza con i cluster sovrapposti prima di leggere le seguenti informazioni. Per le informazioni necessarie, consultare “Cluster sovrapposti” a pagina 35 e Configurazione dei percorsi dei messaggi tra i cluster .

Quando si configura e si gestisce un sistema costituito da cluster sovrapposti, è preferibile attenersi a quanto segue:

- Sebbene i cluster IBM MQ siano 'liberamente accoppiati' come precedentemente descritto, è utile considerare un cluster come una singola unità di amministrazione. Questo concetto viene utilizzato perché l'interazione tra definizioni su singoli gestori code è fondamentale per il regolare funzionamento del cluster. Ad esempio: quando si utilizzano le code cluster bilanciate del carico di lavoro, è importante che un singolo amministratore o team comprenda la serie completa di destinazioni possibili per i messaggi, che dipende dalle definizioni diffuse in tutto il cluster. Più banalmente, le coppie di canali mittente / ricevente del cluster devono essere compatibili.
- Considerando questo concetto precedente, in cui si incontrano più cluster (che devono essere gestiti da team / individui separati), è importante disporre di politiche chiare che controllino l'amministrazione dei gestori code del gateway.
- È utile considerare i cluster sovrapposti come un singolo spazio dei nomi: i nomi dei canali e dei gestori code devono essere univoci in un singolo cluster. La gestione è molto più semplice quando è univoca nell'intera topologia. È consigliabile seguire una convenzione di denominazione appropriata, le convenzioni possibili sono descritte in “Convenzioni di denominazione cluster” a pagina 33.
- A volte la cooperazione amministrativa e di gestione del sistema è essenziale. Ad esempio, la cooperazione tra organizzazioni che possiedono cluster differenti che devono sovrapporsi. Una chiara comprensione di chi possiede cosa e le regole e le convenzioni applicabili aiuta il clustering a funzionare senza problemi quando si sovrappongono i cluster.

Sovrapposizione di cluster: gateway

In generale, un singolo cluster è più facile da gestire rispetto a più cluster. Pertanto, la creazione di un gran numero di piccoli cluster (uno per ogni applicazione, ad esempio) è qualcosa da evitare in generale.

Tuttavia, per fornire classi di servizio, è possibile implementare cluster sovrapposti. Ad esempio:

- Se si dispone di cluster concentrici in cui il più piccolo è per la pubblicazione / sottoscrizione. Per ulteriori informazioni, consultare Come dimensionare i sistemi .
- Se alcuni gestori code devono essere gestiti da team differenti. Consultare la sezione precedente “Proprietà del cluster” a pagina 43per ulteriori informazioni.
- Se ha senso dal punto di vista organizzativo o geografico.
- Se i cluster equivalenti funzionano con la risoluzione dei nomi, ad esempio quando si implementano TLS in un cluster esistente.

Non vi è alcun vantaggio per la sicurezza derivante dalla sovrapposizione di cluster; consentendo ai cluster gestiti da due diversi team di sovrapporsi, si uniscono efficacemente ai team e alla topologia:

- Qualsiasi nome pubblicizzato in un cluster di questo tipo è accessibile all'altro cluster.
- Qualsiasi nome pubblicizzato in un cluster può essere pubblicizzato nell'altro per estrarre i messaggi idonei.
- Qualsiasi oggetto non pubblicizzato su un gestore code adiacente al gateway può essere risolto da qualsiasi cluster di cui il gateway è membro.

Il namespace è l'unione di entrambi i cluster e deve essere considerato come un singolo namespace. Pertanto, la proprietà di un cluster sovrapposto viene condivisa tra tutti gli amministratori di entrambi i cluster.

Quando un sistema contiene più cluster, potrebbe essere necessario instradare i messaggi dai gestori code in un cluster alle code sui gestori code in un altro cluster. In questa situazione, i cluster multipli devono essere interconnessi in qualche modo: un buon modello da seguire è l'utilizzo dei gestori code gateway tra i cluster. Questa disposizione evita di creare una rete difficile da gestire di canali point - to - point e fornisce un buon posto per gestire questioni quali le politiche di sicurezza. Ci sono due modi distinti per raggiungere questo accordo:

1. Inserire uno o più gestori code in entrambi i cluster utilizzando una seconda definizione del destinatario del cluster. Questa disposizione comporta un minor numero di definizioni amministrative, ma, come precedentemente affermato, significa che la proprietà di un cluster sovrapposto è condivisa tra tutti gli amministratori di entrambi i cluster.
2. Associare un gestore code nel cluster uno con un gestore code nel cluster due utilizzando i canali point-to-point tradizionali.

In entrambi i casi, è possibile utilizzare vari strumenti per instradare il traffico in modo appropriato. In particolare, gli alias della coda o del gestore code possono essere utilizzati per eseguire l'instradamento nell'altro cluster e un alias del gestore code con la proprietà **RQMNAME** vuota riguida il bilanciamento del carico di lavoro nel punto desiderato.

Concetti correlati

“Convenzioni di denominazione cluster” a pagina 33
Considerare la denominazione dei gestori code nello stesso cluster utilizzando una convenzione di denominazione che identifica il cluster a cui appartiene il gestore code. Utilizzare una convenzione di denominazione simile per i nomi canale ed estenderla per descrivere le caratteristiche del canale.

Clustering: considerazioni sulla progettazione della topologia
Questo argomento fornisce una guida per la pianificazione e la gestione dei cluster IBM MQ . Queste informazioni sono una guida basata sul test e sul feedback dei clienti.

Pensando a dove le applicazioni utente e i processi amministrativi interni saranno localizzati in anticipo, molti problemi possono essere evitati o ridotti in un secondo momento. Questo argomento contiene informazioni sulle decisioni di progettazione che possono migliorare le prestazioni e semplificare le attività di manutenzione in base alla scalabilità del cluster.

Prestazioni dell'infrastruttura di clustering

Quando un'applicazione tenta di aprire una coda su un gestore code in un cluster, il gestore code registra il proprio interesse con i repository completi per tale coda in modo che possa scoprire dove si trova la coda nel cluster. Gli aggiornamenti alla posizione o alla configurazione della coda vengono inviati automaticamente dai repository completi al gestore code interessato. Questa registrazione di interesse è nota internamente come sottoscrizione (queste sottoscrizioni non sono le stesse di IBM MQ sottoscrizioni utilizzate per la messaggistica di pubblicazione / sottoscrizione in IBM MQ)

Tutte le informazioni su un cluster passano attraverso ogni repository completo. I repository completi vengono quindi sempre utilizzati in un cluster per il traffico di messaggi di gestione. L'elevato utilizzo delle risorse di sistema durante la gestione di queste sottoscrizioni e la loro trasmissione e i messaggi di configurazione risultanti, possono causare un carico considerevole sull'infrastruttura di cluster. Ci sono un certo numero di cose da considerare quando si assicura che questo carico sia compreso e minimizzato dove possibile:

- Maggiore è il numero di singoli gestori code che utilizzano una coda cluster, maggiore è il numero di sottoscrizioni nel sistema e, di conseguenza, maggiore è il sovraccarico di gestione quando si verificano le modifiche e i sottoscrittori interessati devono essere notificati, in particolare sui gestori code del repository completo. Un modo per ridurre il traffico non necessario e il carico del repository completo consiste nel collegare applicazioni simili (ovvero, quelle che gestiscono le stesse code) a un minor numero di gestori code.
- Oltre al numero di sottoscrizioni nel sistema che influiscono sulle prestazioni, la velocità di modifica nella configurazione degli oggetti cluster può influire sulle prestazioni, ad esempio la modifica frequente di una configurazione della coda cluster.
- Quando un gestore code è un membro di più cluster (ovvero, fa parte di un sistema cluster sovrapposto) qualsiasi interesse in una coda risulta in una sottoscrizione per ogni cluster di cui è membro, anche se gli stessi gestori code sono repository completi per più di uno dei cluster. Questa disposizione aumenta il carico sul sistema ed è uno dei motivi per considerare se sono necessari più cluster sovrapposti, piuttosto che un singolo cluster.
- Il traffico dei messaggi dell'applicazione (ossia i messaggi inviati dalle applicazioni IBM MQ alle code cluster) non passa attraverso i repository completi per raggiungere i gestori code di destinazione. Il traffico di questo messaggio viene inviato direttamente tra il gestore code in cui il messaggio entra nel cluster e il gestore code in cui si trova la coda del cluster. Non è pertanto necessario adattare le elevate frequenze del traffico dei messaggi dell'applicazione rispetto ai gestori code del repository completo, a meno che i gestori code del repository completo non siano uno dei due gestori code menzionati. Per questo motivo, si consiglia di non utilizzare i gestori code del repository completo per il traffico di messaggi dell'applicazione nei cluster in cui il carico dell'infrastruttura di cluster è significativo.

Repository completi

Un repository è una raccolta di informazioni sui gestori code che sono membri di un cluster. Un gestore code che ospita una serie completa di informazioni su ogni gestore code nel cluster ha un repository completo. Per ulteriori informazioni sui repository completi e parziali, consultare Repository del cluster.

I repository completi devono essere conservati su server affidabili e il più possibile disponibili e devono essere evitati i singoli punti di errore. La progettazione cluster deve sempre avere due repository completi. Se si verifica un errore di un repository completo, il cluster può ancora funzionare.

I dettagli di tutti gli aggiornamenti alle risorse del cluster effettuati da un gestore code in un cluster; ad esempio, le code con cluster, vengono inviate da tale gestore code a due repository completi al massimo in tale cluster (o a uno se nel cluster è presente un solo gestore code del repository completo). Tali repository completi contengono le informazioni e le propagano a tutti i gestori code nel cluster che mostrano un interesse per tali informazioni (ovvero, sottoscrivono le informazioni). Per garantire che ciascun membro del cluster abbia una vista aggiornata delle risorse del cluster, ciascun gestore code deve essere in grado di comunicare con almeno un gestore code del repository completo alla volta.

Se, per qualsiasi motivo, un gestore code non riesce a comunicare con repository completi, può continuare a funzionare nel cluster in base al livello di informazioni già memorizzato nella cache per un periodo di tempo, ma non sono disponibili nuovi aggiornamenti o accessi a risorse cluster precedentemente inutilizzate.

Per questo motivo, è necessario mantenere i due repository completi disponibili in ogni momento. Tuttavia, questa disposizione non significa che debbano essere prese misure estreme perché il cluster funziona adeguatamente per un breve periodo senza un repository completo.

Esiste un altro motivo per cui un cluster deve avere due gestori code del repository completi, diversi dalla disponibilità delle informazioni del cluster: questo motivo è per garantire che le informazioni del cluster contenute nella cache del repository completo esistano in due posizioni per scopi di ripristino. Se è presente un solo repository completo e perde le informazioni sul cluster, è necessario l'intervento manuale su tutti i gestori code all'interno del cluster per far funzionare nuovamente il cluster. Se ci sono due repository completi, tuttavia, poiché le informazioni vengono sempre pubblicate e sottoscritte da due repository completi, il repository completo non riuscito può essere recuperato con il minimo sforzo.

- È possibile eseguire la manutenzione su gestori code di repository completi in una progettazione di due cluster di repository completi senza impattare gli utenti di tale cluster: il cluster continua a funzionare con un solo repository, quindi, se possibile, disattivare i repository, applicare la manutenzione ed eseguire nuovamente il backup uno alla volta. Anche se si verifica un'interruzione sul secondo repository completo, le applicazioni in esecuzione rimangono inalterate per un minimo di tre giorni.
- A meno che non vi sia un buon motivo per utilizzare un terzo repository, ad esempio utilizzare un repository completo geograficamente locale per motivi geografici, utilizzare la progettazione di due repository. Avere tre repository completi significa che non si sa mai quali sono i due attualmente in uso e potrebbero esserci problemi di gestione causati dalle interazioni tra più parametri di gestione del carico di lavoro. Non è consigliabile avere più di due repository completi.
- Se hai ancora bisogno di una migliore disponibilità, considera di ospitare i gestori code del repository completo come gestori code a più istanze o di utilizzare il supporto di alta disponibilità specifico della piattaforma per migliorarne la disponibilità.
- È necessario collegare completamente tutti i gestori code del repository completo con canali mittenti del cluster definiti manualmente. È necessario prestare particolare attenzione quando il cluster ha, per qualche ragione giustificabile, più di due repository completi. In questa situazione è spesso possibile perdere uno o più canali e non essere immediatamente evidenti. Quando l'interconnessione completa non si verifica, spesso si verificano problemi difficili da diagnosticare. Sono difficili da diagnosticare perché alcuni repository completi non contengono tutti i dati del repository e, di conseguenza, i gestori code nel cluster hanno viste diverse del cluster a seconda dei repository completi a cui si connettono.

Le applicazioni devono utilizzare code su repository completi?

Un repository completo è nella maggior parte dei modi esattamente come qualsiasi altro gestore code, ed è quindi possibile ospitare le code dell'applicazione sul repository completo e connettere le applicazioni direttamente a questi gestori code. Le applicazioni devono utilizzare code su repository completi?

La risposta comunemente accettata è "No?". Sebbene questa configurazione sia possibile, molti clienti preferiscono mantenere questi gestori code dedicati alla gestione della cache del cluster del repository completo. I punti da considerare quando si decide su una delle due opzioni sono descritti qui, ma in definitiva l'architettura del cluster deve essere appropriata alle particolari esigenze dell'ambiente.

- Aggiornamenti: di solito, per poter utilizzare le nuove funzioni del cluster nelle nuove release di IBM MQ , è necessario aggiornare prima i gestori code del repository completo di tale cluster. Quando un'applicazione nel cluster desidera utilizzare nuove funzioni, potrebbe essere utile essere in grado di aggiornare i repository completi (e alcuni sottoinsiemi di repository parziali) senza eseguire il test di un certo numero di applicazioni co - ubicate.
- Manutenzione: in modo simile se è necessario applicare la manutenzione urgente ai repository completi, è possibile riavviarli o aggiornarli con il comando **REFRESH** senza toccare le applicazioni.
- Prestazioni: man mano che i cluster crescono e le richieste di manutenzione della cache del cluster del repository completo diventano maggiori, mantenere separate le applicazioni riduce il rischio che ciò influisca sulle prestazioni dell'applicazione attraverso il conflitto per le risorse di sistema.
- Requisiti hardware: in genere, i repository completi non devono essere potenti; ad esempio, un server UNIX semplice con una buona aspettativa di disponibilità è sufficiente. In alternativa, per cluster molto grandi o in continuo cambiamento, è necessario considerare le prestazioni del computer repository completo.
- Requisiti software: i requisiti sono di solito il motivo principale per cui si sceglie di ospitare le code dell'applicazione su un repository completo. In un cluster di piccole dimensioni, la collocazione potrebbe significare un requisito per un minor numero di gestori code / server su tutti.

Gestione delle definizioni di canale

Anche all'interno di un singolo cluster, possono esistere più definizioni di canale che forniscono più instradamenti tra due gestori code.

A volte c'è un vantaggio nell'avere canali paralleli all'interno di un singolo cluster, ma questa decisione di progettazione deve essere considerata a fondo; a parte l'aggiunta di complessità, questa progettazione

potrebbe risultare in un sottoutilizzo dei canali che riduce le prestazioni. Questa situazione si verifica perché il test di solito implica l'invio di molti messaggi a una velocità costante, quindi i canali paralleli sono completamente utilizzati. Ma con le condizioni reali di un flusso di messaggi non costante, l'algoritmo di bilanciamento del carico di lavoro causa il calo delle prestazioni quando il flusso di messaggi viene commutato da canale a canale.

Quando un gestore code è un membro di più cluster, esiste l'opzione di utilizzare una singola definizione di canale con un elenco nomi cluster, piuttosto che definire un canale CLUSRCVR separato per ciascun cluster. Tuttavia, questa configurazione può causare difficoltà di amministrazione in un secondo momento; considerare, ad esempio, il caso in cui TLS deve essere applicato a un cluster ma non a un secondo. È pertanto preferibile creare definizioni separate e la convenzione di denominazione suggerita in “Convenzioni di denominazione cluster” a pagina 33 lo supporta.

Bilanciamento del carico di lavoro su più canali

Queste informazioni sono intese come una comprensione avanzata del soggetto. Per la spiegazione di base di questo argomento (che deve essere compresa prima di utilizzare le informazioni qui), consultare Utilizzo dei cluster per la gestione del carico di lavoro, Bilanciamento del carico di lavoro in clustere L'algoritmo di gestione del carico di lavoro del cluster.

L'algoritmo di gestione del carico di lavoro del cluster fornisce una grande serie di strumenti, ma non tutti devono essere utilizzati l'uno con l'altro senza comprendere appieno come funzionano e interagiscono. Potrebbe non essere immediatamente ovvio quanto siano importanti i canali per il processo di bilanciamento del carico di lavoro: l'algoritmo round - robin di gestione del carico di lavoro si comporta come se più canali cluster per un gestore code che possiede una coda cluster, fossero considerati come più istanze di quella coda. Questo processo è spiegato in modo più dettagliato nel seguente esempio:

1. Esistono due gestori code che ospitano una coda in un cluster: QM1 e QM2.
2. Esistono cinque canali riceventi del cluster per QM1.
3. Esiste solo un canale ricevente del cluster per QM2.
4. Quando **MQPUT** o **MQOPEN** on QM3 sceglie un'istanza, l'algoritmo è cinque volte più probabile che invii il messaggio a QM1 che a QM2.
5. La situazione nel passo 4 si verifica perché l'algoritmo visualizza sei opzioni tra cui scegliere (5 + 1) e round-robins tra tutti e cinque i canali in QM1 e il singolo canale in QM2.

Un altro comportamento sottile è che anche quando si inseriscono i messaggi in una coda con cluster che ha un'istanza configurata sul gestore code locale, IBM MQ utilizza lo stato del canale ricevente del cluster locale per decidere se i messaggi devono essere inseriti nell'istanza locale della coda o nelle istanze remote della coda. In questo scenario:

1. Quando si inseriscono i messaggi, l'algoritmo di gestione del carico di lavoro non esamina le singole code cluster, ma i canali cluster che possono raggiungere tali destinazioni.
2. Per raggiungere le destinazioni locali, i canali riceventi locali sono inclusi in questo elenco (anche se non vengono utilizzati per inviare il messaggio).
3. Quando un canale ricevente locale viene arrestato, l'algoritmo di gestione del carico di lavoro preferisce un'istanza alternativa per impostazione predefinita se il relativo CLUSRCVR non viene arrestato. Se esistono più istanze CLUSRCVR locali per la destinazione e almeno una non è arrestata, l'istanza locale rimane idonea.

Clustering: isolamento dell'applicazione utilizzando più code di trasmissione cluster
È possibile isolare i flussi di messaggi tra gestori code in un cluster. È possibile inserire i messaggi trasportati da canali mittenti del cluster differenti in code di trasmissione cluster differenti. È possibile utilizzare l'approccio in un singolo cluster o con cluster sovrapposti. L'argomento fornisce esempi e alcune procedure ottimali per guidare l'utente nella scelta di un approccio da utilizzare.

Quando si distribuisce un'applicazione, è possibile scegliere quali risorse IBM MQ condividere con altre applicazioni e quali non condividere. Esistono diversi tipi di risorse che possono essere condivise, le principali sono il server stesso, il gestore code, i canali e le code. È possibile scegliere di configurare

le applicazioni con meno risorse condivise; allocando code, canali, gestori code o anche server separati a singole applicazioni. In questo caso, la configurazione generale del sistema diventa più grande e complessa. L'utilizzo dei cluster IBM MQ riduce la complessità di gestione di più server, gestori code, code e canali, ma introduce un'altra risorsa condivisa, la coda di trasmissione cluster, SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Figura 12 a pagina 49 è una sezione di una grande distribuzione IBM MQ che illustra il significato della condivisione SYSTEM.CLUSTER.TRANSMIT.QUEUE. Nel diagramma, l'applicazione, Client App, è connessa al gestore code QM2 nel cluster CL1. Un messaggio da Client App viene elaborato dall'applicazione, Server App. Il messaggio viene richiamato da Server App dalla coda cluster Q1 sul gestore code QM3 in CLUSTER2. Poiché le applicazioni client e server non sono nello stesso cluster, il messaggio viene trasferito dal gestore code del gateway QM1.

Il modo normale per configurare un gateway del cluster consiste nel rendere il gestore code del gateway membro di tutti i cluster. Sul gestore code del gateway sono definite code alias cluster per le code cluster in tutti i cluster. Gli alias della coda cluster sono disponibili in tutti i cluster. I messaggi inseriti negli alias della coda cluster vengono instradati tramite il gestore code del gateway alla loro destinazione corretta. Il gestore code del gateway inserisce i messaggi inviati alle code alias del cluster in SYSTEM.CLUSTER.TRANSMIT.QUEUE su QM1comune.

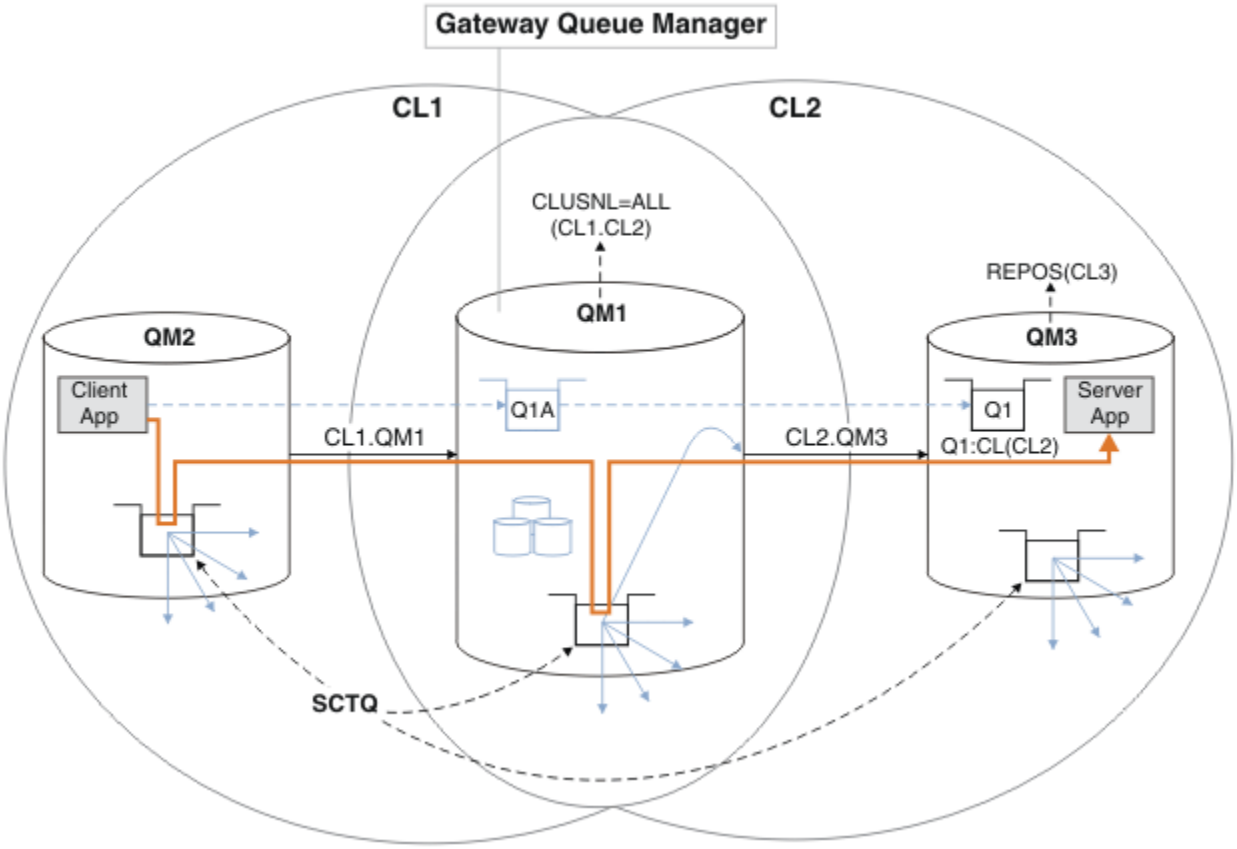
L'architettura hub e spoke richiede tutti i messaggi tra i cluster per passare attraverso il gestore code del gateway. Il risultato è che tutti i messaggi passano attraverso la singola coda di trasmissione del cluster su QM1, SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Dal punto di vista delle prestazioni, una coda singola non è un problema. Una coda di trasmissione comune generalmente non rappresenta un collo di bottiglia delle prestazioni. La velocità di trasmissione dei messaggi sul gateway è in gran parte determinata dalle prestazioni dei canali che si collegano ad esso. La velocità di trasmissione non è generalmente influenzata dal numero di code o dal numero di messaggi sulle code che utilizzano i canali.

Da altre prospettive, l'utilizzo di una singola coda di trasmissione per più applicazioni presenta degli inconvenienti:

- Non è possibile isolare il flusso di messaggi in una destinazione dal flusso di messaggi in un'altra. Non è possibile separare la memoria dei messaggi prima che vengano inoltrati, anche se le destinazioni si trovano in cluster differenti su gestori code differenti.

  Se una destinazione cluster diventa non disponibile, i messaggi per tale destinazione si accumulano nella singola coda di trasmissione e alla fine i messaggi la riempiono. Una volta che la coda di trasmissione è piena, interrompe l'inserimento dei messaggi nella coda di trasmissione per qualsiasi destinazione cluster.
- Non è facile monitorare il trasferimento dei messaggi a diverse destinazioni cluster. Tutti i messaggi sono sulla singola coda di trasmissione. La visualizzazione della profondità della coda di trasmissione fornisce poche indicazioni se i messaggi vengono trasferiti a tutte le destinazioni.



Nota: Le frecce in Figura 12 a pagina 49 e nelle figure seguenti sono di diversi tipi. Le frecce continue rappresentano flussi di messaggi. Le etichette sulle frecce continue sono nomi di canali di messaggi. Le frecce piene grigie sono potenziali flussi di messaggi da SYSTEM.CLUSTER.TRANSMIT.QUEUE sui canali mittente del cluster. Le linee tratteggiate nere collegano le etichette ai loro obiettivi. Le frecce tratteggiate grigie sono riferimenti, ad esempio da una MQOPEN chiamata di `Client` App alla definizione della coda alias del cluster Q1A.

Figura 12. Applicazione client-server distribuita all'architettura hub e spoke utilizzando i cluster IBM MQ

In Figura 12 a pagina 49, i client di `Server` App aprono la coda Q1A. I messaggi vengono collocati in SYSTEM.CLUSTER.TRANSMIT.QUEUE su QM2, trasferiti in SYSTEM.CLUSTER.TRANSMIT.QUEUE su QM1e quindi trasferiti in Q1 su QM3, dove vengono ricevuti dall'applicazione `Server` App .

Il messaggio da `Client` App passa attraverso code di trasmissione del cluster di sistemi su QM2 e QM1. In Figura 12 a pagina 49, l'obiettivo è isolare il flusso di messaggi sul gestore code del gateway dall'applicazione client, in modo che i messaggi non siano memorizzati in SYSTEM.CLUSTER.TRANSMIT.QUEUE. È possibile isolare i flussi su qualsiasi altro gestore code con cluster. È anche possibile isolare i flussi nell'altra direzione, di nuovo al client. Per mantenere brevi le descrizioni delle soluzioni, le descrizioni considerano solo un singolo flusso dall'applicazione client.

Soluzioni per isolare il traffico di messaggi del cluster su un gestore code del gateway cluster

Un modo per risolvere il problema consiste nell'utilizzare gli alias del gestore code o le definizioni di coda remota per collegare i cluster. Creare una definizione di coda remota con cluster, una coda di trasmissione e un canale, per separare ciascun flusso di messaggi sul gestore code del gateway; consultare Aggiunta di una definizione di coda remota per isolare i messaggi inviati da un gestore code del gateway.

A partire da IBM WebSphere MQ 7.5 , i gestori code cluster non sono limitati a una singola coda di trasmissione cluster. Sono disponibili due opzioni:

1. Definire manualmente ulteriori code di trasmissione cluster e definire quali canali mittenti del cluster trasferiscono i messaggi da ciascuna coda di trasmissione; consultare Aggiunta di una coda di trasmissione cluster per isolare il traffico dei messaggi cluster inviati da un gestore code gateway.
2. Consente al gestore code di creare e gestire automaticamente ulteriori code di trasmissione cluster. Definisce una diversa coda di trasmissione cluster per ogni canale mittente del cluster; consultare Modifica del valore predefinito per separare le code di trasmissione cluster per isolare il traffico messaggi.

È possibile combinare manualmente le code di trasmissione del cluster definite per alcuni canali mittente del cluster con il gestore code che gestisce il resto. La combinazione di code di trasmissione è l'approccio utilizzato in Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code del gateway. In questa soluzione, la maggior parte dei messaggi tra i cluster utilizza il comune `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Un'applicazione è critica e tutti i suoi flussi di messaggi sono isolati da altri flussi utilizzando una coda di trasmissione cluster definita manualmente.

La configurazione in Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway è limitata. Non separa il traffico di messaggi diretto a una coda cluster sullo stesso gestore code nello stesso cluster di un'altra coda cluster. È possibile separare il traffico di messaggi in code individuali utilizzando le definizioni di code remote che fanno parte dell'accodamento distribuito. Con i cluster, utilizzando più code di trasmissione cluster, è possibile separare il traffico dei messaggi che va a canali mittenti del cluster differenti. Più code del cluster nello stesso cluster, sullo stesso gestore code, condividono un canale mittente del cluster. I messaggi per tali code vengono memorizzati nella stessa coda di trasmissione, prima di essere inoltrati dal gestore code gateway. Nella configurazione in Aggiunta di un cluster e di una coda di trasmissione del cluster per isolare il traffico di messaggi del cluster inviato da un gestore code del gateway, la limitazione viene annullata aggiungendo un altro cluster e rendendo il gestore code e la coda del cluster membri del nuovo cluster. Il nuovo gestore code potrebbe essere l'unico gestore code nel cluster. È possibile aggiungere più gestori code al cluster e utilizzare lo stesso cluster per isolare anche le code cluster su tali gestori code.

Concetti correlati

“Controllo accessi e code di trasmissione di più cluster” a pagina 28
Scegliere tra tre modalità di controllo quando un'applicazione inserisce i messaggi nelle code cluster remote. Le modalità sono la verifica in remoto rispetto alla coda del cluster, la verifica in locale rispetto a `SYSTEM.CLUSTER.TRANSMIT.QUEUE`o la verifica rispetto ai profili locali per la coda del cluster o il gestore code del cluster.

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

“Cluster sovrapposti” a pagina 35
I cluster sovrapposti forniscono ulteriori funzioni di gestione. Utilizzare gli elenchi nomi per ridurre il numero di comandi necessari per gestire i cluster che si sovrappongono.

Attività correlate

Autorizzazione all'inserimento di messaggi nelle code del cluster remoto

Aggiunta di una definizione di coda remota per isolare i messaggi inviati da un gestore code del gateway

Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway

Aggiunta di un cluster e di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway

Modifica del valore predefinito per separare le code di trasmissione del cluster per isolare il traffico dei messaggi

Creazione di cluster a due sovrapposizioni con un gestore code del gateway

Configurazione dei percorsi dei messaggi tra cluster

Protezione

Riferimenti correlati

setmqaut

Clustering: pianificazione della configurazione delle code di trasmissione del cluster
L'utente viene guidato nelle scelte delle code di trasmissione cluster. È possibile configurare una coda predefinita comune, code predefinite separate o code definite manualmente.

Prima di iniziare

Consultare “Come scegliere quale tipo di coda di trasmissione del cluster utilizzare” a pagina 54.

Informazioni su questa attività

Sono disponibili alcune scelte da effettuare quando si pianifica come configurare un gestore code per selezionare una coda di trasmissione cluster.

1. Qual è la coda di trasmissione cluster predefinita per i trasferimenti di messaggi cluster?
   a. Una coda di trasmissione cluster comune, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.
   b. Separare le code di trasmissione cluster. Il gestore code gestisce le code di trasmissione cluster separate. Le crea come code dinamiche permanenti dalla coda modello, `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE`. Crea una coda di trasmissione cluster per ogni canale mittente del cluster che utilizza.
2. Per le code di trasmissione del cluster che si decide di creare manualmente, sono disponibili altre due opzioni:
   a. Definire una coda di trasmissione separata per ogni canale mittente del cluster che si decide di configurare manualmente. In questo caso, impostare l'attributo della coda **CLCHNAME** della coda di trasmissione sul nome di un canale mittente del cluster. Selezionare il canale mittente del cluster che deve trasferire i messaggi da questa coda di trasmissione.
   b. Combinare il traffico dei messaggi per un gruppo di canali mittente del cluster sulla stessa coda di trasmissione del cluster; consultare Figura 13 a pagina 52. In questo caso, impostare l'attributo della coda **CLCHNAME** di ogni coda di trasmissione comune su un nome canale mittente del cluster generico. Un nome canale mittente del cluster generico è un filtro per raggruppare i nomi canale mittente del cluster. Ad esempio, `SALES.*` raggruppa tutti i canali mittente del cluster che hanno nomi che iniziano con `SALES.`. È possibile inserire più caratteri jolly ovunque nella stringa filtro. Il carattere jolly è un asterisco, "`*`". Rappresenta da zero a qualsiasi numero di caratteri.

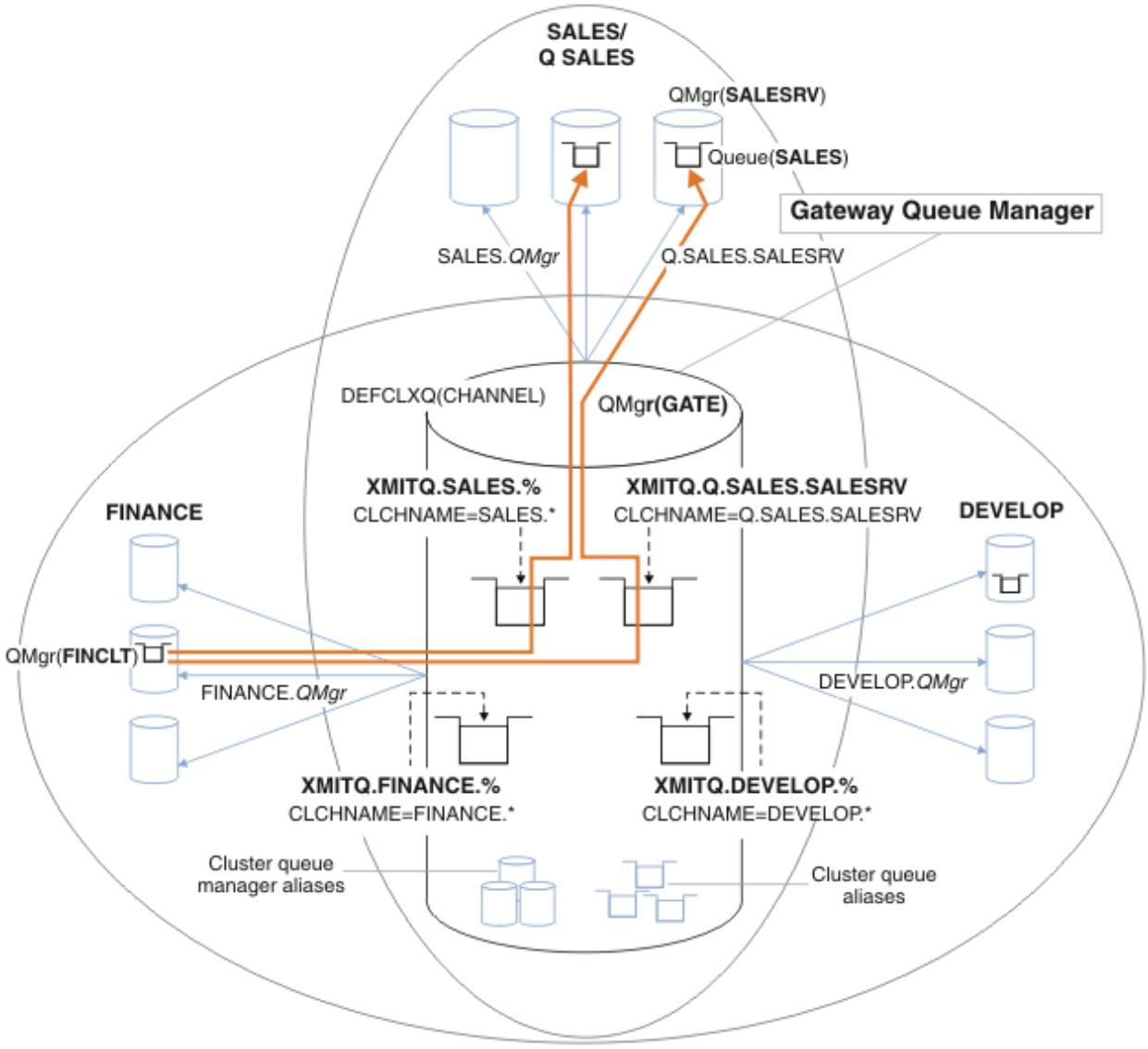


Figura 13. Esempio di code di trasmissione specifiche per cluster IBM MQ dipartimentali differenti

Procedura

1. Selezionare il tipo di coda di trasmissione cluster predefinita da utilizzare.
   - Scegliere una singola coda di trasmissione cluster o separare le code per ogni connessione cluster.

   Lasciare l'impostazione predefinita o eseguire il comando **MQSC** :

   ```
   ALTER QMGR DEFCLXQ(CHANNEL)
   ```

2. Isolare i flussi di messaggi che non devono condividere una coda di trasmissione cluster con altri flussi.
   - Consultare “Clustering: configurazione di esempio di più code di trasmissione cluster” a pagina 55. Nell'esempio, la coda SALES , che deve essere isolata, è un membro del cluster SALES , su SALESRV. Per isolare la coda SALES , creare un nuovo cluster `Q.SALES`, rendere membro il gestore code SALESRV e modificare la coda SALES in modo che appartenga a `Q.SALES`.
   - I gestori code che inviano messaggi a SALES devono essere anche membri del nuovo cluster. Se si utilizza un alias della coda del cluster e un gestore code del gateway, come nell'esempio, in molti

casi è possibile limitare le modifiche per rendere il gestore code del gateway membro del nuovo cluster.

- Tuttavia, la separazione dei flussi dal gateway alla destinazione non separa i flussi al gateway dal gestore code di origine. Ma a volte risulta essere sufficiente per separare i flussi dal gateway e non i flussi verso il gateway. Se non è sufficiente, aggiungere il gestore code di origine nel nuovo cluster. Se si desidera che i messaggi passino attraverso il gateway, spostare l'alias del cluster sul nuovo cluster e continuare a inviare i messaggi all'alias del cluster sul gateway e non direttamente al gestore code di destinazione.

Seguire questa procedura per isolare i flussi di messaggi:

a) Configurare le destinazioni dei flussi in modo tale che ciascuna coda di destinazione sia l'unica coda in un cluster specifico su tale gestore code.

b) Creare i canali mittente e ricevente del cluster per tutti i nuovi cluster creati in base a una convenzione di denominazione sistematica.

- Consultare “Clustering: considerazioni speciali per i cluster che si sovrappongono” a pagina 43.

c) Definire una coda di trasmissione cluster per ogni destinazione isolata su ogni gestore code che invia messaggi alla coda di destinazione.

- Una convenzione di denominazione per le code di trasmissione del cluster consiste nell'utilizzare il valore dell'attributo del nome del canale cluster, `CLCHNAME`, con prefisso `XMITQ.`

3. Creare code di trasmissione del cluster per soddisfare i requisiti di governance o di controllo.

- I tipici requisiti di governance e di monitoraggio risultano in una coda di trasmissione per cluster o in una coda di trasmissione per gestore code. Se si segue la convenzione di denominazione per i canali cluster, *`ClusterName.  QueueManagerName`*, è semplice creare nomi di canali generici che selezionino un cluster di gestori code o tutti i cluster di cui è membro un gestore code; consultare “Clustering: configurazione di esempio di più code di trasmissione cluster” a pagina 55.
- Estendere la convenzione di denominazione per le code di trasmissione del cluster per soddisfare i nomi di canale generici, sostituendo il simbolo asterisco con un segno di percentuale. Ad esempio:

```
DEFINE QLOCAL(XMITQ.SALES.%) USAGE(XMITQ) CLCHNAME(SALES.*)
```

Concetti correlati

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

“Controllo accessi e code di trasmissione di più cluster” a pagina 28
Scegliere tra tre modalità di controllo quando un'applicazione inserisce i messaggi nelle code cluster remote. Le modalità sono la verifica in remoto rispetto alla coda del cluster, la verifica in locale rispetto a `SYSTEM.CLUSTER.TRANSMIT.QUEUE`o la verifica rispetto ai profili locali per la coda del cluster o il gestore code del cluster.

“Cluster sovrapposti” a pagina 35
I cluster sovrapposti forniscono ulteriori funzioni di gestione. Utilizzare gli elenchi nomi per ridurre il numero di comandi necessari per gestire i cluster che si sovrappongono.

Attività correlate

Aggiunta di una definizione di coda remota per isolare i messaggi inviati da un gestore code del gateway

Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway

Aggiunta di un cluster e di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway

Modifica del valore predefinito per separare le code di trasmissione del cluster per isolare il traffico dei messaggi

Creazione di cluster a due sovrapposizioni con un gestore code del gateway

Configurazione dei percorsi dei messaggi tra cluster

Come scegliere quale tipo di coda di trasmissione del cluster utilizzare
Come scegliere tra diverse opzioni di configurazione della coda di trasmissione del cluster.

È possibile scegliere quale coda di trasmissione cluster è associata a un canale mittente del cluster.

1. È possibile avere tutti i canali mittenti del cluster associati alla singola coda di trasmissione del cluster predefinita, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`; questa è l'opzione predefinita.
2. È possibile impostare tutti i canali mittenti del cluster in modo che vengano associati automaticamente a una coda di trasmissione cluster separata. Le code vengono create dal gestore code dalla coda modello `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` e denominate `SYSTEM.CLUSTER.TRANSMIT.`*`ChannelName`*. I canali utilizzeranno la relativa coda di trasmissione cluster con nome univoco se l'attributo del gestore code **DEFCLXQ** è impostato su `CHANNEL`.
3. È possibile impostare canali mittenti del cluster specifici che devono essere serviti da una singola coda di trasmissione del cluster. Selezionare questa opzione creando una coda di trasmissione e impostandone il relativo attributo **CLCHNAME** sul nome del canale mittente del cluster.
4. È possibile selezionare gruppi di canali mittente del cluster che devono essere serviti da una singola coda di trasmissione del cluster. Selezionare questa opzione creando una coda di trasmissione e impostando l'attributo **CLCHNAME** su un nome canale generico, come *`ClusterName`*`.*`. Se si denominano i canali del cluster seguendo le convenzioni di denominazione in "Clustering: considerazioni speciali per i cluster che si sovrappongono" a pagina 43, questo nome seleziona tutti i canali del cluster connessi ai gestori code nel cluster *`ClusterName`*.

È possibile combinare una delle opzioni predefinite della coda di trasmissione del cluster per alcuni canali mittente del cluster con un numero qualsiasi di configurazioni specifiche e generiche della coda di trasmissione del cluster.

Best practice

Nella maggior parte dei casi, per le installazioni esistenti di IBM MQ , la configurazione predefinita è la scelta migliore. Un gestore code del cluster memorizza i messaggi cluster su una singola coda di trasmissione cluster, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. È possibile modificare il valore predefinito per memorizzare i messaggi per diversi gestori code e cluster su code di trasmissione separate oppure definire le proprie code di trasmissione.

Nella maggior parte dei casi, per le nuove installazioni di IBM MQ , la configurazione predefinita è anche la scelta migliore. Il processo di passaggio dalla configurazione predefinita al valore predefinito alternativo di avere una coda di trasmissione per ogni canale mittente del cluster è automatico. Anche il ritorno è automatico. La scelta dell'uno o dell'altro non è critica, si può invertire.

Il motivo per scegliere una configurazione diversa è più relativo alla governance e alla gestione che alla funzionalità o alle prestazioni. Con un paio di eccezioni, la configurazione di più code di trasmissione cluster non avvantaggia il comportamento del gestore code. Risulta in un numero maggiore di code e richiede la modifica delle procedure di monitoraggio e gestione già impostate che fanno riferimento alla singola coda di trasmissione. Questo è il motivo per cui, a conti fatti, rimanere con la configurazione predefinita è la scelta migliore, a meno che non si abbiano forti ragioni di governance o di gestione per una scelta diversa.

Le eccezioni sono entrambe relative a ciò che accade se il numero di messaggi memorizzati su `SYSTEM.CLUSTER.TRANSMIT.QUEUE` aumenta. Se si esegue ogni passo per separare i messaggi per una destinazione dai messaggi per un'altra destinazione, i problemi di canale e di consegna con una destinazione non dovrebbero influire sulla consegna ad un'altra destinazione. Tuttavia, il numero di messaggi memorizzati su `SYSTEM.CLUSTER.TRANSMIT.QUEUE` può aumentare a causa della mancata consegna dei messaggi abbastanza rapida a una destinazione. Il numero di messaggi su `SYSTEM.CLUSTER.TRANSMIT.QUEUE` per una destinazione può influire sulla consegna dei messaggi ad altre destinazioni.

Per evitare problemi derivanti dal riempimento di una singola coda di trasmissione, è necessario creare una capacità sufficiente nella configurazione. Quindi, se una destinazione non riesce e un backlog di messaggi inizia a crescere, hai tempo per risolvere il problema.

Se i messaggi vengono instradati tramite un gestore code hub, come un gateway cluster, condividono una coda di trasmissione comune, SYSTEM.CLUSTER.TRANSMIT.QUEUE. Se il numero di messaggi memorizzati su SYSTEM.CLUSTER.TRANSMIT.QUEUE sul gestore code del gateway raggiunge la profondità massima, il gestore code inizia a rifiutare i nuovi messaggi per la coda di trasmissione fino a quando la profondità non si riduce. La congestione influisce sui messaggi per tutte le destinazioni instradati attraverso il gateway. I messaggi eseguono il backup delle code di trasmissione di altri gestori code che inviano messaggi al gateway. Il problema si manifesta nei messaggi scritti nei log degli errori del gestore code, con un calo della velocità di trasmissione dei messaggi e tempi più lunghi tra l'invio di un messaggio e il momento in cui un messaggio arriva a destinazione.

L'effetto della congestione su una singola coda di trasmissione può diventare evidente, anche prima che sia piena. Se si dispone di un traffico di messaggi misto, con alcuni messaggi non persistenti di grandi dimensioni e alcuni messaggi di piccole dimensioni, il tempo di consegna dei messaggi di piccole dimensioni aumenta man mano che la coda di trasmissione si riempie. Il ritardo è dovuto alla scrittura di messaggi non persistenti di grandi dimensioni su disco che normalmente non vengono scritti su disco. Se si dispone di flussi di messaggi critici per il tempo, che condividono una coda di trasmissione cluster con altri flussi di messaggi misti, potrebbe essere utile configurare un percorso di messaggi speciale per isolarlo da altri flussi di messaggi; consultare Aggiunta di un cluster e di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviato da un gestore code gateway.

Gli altri motivi per configurare code di trasmissione cluster separate sono per soddisfare i requisiti di governance o per semplificare i messaggi di monitoraggio inviati a destinazioni cluster differenti. Ad esempio, potrebbe essere necessario dimostrare che i messaggi per una destinazione non condividono mai una coda di trasmissione con i messaggi per un'altra destinazione.

Modificare l'attributo del gestore code **DEFCLXQ** che controlla la coda di trasmissione cluster predefinita, per creare code di trasmissione cluster differenti per ogni canale mittente del cluster. Più destinazioni possono condividere un canale mittente del cluster, quindi devi pianificare i tuoi cluster per soddisfare pienamente questo obiettivo. Applicare sistematicamente il metodo Aggiunta di un cluster e di una coda di trasmissione del cluster per isolare il traffico di messaggi del cluster inviati da un gestore code del gateway a tutte le code del cluster. Il risultato che si desidera ottenere è che nessuna destinazione cluster condivida un canale mittente del cluster con un'altra destinazione cluster. Di conseguenza, nessun messaggio per una destinazione cluster condivide la propria coda di trasmissione cluster con un messaggio per un'altra destinazione.

La creazione di una coda di trasmissione cluster separata per un flusso di messaggi specifico rende più semplice il monitoraggio del flusso di messaggi verso tale destinazione. Per utilizzare una nuova coda di trasmissione cluster, definire la coda, associarla a un canale mittente del cluster e arrestare e avviare il canale. La modifica non deve essere permanente. È possibile isolare un flusso di messaggi per un certo periodo di tempo, per monitorare la coda di trasmissione e tornare quindi a utilizzare nuovamente la coda di trasmissione predefinita.

Attività correlate

Clustering: configurazione di esempio di più code di trasmissione cluster
In questa attività si applicano le operazioni per pianificare più code di trasmissione del cluster a tre cluster sovrapposti. I requisiti sono di separare i flussi di messaggi in una coda cluster, da tutti gli altri flussi di messaggi e di memorizzare i messaggi per cluster differenti su code di trasmissione cluster differenti.

Clustering: commutazione delle code di trasmissione del cluster
Pianificare come rendere effettive le modifiche alle code di trasmissione del cluster di un gestore code di produzione esistente.

Clustering: configurazione di esempio di più code di trasmissione cluster
In questa attività si applicano le operazioni per pianificare più code di trasmissione del cluster a tre cluster sovrapposti. I requisiti sono di separare i flussi di messaggi in una coda cluster, da tutti gli altri flussi di messaggi e di memorizzare i messaggi per cluster differenti su code di trasmissione cluster differenti.

Informazioni su questa attività

I passi in questa attività mostrano come applicare la procedura in “Clustering: pianificazione della configurazione delle code di trasmissione del cluster” a pagina 51 e arrivare alla configurazione mostrata in Figura 14 a pagina 56. È un esempio di tre cluster sovrapposti, con un gestore code del gateway, configurato con code di trasmissione del cluster separate. I comandi MQSC per definire i cluster sono descritti in “Creazione dei cluster di esempio” a pagina 58.

Ad esempio, ci sono due requisiti. Uno è separare il flusso di messaggi dal gestore code del gateway all'applicazione di vendita che registra le vendite. Il secondo è quello di interrogare quanti messaggi sono in attesa di essere inviati a diverse aree dipartimentali in qualsiasi momento. I cluster `SALES`, `FINANCE`e `DEVELOP` sono già definiti. I messaggi cluster vengono attualmente inoltrati da `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

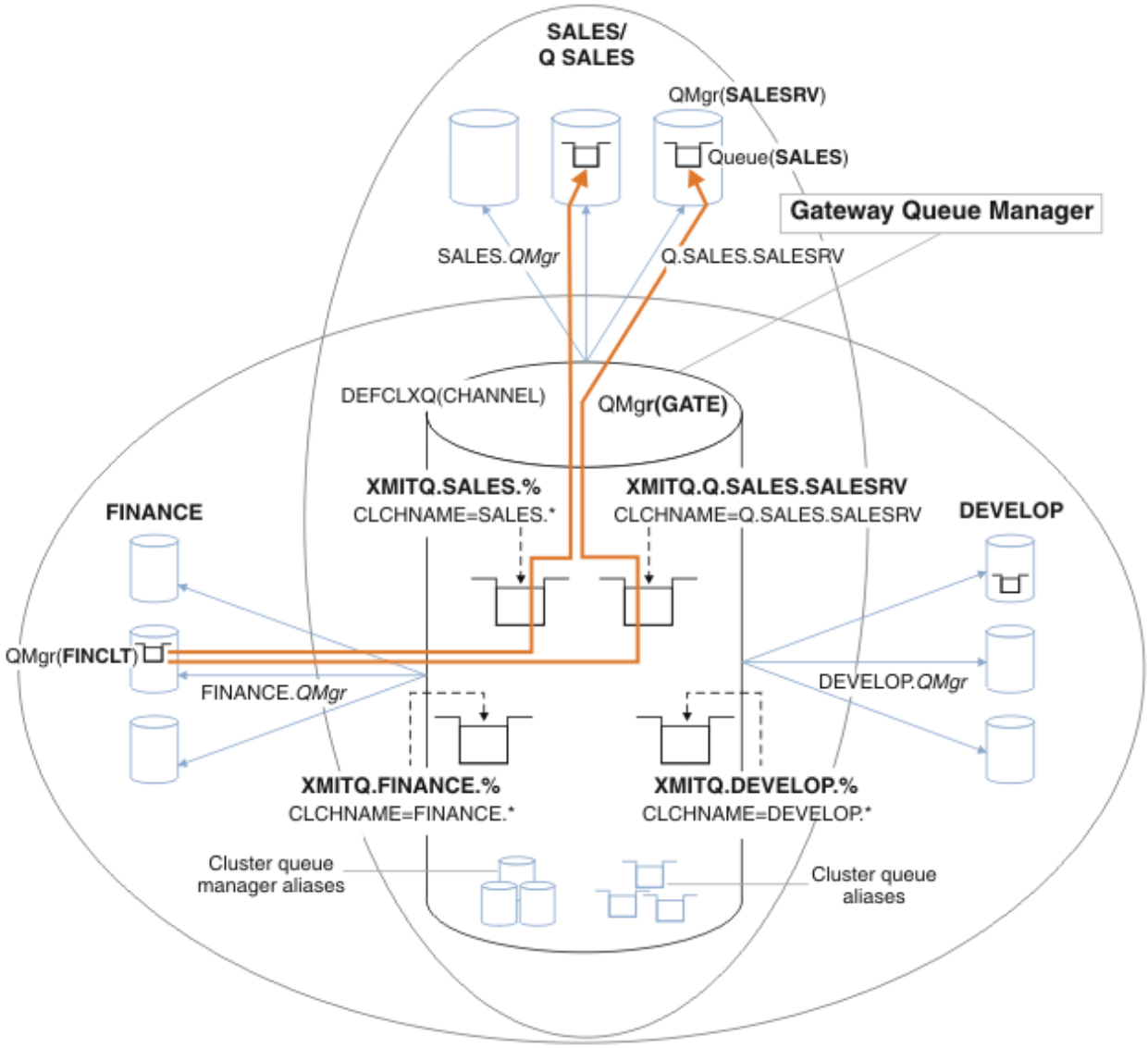


Figura 14. Esempio di code di trasmissione specifiche per cluster IBM MQ dipartimentali differenti

I passi per modificare i cluster sono i seguenti. Per le definizioni, vedi Modifiche per isolare la coda di vendite in un nuovo cluster e separare le code di trasmissione cluster gateway.

Procedura

1. Il primo passo di configurazione è " Selezionare il tipo di coda di trasmissione cluster predefinita da utilizzare ".

La decisione è di creare code di trasmissione del cluster predefinite separate eseguendo il seguente comando **MQSC** sul gestore code GATE .

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

Non esiste un motivo valido per scegliere questo valore predefinito, poiché l'intento è definire manualmente le code di trasmissione cluster. La scelta ha un valore diagnostico debole. Se una definizione manuale viene eseguita in modo non corretto e un messaggio scorre in una coda di trasmissione del cluster predefinita, viene visualizzato nella creazione di una coda di trasmissione del cluster dinamica permanente.

2. Il secondo passo di configurazione è " Isolare i flussi di messaggi che non devono condividere una coda di trasmissione cluster con altri flussi ".

   In questo caso, l'applicazione di vendita che riceve i messaggi dalla coda SALES su SALESRV richiede isolamento. È richiesto solo l'isolamento dei messaggi dal gestore code del gateway. Le tre fasi secondarie sono:

   a) " Configurare le destinazioni dei flussi in modo tale che ciascuna coda di destinazione sia l'unica coda in un cluster specifico su tale gestore code ".

      L'esempio richiede l'aggiunta di un gestore code SALESRV a un nuovo cluster all'interno del reparto vendite. Se si hanno poche code che richiedono l'isolamento, è possibile decidere di creare un cluster specifico per la coda SALES . Una convenzione di denominazione possibile per il nome del cluster consiste nel denominare tali cluster, `Q.` *QueueName*, ad esempio `Q.SALES`. Un approccio alternativo, che potrebbe essere più pratico se si dispone di un numero elevato di code da isolare, consiste nel creare cluster di code isolate dove e quando necessario. I nomi dei cluster potrebbero essere `QUEUES.` *n*.

      Nell'esempio, il nuovo cluster è denominato `Q.SALES`. Per aggiungere il nuovo cluster, vedere le definizioni in Modifiche per isolare la coda di vendita in un nuovo cluster e separare le code di trasmissione cluster gateway. Il riepilogo delle modifiche di definizione è il seguente:

      i) Aggiungere `Q.SALES` all'elenco nomi dei cluster sui gestori code del repository. Si fa riferimento all'elenco nomi nel parametro **REPOSNL** del gestore code.

      ii) Aggiungere `Q.SALES` all'elenco dei nomi dei cluster sul gestore code gateway. Si fa riferimento all'elenco nomi in tutte le definizioni alias della coda cluster e alias del gestore code cluster sul gestore code del gateway.

      iii) Creare un elenco nomi sul gestore code SALESRV, per entrambi i cluster di cui è membro e modificare l'appartenenza del cluster della coda SALES :

      ```
      DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES) REPLACE
      ALTER QLOCAL(SALES) CLUSTER(' ') CLUSNL(SALESRV.CLUSTERS)
      ```

      La coda SALES è un membro di entrambi i cluster, solo per la transizione. Una volta eseguita la nuova configurazione, si rimuove la coda SALES dal cluster SALES ; consultare Figura 15 a pagina 61.

   b) " Creare i canali mittente e ricevente del cluster per tutti i nuovi cluster creati in base a una convenzione di denominazione sistematica ".

      i) Aggiungere il canale ricevente del cluster `Q.SALES.` *RepositoryQMgr* a ciascuno dei gestori code del repository

      ii) Aggiungere il canale mittente del cluster `Q.SALES.` *OtherRepositoryQMgr* a ciascuno dei gestori code del repository, per connettersi all'altro gestore repository. Avviare questi canali.

      iii) Aggiungere i canali riceventi del cluster `Q.SALES.SALESRV`e `Q.SALES.GATE` a uno dei gestori code del repository in esecuzione.

      iv) Aggiungere i canali mittente del cluster `Q.SALES.SALESRV`e `Q.SALES.GATE` ai gestori code SALESRV e GATE . Connetti il canale mittente del cluster al gestore code del repository su cui hai creato i canali riceventi del cluster.

c) " Definire una coda di trasmissione cluster per ogni destinazione isolata su ogni gestore code che invia messaggi alla coda di destinazione ".

Sul gestore code del gateway definire la coda di trasmissione del cluster XMITQ.Q.SALES.SALESRV per il canale mittente del cluster Q.SALES.SALESRV :

```
DEFINE QLOCAL(XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
```

3. Il terzo passo di configurazione è " Creare code di trasmissione del cluster per soddisfare i requisiti di governance o di controllo ".

   Sul gestore code gateway definire le code di trasmissione del cluster:

```
DEFINE QLOCAL(XMITQ.SALES)  USAGE(XMITQ) CLCHNAME(SALES.*)  REPLACE
DEFINE QLOCAL(XMITQ.DEVELOP) USAGE(XMITQ) CLCHNAME(DEVELOP.*) REPLACE
DEFINE QLOCAL(XMITQ.FINANCE) USAGE(XMITQ) CLCHNAME(SALES.*)  REPLACE
```

Operazioni successive

Passare alla nuova configurazione sul gestore code gateway.

Lo switch viene attivato avviando i nuovi canali e riavviando i canali che ora sono associati a code di trasmissione differenti. In alternativa, è possibile arrestare e avviare il gestore code gateway.

1. Arrestare i canali seguenti sul gestore code gateway:

```
SALES. Qmgr
DEVELOP. Qmgr
FINANCE. Qmgr
```

2. Avviare i canali seguenti sul gestore code del gateway:

```
SALES. Qmgr
DEVELOP. Qmgr
FINANCE. Qmgr
Q.SALES.SAVESRV
```

Una volta completata la commutazione, rimuovere la coda SALES dal cluster SALES ; consultare Figura 15 a pagina 61.

Concetti correlati

Come scegliere quale tipo di coda di trasmissione del cluster utilizzare
Come scegliere tra diverse opzioni di configurazione della coda di trasmissione del cluster.

Attività correlate

Clustering: commutazione delle code di trasmissione del cluster
Pianificare come rendere effettive le modifiche alle code di trasmissione del cluster di un gestore code di produzione esistente.

Creazione dei cluster di esempio
Le definizioni e istruzioni per creare il cluster di esempio e modificarlo per isolare la coda SALES e separare i messaggi sul gestore code del gateway.

Informazioni su questa attività

I comandi **MQSC** completi per creare i cluster FINANCE, SALESe Q.SALES sono forniti in Definizioni per i cluster di base, Modifiche per isolare la coda di vendita in un nuovo cluster e separare le code di trasmissione del cluster gatewaye Rimuovi la coda di vendita sul gestore code SALESRV dal cluster di vendita. Il cluster DEVELOP viene omesso dalle definizioni, per mantenerle più brevi.

Procedura

1. Creare i cluster SALES e FINANCE e il gestore code del gateway.
   a) Creare i gestori code.

      Eseguire il comando: `crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE` *QmgrName* per ciascuno dei nomi gestore code in Tabella 4 a pagina 59.

Tabella 4. Nomi di gestori code e numeri di porta		
Descrizione	**Nome del gestore code**	**Numero di porta**
Repository finanziario	FINR1	1414
Repository finanziario	FINR2	1415
Cliente finanziario	FINCLT	1418
Archivio vendite	SALER1	1416
Archivio vendite	SALER2	1417
Server di vendita	SALESRV	1419
Gateway	GATE	1420

   b) Avvia tutti i gestori code

      Eseguire il comando: `strmqm` *QmgrName* per ciascuno dei nomi gestore code in Tabella 4 a pagina 59.

   c) Creare le definizioni per ciascuno dei gestori code

      Eseguire il seguente comando: `runmqsc` *QmgrName* < *filename* dove i file sono elencati in Definizioni per i cluster di basee il nome file corrisponde al nome del gestore code.

      Definizioni per i cluster di base
      finr1.txt

```
DEFINE LISTENER(1414) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1414) REPLACE
START LISTENER(1414)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
```

      finr2.txt

```
DEFINE LISTENER(1415) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1415) REPLACE
START LISTENER(1415)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
```

      finclt.txt

```
DEFINE LISTENER(1418) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1418) REPLACE
START LISTENER(1418)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR)  CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINCLT) CHLTYPE(CLUSRCVR) CONNAME('localhost(1418)')
CLUSTER(FINANCE) REPLACE
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

saler1.txt

```
DEFINE LISTENER(1416) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1416) REPLACE
START LISTENER(1416)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
```

saler2.txt

```
DEFINE LISTENER(1417) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1417) REPLACE
START LISTENER(1417)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
```

salesrv.txt

```
DEFINE LISTENER(1419) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1419) REPLACE
START LISTENER(1419)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(SALES) REPLACE
DEFINE QLOCAL(SALES) CLUSTER(SALES) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

gate.txt

```
DEFINE LISTENER(1420) TRPTYPE(TCP) IPADDR(LOCALHOST) CONTROL(QMGR) PORT(1420) REPLACE
START LISTENER(1420)
DEFINE NAMELIST(ALL) NAMES(SALES, FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.GATE)  CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(SALES) REPLACE
DEFINE QALIAS(A.SALES) CLUSNL(ALL) TARGET(SALES) TARGTYPE(QUEUE) DEFBIND(NOTFIXED)
REPLACE
DEFINE QREMOTE(FINCLT) RNAME(' ') RQMNAME(FINCLT) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(SALESRV) RNAME(' ') RQMNAME(SALESRV) CLUSNL(ALL) REPLACE
```

2. Verificare la configurazione eseguendo il programma di richiesta di esempio.
   a) Avviare il programma di controllo dei trigger sul gestore code SALESRV

      Su Windows, aprire una finestra di comandi ed eseguire il comando `runmqtrm -m SALESRV`
   b) Eseguire il programma di richiesta di esempio e inviare una richiesta.

      Su Windows, aprire una finestra di comandi ed eseguire il comando `amqsreq A.SALES FINCLT`

      Il messaggio di richiesta viene ripetuto e dopo 15 secondi il programma di esempio termina.
3. Creare le definizioni per isolare la coda SALES nel cluster `Q.SALES` e separare i messaggi cluster per il cluster SALES e FINANCE nel gestore code gateway.

   Eseguire il comando: `runmqsc` *QmgrName* < *filename* dove i file sono elencati nel seguente elenco e il nome file quasi corrisponde al nome del gestore code.

Modifiche per isolare la coda di vendita in un nuovo cluster e separare le code di trasmissione del cluster gateway
chgsaler1.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
```

chgsaler2.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
```

chgsalesrv.txt

```
DEFINE NAMELIST (CLUSTERS) NAMES(SALES, Q.SALES)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SAVESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(Q.SALES) REPLACE
ALTER QLOCAL (SALES) CLUSTER(' ') CLUSNL(CLUSTERS)
```

chggate.txt

```
ALTER NAMELIST(ALL) NAMES(SALES, FINANCE, Q.SALES)
ALTER QMGR DEFCLXQ(CHANNEL)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR)  CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.GATE)  CHLTYPE(CLUSRCVR) CONNAME('localhost(1420)')
CLUSTER(Q.SALES) REPLACE
DEFINE QLOCAL (XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
DEFINE QLOCAL (XMITQ.SALES)     USAGE(XMITQ) CLCHNAME(SALES.*)  REPLACE
DEFINE QLOCAL (XMITQ.FINANCE)    USAGE(XMITQ) CLCHNAME(FINANCE.*) REPLACE
```

4. Rimuovere la coda SALES dal cluster SALES .

   Eseguire il comando **MQSC** in Figura 15 a pagina 61:

   ```
   ALTER QLOCAL(SALES) CLUSTER('Q.SALES') CLUSNL(' ')
   ```

   Figura 15. Rimuovere la coda delle vendite sul gestore code SALESRV dal cluster delle vendite
5. Passare i canali alle nuove code di trasmissione.

   Il requisito è quello di arrestare e avviare tutti i canali utilizzati dal gestore code GATE . Per eseguire questa operazione con un numero minimo di comandi, arrestare e avviare il gestore code

   ```
   endmqm -i GATE
   strmqm GATE
   ```

Operazioni successive

1. Eseguire di nuovo il programma di richiesta di esempio per verificare il funzionamento della nuova configurazione; consultare il passo “2” a pagina 60
2. Monitorare i messaggi che passano attraverso tutte le code di trasmissione del cluster sul gestore code GATE :

a. Modificare la definizione di ciascuna delle code di trasmissione del cluster per attivare il controllo della coda.

```
ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.
name) STATQ(ON)
```

b. Controllare che il monitoraggio delle statistiche del gestore code sia OFF, per ridurre al minimo l'output e impostare l'intervallo di controllo su un valore inferiore per eseguire comodamente più verifiche.

```
ALTER QMGR STATINT(60) STATCHL(OFF) STATQ(OFF) STATMQI(OFF) STATACLS(OFF)
```

c. Riavviare il gestore code GATE .

d. Eseguire il programma di richiesta di esempio alcune volte per verificare che un numero uguale di messaggi stia passando attraverso SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV e SYSTEM.CLUSTER.TRANSMIT.QUEUE. Le richieste passano attraverso SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV e le risposte attraverso SYSTEM.CLUSTER.TRANSMIT.QUEUE.

```
amqsmon -m GATE -t statistics
```

e. I risultati su un paio di intervalli sono i seguenti:

```
C:\Documents and Settings\Admin>amqsmon -m GATE -t statistics
MonitoringType: QueueStatistics
QueueManager: 'GATE'
IntervalStartDate: '2012-02-27'
IntervalStartTime: '14.59.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.00.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
CreateDate: '2012-02-24'
CreateTime: '15.58.15'
...
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [435, 0]
GetCount: [1, 0]
GetBytes: [435, 0]
...
QueueStatistics: 1
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
CreateDate: '2012-02-24'
CreateTime: '16.37.43'
...
PutCount: [1, 0]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [435, 0]
GetCount: [1, 0]
GetBytes: [435, 0]
```

```
...
MonitoringType: QueueStatistics
QueueManager: 'GATE'
IntervalStartDate: '2012-02-27'
IntervalStartTime: '15.00.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.01.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
CreateDate: '2012-02-24'
CreateTime: '15.58.15'
...
PutCount: [2, 0]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [863, 0]
GetCount: [2, 0]
GetBytes: [863, 0]
...
QueueStatistics: 1
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
CreateDate: '2012-02-24'
CreateTime: '16.37.43'
...
PutCount: [2, 0]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [863, 0]
GetCount: [2, 0]
GetBytes: [863, 0]
...
2 Records Processed.
```

Un messaggio di richiesta e risposta è stato inviato nel primo intervallo e due nel secondo. È possibile dedurre che i messaggi di richiesta sono stati collocati in SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRVe i messaggi di risposta in SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Clustering: commutazione delle code di trasmissione del cluster
Pianificare come rendere effettive le modifiche alle code di trasmissione del cluster di un gestore code di produzione esistente.

Prima di iniziare

Se si riduce il numero di messaggi che il processo di commutazione deve trasferire alla nuova coda di trasmissione, la commutazione viene completata più rapidamente. Leggere Modalità di funzionamento del processo di commutazione del canale mittente del cluster in una coda di trasmissione differente per i motivi per cui si tenta di svuotare la coda di trasmissione prima di procedere ulteriormente.

Informazioni su questa attività

È possibile scegliere tra due modi per rendere effettive le modifiche alle code di trasmissione del cluster.

1. Consentire al gestore code di apportare le modifiche automaticamente. Questa è l'opzione predefinita. Il gestore code commuta i canali mittente del cluster con modifiche della coda di trasmissione in sospeso al successivo avvio di un canale mittente del cluster.
2. Apportare le modifiche manualmente. È possibile apportare le modifiche ad un canale mittente del cluster quando viene arrestato. È possibile passare da una coda di trasmissione cluster ad un'altra prima dell'avvio del canale mittente del cluster.

Quali fattori vengono presi in considerazione quando si decide quale delle due opzioni scegliere e come gestire l'interruttore?

Procedura

- Opzione 1: consentire al gestore code di apportare le modifiche automaticamente; consultare "Commutazione dei canali mittenti del cluster attivi in un'altra serie di code di trasmissione cluster" a pagina 65.

  Scegliere questa opzione se si desidera che il gestore code effettui lo switch.

  Un modo alternativo per descrivere questa opzione consiste nel dire che il gestore code commuta un canale mittente del cluster senza forzare l'arresto del canale. Hai la possibilità di forzare l'arresto del canale, e quindi avviare il canale, per fare in modo che lo switch avvenga prima. Lo switch viene avviato all'avvio del canale e viene eseguito mentre il canale è in esecuzione, il che è diverso dall'opzione 2. Nell'opzione 2, l'interruttore si verifica quando il canale viene arrestato.

  Se si sceglie questa opzione consentendo allo switch di verificarsi automaticamente, il processo di commutazione inizia all'avvio di un canale mittente del cluster. Se il canale non è arrestato, viene avviato dopo che diventa inattivo, se è presente un messaggio da elaborare. Se il canale è arrestato, avviarlo con il comando `START CHANNEL`.
  Il processo di commutazione viene completato non appena non rimangono messaggi per il canale mittente del cluster sulla coda di trasmissione che il canale stava servendo. Non appena questo è il caso, i messaggi appena arrivati per il canale mittente del cluster vengono memorizzati direttamente sulla nuova coda di trasmissione. Fino ad allora, i messaggi vengono memorizzati nella vecchia coda di trasmissione e il processo di commutazione trasferisce i messaggi dalla vecchia coda di trasmissione alla nuova coda di trasmissione. Il canale mittente del cluster inoltra i messaggi dalla nuova coda di trasmissione del cluster durante l'intero processo di commutazione.
  Quando il processo di commutazione viene completato dipende dallo stato del sistema. Se si stanno apportando modifiche in una finestra di manutenzione, valutare in anticipo se il processo di commutazione verrà completato in tempo. Il completamento in tempo dipende dal fatto che il numero di messaggi in attesa di trasferimento dalla vecchia coda di trasmissione raggiunga o meno lo zero.

  Il vantaggio del primo metodo è che è automatico. Uno svantaggio è che, se il tempo per apportare le modifiche alla configurazione è limitato a una finestra di manutenzione, è necessario essere certi di poter controllare il sistema per completare il processo di commutazione all'interno della finestra di manutenzione. Se non si è sicuri, l'opzione 2 potrebbe essere una scelta migliore.
- Opzione 2: apportare le modifiche manualmente; vedere "Commutazione di un canale mittente del cluster arrestato ad un'altra coda di trasmissione del cluster" a pagina 66.

  Scegliere questa opzione se si desidera controllare l'intero processo di commutazione manualmente o se si desidera commutare un canale arrestato o inattivo. È una buona scelta, se si stanno commutando alcuni canali mittente del cluster e si desidera eseguire lo switch durante una finestra di manutenzione.

  Una descrizione alternativa di questa opzione consiste nel commutare il canale mittente del cluster, mentre il canale mittente del cluster viene arrestato.

  Se si sceglie questa opzione, si ha il controllo completo quando si verifica l'interruttore.
  È possibile essere certi di completare il processo di commutazione in un periodo di tempo fisso, all'interno di una finestra di manutenzione. Il tempo impiegato dallo switch dipende dal numero di messaggi che devono essere trasferiti da una coda di trasmissione all'altra. Se i messaggi

continuano ad arrivare, potrebbe essere necessario del tempo prima che il processo trasferisca tutti i messaggi.

È possibile commutare il canale senza trasferire i messaggi dalla vecchia coda di trasmissione. Lo switch è "istantaneo".

Quando si riavvia il canale mittente del cluster, inizia l'elaborazione dei messaggi sulla coda di trasmissione appena assegnata ad esso.

Il vantaggio del secondo metodo è che si ha il controllo sul processo di commutazione. Lo svantaggio è che è necessario identificare i canali mittente del cluster da commutare, eseguire i comandi necessari e risolvere i canali in dubbio che potrebbero impedire l'arresto del canale mittente del cluster.

Concetti correlati

Come scegliere quale tipo di coda di trasmissione del cluster utilizzare
Come scegliere tra diverse opzioni di configurazione della coda di trasmissione del cluster.

Come funziona il processo di commutazione del canale mittente del cluster in una coda di trasmissione differente

Attività correlate

Clustering: configurazione di esempio di più code di trasmissione cluster
In questa attività si applicano le operazioni per pianificare più code di trasmissione del cluster a tre cluster sovrapposti. I requisiti sono di separare i flussi di messaggi in una coda cluster, da tutti gli altri flussi di messaggi e di memorizzare i messaggi per cluster differenti su code di trasmissione cluster differenti.

Commutazione dei canali mittenti del cluster attivi in un'altra serie di code di trasmissione cluster
Questa attività fornisce tre opzioni per la commutazione dei canali mittenti del cluster attivi. Un'opzione consiste nel consentire al gestore code di effettuare lo switch automaticamente, il che non influisce sulle applicazioni in esecuzione. Le altre opzioni sono l'arresto e l'avvio manuale dei canali o il riavvio del gestore code.

Prima di iniziare

Modificare la configurazione della coda di trasmissione cluster. È possibile modificare l'attributo del gestore code **DEFCLXQ** oppure aggiungere o modificare l'attributo **CLCHNAME** delle code di trasmissione.

Se si riduce il numero di messaggi che il processo di commutazione deve trasferire alla nuova coda di trasmissione, la commutazione viene completata più rapidamente. Leggere Modalità di funzionamento del processo di commutazione del canale mittente del cluster in una coda di trasmissione differente per i motivi per cui si tenta di svuotare la coda di trasmissione prima di procedere ulteriormente.

Informazioni su questa attività

Utilizzare i passaggi dell'attivit ... come base per elaborare un proprio piano per apportare modifiche alla configurazione della coda di trasmissione cluster.

Procedura

1. Opzionale: Registra lo stato del canale corrente

   Registrare lo stato dei canali correnti e salvati che servono le code di trasmissione del cluster. I seguenti comandi visualizzano lo stato associato alle coda di trasmissione del cluster di sistema. Aggiungere i propri comandi per visualizzare lo stato associato alle code di trasmissione cluster definite. Utilizzare una convenzione, come ad esempio `XMITQ.` *`ChannelName`*, per denominare le code di trasmissione del cluster definite per semplificare la visualizzazione dello stato del canale per tali code di trasmissione.

   ```
   DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
   DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
   ```

2. Commutare le code di trasmissione.
   - Non eseguire alcuna azione. Il gestore code commuta i canali mittenti del cluster quando vengono riavviati dopo essere stati arrestati o inattivi.

     Scegliere questa opzione se non si hanno regole o dubbi sulla modifica di una configurazione del gestore code. Le applicazioni in esecuzione non sono influenzate dalle modifiche.
   - Riavviare il gestore code. Tutti i canali mittente del cluster vengono arrestati e riavviati automaticamente su richiesta.

     Scegliere questa opzione per avviare immediatamente tutte le modifiche. Le applicazioni in esecuzione vengono interrotte dal gestore code quando viene arrestato e riavviato.
   - Arrestare i singoli canali mittente del cluster e riavviarli.

     Scegliere questa opzione per modificare immediatamente alcuni canali. Le applicazioni in esecuzione riscontrano un breve ritardo nel trasferimento dei messaggi tra l'avvio e l'arresto del canale di messaggi. Il canale mittente del cluster rimane in esecuzione, tranne durante il periodo di tempo in cui è stato arrestato. Durante il processo di commutazione i messaggi vengono consegnati alla vecchia coda di trasmissione, trasferiti alla nuova coda di trasmissione dal processo di commutazione e inoltrati dalla nuova coda di trasmissione dal canale mittente del cluster.
3. Opzionale: Monitora i canali mentre cambiano

   Visualizzare lo stato del canale e la profondità della coda di trasmissione durante lo switch. Il seguente esempio visualizza lo stato delle code di trasmissione del cluster di sistema.

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

4. Opzionale: Monitorare i messaggi AMQ7341 `La coda di trasmissione per il canale` *`ChannelName`* `passato dalla coda` *`QueueName`* `a` *`QueueName`* scritti nel log degli errori del gestore code.

Commutazione di un canale mittente del cluster arrestato ad un'altra coda di trasmissione del cluster
Se si sceglie di apportare modifiche manualmente, si apportano le modifiche ad un canale mittente del cluster quando viene arrestato e si passa da una coda di trasmissione cluster ad un'altra prima dell'avvio del canale mittente del cluster.

Prima di iniziare

È possibile apportare alcune modifiche alla configurazione e ora si desidera renderle effettive senza avviare i canali mittenti del cluster interessati. In alternativa, effettuare le modifiche di configurazione richieste come una delle fasi dell'attività.

Se si riduce il numero di messaggi che il processo di commutazione deve trasferire alla nuova coda di trasmissione, la commutazione viene completata più rapidamente. Leggere Modalità di funzionamento del processo di commutazione del canale mittente del cluster in una coda di trasmissione differente per i motivi per cui si tenta di svuotare la coda di trasmissione prima di procedere ulteriormente.

Informazioni su questa attività

Questa attività commuta le code di trasmissione servite da canali mittenti del cluster arrestati o inattivi. È possibile eseguire questa attività perché un canale mittente del cluster è stato arrestato e si desidera commutarne immediatamente la coda di trasmissione. Ad esempio, per qualche motivo un canale mittente del cluster non è in fase di avvio o ha qualche altro problema di configurazione. Per risolvere il problema, si decide di creare un canale mittente del cluster e di associare la coda di trasmissione per il vecchio canale mittente del cluster al nuovo canale mittente del cluster definito.

Uno scenario più probabile è quello in cui si desidera controllare quando viene eseguita la riconfigurazione delle code di trasmissione del cluster. Per controllare completamente la riconfigurazione, arrestare i canali, modificare la configurazione e quindi commutare le code di trasmissione.

Procedura

1. Arrestare i canali che si intende commutare
   a) Arrestare tutti i canali in esecuzione o inattivi che si intende commutare. L'arresto di un canale mittente del cluster inattivo ne impedisce l'avvio mentre si stanno apportando modifiche alla configurazione.

   ```
   STOP CHANNEL(ChannelName) MODE(QUIESCSE) STATUS(STOPPED)
   ```

2. Opzionale: Apportare le modifiche alla configurazione.

   Ad esempio, consultare "Clustering: configurazione di esempio di più code di trasmissione cluster" a pagina 55.

3. Passare i canali mittente del cluster alle nuove code di trasmissione del cluster.

   Multi Su Multipiattaforme, immettere il seguente comando:

   ```
   runswchl -m QmgrName -c ChannelName
   ```

   z/OS Su z/OS, utilizzare la funzione SWITCH del comando CSQUTIL per commutare i messaggi o monitorare ciò che accade. utilizzare il seguente comando.

   ```
   SWITCH CHANNEL(channel_name) MOVEMSGS(YES)
   ```

   Per ulteriori informazioni, consultare Funzione SWITCH.

   Il comando **runswchl**, o CSQUTIL SWITCH, trasferisce i messaggi sulla vecchia coda di trasmissione alla nuova coda di trasmissione. Quando il numero di messaggi sulla vecchia coda di trasmissione per questo canale raggiunge lo zero, lo switch viene completato. Il comando è sincrono. Il comando scrive i messaggi di avanzamento nella finestra durante il processo di commutazione.

   Durante la fase di trasferimento, i messaggi nuovi ed esistenti destinati al canale mittente del cluster vengono trasferiti alla nuova coda di trasmissione.

   Poiché il canale mittente del cluster è arrestato, i messaggi si accumulano nella nuova coda di trasmissione. Confrontare il canale mittente del cluster arrestato con il passo "2" a pagina 66 in "Commutazione dei canali mittenti del cluster attivi in un'altra serie di code di trasmissione cluster" a pagina 65. In questo passo, il canale mittente del cluster è in esecuzione, quindi i messaggi non si accumulano necessariamente sulla nuova coda di trasmissione.

4. Opzionale: Monitora i canali mentre cambiano

   In una finestra comandi differente, visualizzare la profondità della coda di trasmissione durante lo switch. Il seguente esempio visualizza lo stato delle code di trasmissione del cluster di sistema.

   ```
   DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
   ```

5. Opzionale: Monitorare i messaggi AMQ7341 `La coda di trasmissione per il canale` *ChannelName* `passato dalla coda` *QueueName* `a` *QueueName* scritti nel log degli errori del gestore code.
6. Riavviare i canali mittenti del cluster arrestati.

   I canali non vengono avviati automaticamente, poiché sono stati arrestati e vengono inseriti nello stato ARRESTATO .

   ```
   START CHANNEL(ChannelName)
   ```

Riferimenti correlati

runswchl

Risoluzione canale
Arresto canale

Clustering: procedure ottimali di migrazione e modifica
Questo argomento fornisce una guida per la pianificazione e la gestione dei cluster IBM MQ . Queste informazioni sono una guida basata sul test e sul feedback dei clienti.

1. “Spostamento di oggetti in un cluster” a pagina 68 (Procedure ottimali per spostare gli oggetti all'interno di un cluster, senza installare fix pack o nuove versioni di IBM MQ).
2. “Aggiornamenti e installazioni di manutenzione” a pagina 69 (Procedure ottimali per mantenere attiva e in esecuzione un'architettura cluster funzionante, applicando la manutenzione o gli aggiornamenti e testando la nuova architettura).

Spostamento di oggetti in un cluster

Applicazioni e relative code

Quando è necessario spostare un'istanza della coda ospitata su un gestore code per essere ospitata su un altro gestore code, è possibile lavorare con i parametri di bilanciamento del carico di lavoro per garantire una transizione graduale.

Creare un'istanza della coda in cui deve essere ospitata di nuovo, ma utilizzare le impostazioni di bilanciamento del carico di lavoro del cluster per continuare a inviare messaggi all'istanza originale fino a quando la propria applicazione non è pronta per lo switch. Ciò si ottiene con la seguente procedura:

1. Impostare la proprietà **CLWLRANK** della coda esistente su un valore elevato, ad esempio cinque.
2. Creare la nuova istanza della coda e impostare la relativa proprietà **CLWLRANK** su zero.
3. Completare qualsiasi ulteriore configurazione del nuovo sistema, ad esempio la distribuzione e l'avvio dell'utilizzo delle applicazioni rispetto alla nuova istanza della coda.
4. Impostare la proprietà **CLWLRANK** della nuova istanza della coda in modo che sia superiore all'istanza originale, ad esempio nove.
5. Consentire all'istanza della coda originale di elaborare i messaggi accodati nel sistema e quindi eliminare la coda.

Spostamento di interi gestori code

Se il gestore code rimane sullo stesso host, ma l'indirizzo IP viene modificato, il processo è il seguente:

- Il DNS, se usato correttamente, può aiutare a semplificare il processo. Per informazioni sull'utilizzo di DNS impostando l'attributo del canale Nome connessione (CONNAME) , consultare ALTER CHANNEL.
- Se si sposta un repository completo, assicurarsi di disporre di almeno un altro repository completo che sia in esecuzione senza problemi (ad esempio, nessun problema con lo stato del canale) prima di apportare le modifiche.
- Sospendere il gestore code utilizzando il comando SUSPEND QMGR per evitare la creazione di traffico.
- Modificare l'indirizzo IP del computer. Se la definizione di canale CLUSRCVR utilizza un indirizzo IP nel campo CONNAME, modificare questa voce di indirizzo IP. Potrebbe essere necessario eseguire il flush della cache DNS per garantire che gli aggiornamenti siano disponibili ovunque.
- Quando il gestore code si riconnette ai repository completi, le definizioni automatiche del canale si risolvono automaticamente.
- Se il gestore code ospitava un repository completo e l'indirizzo IP cambia, è importante assicurarsi che le parti vengano commutate il prima possibile per puntare i canali CLUSSDR definiti manualmente alla nuova ubicazione. Finché questo switch non viene eseguito, questi gestori code potrebbero essere in grado di contattare solo il repository completo rimanente (non modificato) e potrebbero essere visualizzati messaggi di avvertenza relativi alla definizione di canale non corretta.

- Riprendere il gestore code utilizzando il comando RESUME QMGR .

Se il gestore code deve essere spostato su un altro host, è possibile copiare i dati del gestore code e ripristinare da un backup. Questo processo non è tuttavia consigliato, a meno che non vi siano altre opzioni; potrebbe essere preferibile creare un gestore code su una nuova macchina e replicare le code e le applicazioni come descritto nella sezione precedente. Questa situazione fornisce un meccanismo di rollover / rollback semplice.

Se si è determinati a spostare un gestore code completo utilizzando il backup, attenersi alle seguenti procedure ottimali:

- Considerare l'intero processo come un ripristino di un gestore code dal backup, applicando tutti i processi che di solito si utilizzano per il recupero del sistema come appropriato per l'ambiente del sistema operativo.
- Utilizzare il comando **REFRESH CLUSTER** dopo la migrazione per eliminare tutte le informazioni sul cluster conservate localmente (inclusi i canali definiti automaticamente che sono in dubbio) e forzarne la ricostruzione.

  Nota: Per i cluster di grandi dimensioni, l'utilizzo del comando **REFRESH CLUSTER** può danneggiare il cluster mentre è in esecuzione e, di nuovo, a intervalli di 27 giorni, quando gli oggetti del cluster inviano automaticamente aggiornamenti sullo stato a tutti i gestori code interessati. Consultare Refreshing in a large cluster can affect performance and availability of the cluster.

Quando si crea un gestore code e si replica l'impostazione da un gestore code esistente nel cluster (come descritto in precedenza in questo argomento), non considerare mai i due gestori code differenti come se fossero gli stessi. In particolare, non assegnare a un nuovo gestore code lo stesso nome e indirizzo IP. Il tentativo di 'inserire ' un gestore code di sostituzione è una causa frequente di problemi nei cluster IBM MQ . La cache prevede di ricevere gli aggiornamenti incluso l'attributo **QMID** e lo stato può essere danneggiato.

Se due gestori code differenti vengono creati accidentalmente con lo stesso nome, si consiglia di utilizzare il comando RESET CLUSTER **QMID** per espellere la voce non corretta dal cluster.

Aggiornamenti e installazioni di manutenzione

Evitare il cosiddetto scenario big bang (ad esempio, l'arresto di tutte le attività del cluster e del gestore code, l'applicazione di tutti gli aggiornamenti e la manutenzione a tutti i gestori code, quindi l'avvio di tutto contemporaneamente). I cluster sono progettati per funzionare ancora con più versioni di gestori code coesistenti, quindi si consiglia un approccio di manutenzione pianificato e graduale.

Disporre di un piano di backup:

- Hai fatto dei backup?
- Evitare di utilizzare immediatamente la nuova funzionalità del cluster: attendere fino a quando non si è certi che tutti i gestori code siano aggiornati al nuovo livello e si è certi che non verrà eseguito il rollback di nessuno di essi. L'utilizzo di una nuova funzione cluster in un cluster in cui alcuni gestori code sono ancora a un livello precedente può portare a un comportamento non definito.

  Un repository memorizza un record ricevuto nella propria versione. Se il record che riceve è di una versione successiva, gli attributi della versione successiva vengono eliminati quando il record viene memorizzato. Un gestore code IBM MQ 9.3 che riceve informazioni su un gestore code IBM MQ 9.4 memorizza solo IBM MQ 9.3 informazioni. Un repository IBM MQ 9.4 che riceve un record IBM MQ 9.3 memorizza i valori predefiniti per gli attributi introdotti nella versione successiva. I valori predefiniti definiscono i valori per gli attributi che non vengono inclusi nel record ricevuto.

Migrare prima i repository completi. Anche se possono trasmettere informazioni che non comprendono, non possono persistere, quindi non è l'approccio raccomandato a meno che non sia assolutamente necessario. Per ulteriori informazioni, fare riferimento alla sezione Migrazione del cluster del gestore code.

Cluster: utilizzo delle procedure consigliate per REFRESH CLUSTER
Utilizzare il comando **REFRESH CLUSTER** per eliminare tutte le informazioni contenute localmente su un cluster e ricreare tali informazioni dai repository completi nel cluster. Non è necessario utilizzare questo comando, tranne in circostanze eccezionali. Se hai bisogno di usarlo, ci sono considerazioni speciali su come usarlo. Queste informazioni sono una guida basata sul test e sul feedback dei clienti.

Eseguire REFRESH CLUSTER solo se necessario

La tecnologia del cluster IBM MQ garantisce che qualsiasi modifica alla configurazione del cluster, ad esempio una modifica a una coda cluster, diventi automaticamente nota a qualsiasi membro del cluster che deve conoscere le informazioni. Non è necessario adottare ulteriori misure amministrative per ottenere tale diffusione delle informazioni.

Se tali informazioni non raggiungono i gestori code nel cluster in cui sono richieste, ad esempio una coda cluster non è riconosciuta da un altro gestore code nel cluster quando un'applicazione tenta di aprirla per la prima volta, ciò implica un problema nell'infrastruttura del cluster. Ad esempio, è possibile che un canale non possa essere avviato tra un gestore code e un gestore code del repository completo. Pertanto, qualsiasi situazione in cui si osservino incongruenze deve essere esaminata. Se possibile, risolvere la situazione senza utilizzare il comando **REFRESH CLUSTER** .

In rare circostanze documentate altrove in questa documentazione del prodotto, o quando richiesto dal supporto IBM , è possibile utilizzare il comando **REFRESH CLUSTER** per eliminare tutte le informazioni conservate localmente su un cluster e ricreare tali informazioni dai repository completi nel cluster.

L'aggiornamento in un cluster di grandi dimensioni può influire sulle prestazioni e sulla disponibilità del cluster

L'utilizzo del comando **REFRESH CLUSTER** può essere disruttivo per il cluster mentre è in corso, ad esempio creando un aumento improvviso del lavoro per i repository completi mentre elaborano la propagazione delle risorse del cluster del gestore code. Se si sta aggiornando in un cluster di grandi dimensioni (ovvero, molte centinaia di gestori code) è necessario evitare l'utilizzo del comando nel lavoro quotidiano, se possibile, e utilizzare metodi alternativi per correggere specifiche incongruenze. Ad esempio, se una coda cluster non viene propagata correttamente nel cluster, una tecnica di analisi iniziale di aggiornamento della definizione della coda cluster, ad esempio la modifica della relativa descrizione, propaga la configurazione della coda nel cluster. Questo processo può aiutare a identificare il problema e potenzialmente a risolvere un'incongruenza temporanea.

Se non è possibile utilizzare metodi alternativi e si deve eseguire **REFRESH CLUSTER** in un cluster di grandi dimensioni, è necessario farlo in orari non di punta o durante una finestra di manutenzione per evitare l'impatto sui carichi di lavoro degli utenti. Si dovrebbe anche evitare di aggiornare un cluster di grandi dimensioni in un singolo batch, e di sfalsare l'attività come spiegato in “Evitare problemi di prestazioni e disponibilità quando gli oggetti cluster inviano aggiornamenti automatici” a pagina 70.

Evitare problemi di prestazioni e disponibilità quando gli oggetti cluster inviano aggiornamenti automatici

Dopo che un nuovo oggetto cluster è stato definito su un gestore code, un aggiornamento per questo oggetto viene generato ogni 27 giorni dal momento della definizione e inviato a ogni repository completo nel cluster e in seguito a qualsiasi altro gestore code interessato. Quando si immette il comando **REFRESH CLUSTER** su un gestore code, si reimposta l'orologio per questo aggiornamento automatico su tutti gli oggetti definiti localmente nel cluster specificato.

Se si aggiorna un cluster di grandi dimensioni (ossia, molte centinaia di gestori code) in un singolo batch o in altre circostanze, come la ricreazione di un sistema dal backup della configurazione, dopo 27 giorni tutti questi gestori code pubblicizzeranno nuovamente tutte le relative definizioni di oggetti nei repository completi contemporaneamente. Ciò potrebbe di nuovo causare un'esecuzione del sistema significativamente più lenta, o addirittura diventare non disponibile, fino a quando tutti gli aggiornamenti non sono stati completati. Pertanto, quando è necessario aggiornare o ricreare più gestori code in un

cluster di grandi dimensioni, è necessario scaglionare l'attività per diverse ore o diversi giorni, in modo che i successivi aggiornamenti automatici non influiscono regolarmente sulle prestazioni del sistema.

La coda di cronologia cluster di sistema

Quando viene eseguito un **REFRESH CLUSTER** , il gestore code acquisisce un'istantanea dello stato del cluster prima dell'aggiornamento e la memorizza su SYSTEM.CLUSTER.HISTORY.QUEUE (SCHQ) se è definito sul gestore code. Questa istantanea è solo per scopi di servizio IBM , in caso di problemi successivi con il sistema.

SCHQ è definito per impostazione predefinita sui gestori code distribuiti all'avvio. Per la migrazione z/OS , SCHQ deve essere definito manualmente.

I messaggi sullo SCHQ scadono dopo tre mesi.

Concetti correlati

"REFRESH CLUSTER considerazioni per i cluster di pubblicazione / sottoscrizione" a pagina 107
Immettendo il comando **REFRESH CLUSTER** il gestore code elimina temporaneamente le informazioni conservate localmente su un cluster, inclusi gli argomenti del cluster e le relative sottoscrizioni proxy associate.

Riferimenti correlati

Problemi dell'applicazione durante l'esecuzione di REFRESH CLUSTER

Riferimento comandi MQSC: REFRESH CLUSTER

Clustering: disponibilità, più istanze e ripristino di emergenza

Questo argomento fornisce una guida per la pianificazione e la gestione dei cluster IBM MQ . Queste informazioni sono una guida basata sul test e sul feedback dei clienti.

IBM MQ Il clustering stesso non è una soluzione ad alta disponibilità, ma in alcune circostanze può essere utilizzato per migliorare la disponibilità dei servizi utilizzando IBM MQ, ad esempio disponendo di più istanze di una coda su gestori code differenti. Questa sezione fornisce una guida per garantire che l'infrastruttura IBM MQ sia il più disponibile possibile in modo che possa essere utilizzata in tale architettura.

Nota: Altre soluzioni di alta disponibilità e ripristino di emergenza sono disponibili per IBM MQ, consultare Configurazione dell'alta disponibilità, ripristino e riavvio.

Disponibilità di risorse cluster

Il motivo per cui si consiglia di conservare due repository completi è che la perdita di uno non è critica per il corretto funzionamento del cluster. Anche se entrambi diventano non disponibili, esiste un periodo di tolleranza di 60 giorni per le conoscenze esistenti detenute da repository parziali, anche se le risorse nuove o non precedentemente accedute (code ad esempio) non sono disponibili in questo evento.

Utilizzo dei cluster per migliorare la disponibilità delle applicazioni

Un cluster può aiutare nella progettazione di applicazioni ad alta disponibilità (ad esempio un'applicazione server di tipo richiesta / risposta), utilizzando più istanze della coda e dell'applicazione. Se necessario, gli attributi di priorità possono dare la preferenza all'applicazione 'live', a meno che un gestore code o un canale non diventino, ad esempio, non disponibili. Ciò è utile per passare rapidamente all'elaborazione di nuovi messaggi quando si verifica un problema.

Tuttavia, i messaggi che sono stati consegnati a un determinato gestore code in un cluster vengono conservati solo su tale istanza della coda e non sono disponibili per l'elaborazione fino a quando tale gestore code non viene recuperato. Per questo motivo, per l'alta disponibilità dei dati reali, è possibile considerare altre tecnologie come i gestori code a più istanze.

Gestori code a più istanze

Software High Availability (multi - istanza) è un'offerta integrata per mantenere disponibili i messaggi esistenti. Fare riferimento a Utilizzo di IBM MQ con configurazioni ad alta disponibilità, Creazione di un gestore code a più istanzee alla seguente sezione per ulteriori informazioni. Qualsiasi gestore code in un cluster può essere reso altamente disponibile utilizzando questa tecnica, purché tutti i gestori code nel cluster siano in esecuzione almeno IBM WebSphere MQ 7.0.1. Se i gestori code nel cluster si

trovano a livelli precedenti, potrebbero perdere la connettività con i gestori code a più istanze se viene eseguito il failover su un IP secondario.

Come discusso precedentemente in questo argomento, finché sono configurati due repository completi, sono quasi per loro natura altamente disponibili. Se necessario, è possibile utilizzare i gestori code del software IBM MQ ad alta disponibilità / a più istanze per repository completi. Non esiste un motivo valido per utilizzare questi metodi e, in effetti, per interruzioni temporanee, tali metodi potrebbero causare ulteriori costi di prestazioni durante il failover. L'utilizzo della HA del software invece di eseguire due repository completi è sconsigliato perché in caso di interruzione di un singolo canale, ad esempio, non necessariamente eseguirebbe il failover, ma potrebbe lasciare repository parziali non in grado di eseguire la query per le risorse cluster.

Ripristino di emergenza

Il ripristino di emergenza, ad esempio il ripristino da quando i dischi che memorizzano i dati di un gestore code diventano danneggiati, è difficile da eseguire correttamente; IBM MQ può aiutare, ma non può farlo automaticamente. L'unica opzione di ripristino di emergenza 'true' in IBM MQ (escludendo qualsiasi sistema operativo o altre tecnologie di replica sottostanti) è il ripristino da un backup. Ci sono alcuni punti specifici del cluster da considerare in queste situazioni:

- Prestare attenzione quando si verificano scenari di ripristino di emergenza. Ad esempio, se si verifica l'operazione dei gestori code di backup, prestare attenzione quando si portano in linea nella stessa rete poiché è possibile unirsi accidentalmente al cluster attivo e iniziare a 'rubare' i messaggi ospitando le stesse code denominate dei gestori code del cluster attivo.
- Il test del ripristino di emergenza non deve interferire con un cluster attivo in esecuzione. Le tecniche per evitare interferenze includono:
  - Completare la separazione o la separazione della rete a livello di firewall.
  - z/OS Non avvio dell'avvio del canale o dello spazio di indirizzo z/OS **chinit** .
  - Non emettere il certificato TLS attivo per il sistema di ripristino di emergenza fino a quando, o a meno che, non si verifichi uno scenario di ripristino di emergenza effettivo.
- Quando si ripristina un backup di un gestore code nel cluster, è possibile che il backup non sia sincronizzato con il resto del cluster. Il comando **REFRESH CLUSTER** può risolvere gli aggiornamenti e sincronizzarli con il cluster, ma il comando **REFRESH CLUSTER** deve essere utilizzato come ultima risorsa. Consultare “Cluster: utilizzo delle procedure consigliate per REFRESH CLUSTER” a pagina 70. Esaminare tutta la documentazione del processo interno e la documentazione di IBM MQ per vedere se è stato mancato un semplice passo prima di ricorrere all'utilizzo del comando.
- Come per qualsiasi ripristino, le applicazioni devono gestire la ripetizione e la perdita di dati. È necessario decidere se cancellare le code in uno stato noto o se ci sono informazioni sufficienti altrove per gestire le ripetizioni.

Pianificazione della rete di pubblicazione / sottoscrizione distribuita

È possibile creare una rete di gestori code in cui le sottoscrizioni create su un gestore code riceveranno i messaggi corrispondenti pubblicati da un'applicazione connessa a un altro gestore code nella rete. Per scegliere una topologia adatta, è necessario considerare i requisiti per il controllo manuale, la dimensione della rete, la frequenza di modifica, la disponibilità e la scalabilità.

Prima di iniziare

Questa attività presuppone che l'utente comprenda quali sono le reti di pubblicazione / sottoscrizione distribuite e come funzionano. Per una panoramica tecnica, consultare Distributed publish/subscribe networks.

Informazioni su questa attività

Esistono tre topologie di base per una rete di pubblicazione / sottoscrizione:

- Cluster instradato direttamente
- Cluster instradato host argomento

- Gerarchia

Per le prime due topologie, il punto di partenza è una configurazione cluster IBM MQ . La terza topologia può essere creata con o senza un cluster. Consultare“Pianificazione delle code e dei cluster distribuiti” a pagina 20per informazioni sulla pianificazione della rete di gestori code sottostante.

Un *cluster instradato direttamente* è la topologia più semplice da configurare quando un cluster è già presente. Qualsiasi argomento definito su qualsiasi gestore code viene automaticamente reso disponibile su ogni gestore code nel cluster e le pubblicazioni vengono instradate direttamente da qualsiasi gestore code a cui si connette un'applicazione di pubblicazione, a ciascuno dei gestori code in cui esistono sottoscrizioni corrispondenti. Questa semplicità di configurazione si basa sul fatto che IBM MQ mantiene un alto livello di condivisione di informazioni e connettività tra ogni gestore code nel cluster. Per reti piccole e semplici (ossia un numero ridotto di gestori code e un insieme abbastanza statico di publisher e sottoscrittori), ciò è accettabile. Tuttavia, se utilizzato in ambienti più grandi o più dinamici, il sovraccarico potrebbe essere proibitivo. Consultare“Instradamento diretto nei cluster di pubblicazione / sottoscrizione” a pagina 78.

Un *cluster instradato all'host dell'argomento* fornisce lo stesso vantaggio di un cluster instradato direttamente, rendendo qualsiasi argomento definito su qualsiasi gestore code nel cluster automaticamente disponibile su ogni gestore code nel cluster. Tuttavia, i cluster instradati dell'host argomento richiedono di scegliere attentamente i gestori code che ospitano ciascun argomento, poiché tutte le informazioni e le pubblicazioni per tale argomento passano attraverso tali gestori code dell'host argomento. Ciò significa che il sistema non deve gestire i canali e i flussi di informazioni tra tutti i gestori code. Tuttavia, significa anche che le pubblicazioni potrebbero non essere più inviate direttamente ai sottoscrittori, ma potrebbero essere instradate tramite un gestore code dell'host argomento. Per questi motivi, potrebbe essere necessario un ulteriore carico sul sistema, in particolare sui gestori code che ospitano gli argomenti, pertanto è necessaria un'attenta pianificazione della topologia. Questa topologia è particolarmente efficace per le reti che contengono molti gestori code o che ospitano una serie dinamica di publisher e sottoscrittori (ossia, publisher o sottoscrittori che vengono aggiunti o rimossi di frequente). È possibile definire ulteriori host argomento per migliorare la disponibilità degli instradamenti e per scalare orizzontalmente il carico di lavoro di pubblicazione. Consultare“Instradamento host argomento nei cluster di pubblicazione / sottoscrizione” a pagina 83.

Una *gerarchia* richiede la maggior parte della configurazione manuale per essere impostata ed è la topologia più difficile da modificare. È possibile configurare manualmente le relazioni tra ciascun gestore code nella gerarchia e le sue relazioni dirette. Una volta configurate le relazioni, le pubblicazioni (come per le due topologie precedenti) verranno instradate alle sottoscrizioni su altri gestori code nella gerarchia. Le pubblicazioni vengono instradate utilizzando le relazioni della gerarchia. Ciò consente la configurazione di topologie molto specifiche per soddisfare requisiti differenti, ma può anche risultare in pubblicazioni che richiedono molti "hop" tramite gestori code intermedi per raggiungere le sottoscrizioni. Esiste sempre un solo instradamento attraverso una gerarchia per una pubblicazione, quindi la disponibilità di ogni gestore code è critica. Le gerarchie sono generalmente preferibili solo quando non è possibile configurare un singolo cluster, ad esempio quando si estendono più organizzazioni. Consultare“Instradamento nelle gerarchie di pubblicazione / sottoscrizione” a pagina 108.

Se necessario, le tre topologie di cui sopra possono essere combinate per risolvere specifici requisiti topografici. Per un esempio, consultare Combinazione degli spazi argomento di più cluster.

Per scegliere una topologia adatta per la propria rete di pubblicazione / sottoscrizione distribuita, è necessario considerare le seguenti domande generali:

- Quanto sarà grande la tua rete?
- Di quale controllo manuale hai bisogno per la sua configurazione?
- Quanto sarà dinamico il sistema, sia in termini di argomenti e sottoscrizioni, sia in termini di gestori code?
- Quali sono i requisiti di disponibilità e scalabilità?
- Tutti i gestori code possono connettersi direttamente tra loro?

Procedura

- Stimare la dimensione della rete.
  a) Stimare il numero di argomenti necessari.
  b) Stimare quanti publisher e sottoscrittori si prevede di avere.
  c) Stimare il numero di gestori code coinvolti nelle attività di pubblicazione / sottoscrizione.

  Consultare anche“Clustering di pubblicazione / sottoscrizione: procedure ottimali” a pagina 93, in particolare le seguenti sezioni:

  – Modalità di dimensionamento del sistema
  – Motivi per limitare il numero di gestori code del cluster coinvolti nell'attività di pubblicazione / sottoscrizione
  – Come decidere quali argomenti raggruppare

  Se la rete avrà molti gestori code e gestirà molti publisher e sottoscrittori, è probabilmente necessario utilizzare un cluster instradato dell'host argomento o una gerarchia. I cluster instradati direttamente non richiedono quasi alcuna configurazione manuale e possono essere una buona soluzione per reti piccole o statiche.
- Considerare quanto controllo manuale è necessario su quale gestore code ospita ciascun argomento, publisher o sottoscrittore.
  a) Considerare se alcuni dei gestori code sono meno capaci di altri.
  b) Considerare se i collegamenti di comunicazione ad alcuni dei gestori code sono più fragili rispetto ad altri.
  c) Identificare i casi in cui si prevede che un argomento abbia molte pubblicazioni e pochi sottoscrittori.
  d) Identificare i casi in cui si prevede che un argomento abbia molti sottoscrittori e poche pubblicazioni.

  In tutte le topologie, le pubblicazioni vengono consegnate alle sottoscrizioni su altri gestori code. In un cluster instradato direttamente tali pubblicazioni prendono il percorso più breve alle sottoscrizioni. In un cluster instradato dell'host argomento o in una gerarchia, è possibile controllare l'instradamento seguito dalle pubblicazioni. Se i gestori code differiscono per capacità o hanno livelli differenti di disponibilità e connettività, è probabile che si desideri assegnare carichi di lavoro specifici a gestori code specifici. È possibile effettuare questa operazione utilizzando un cluster instradato dell'host argomento o una gerarchia.

  In tutte le topologie, la co - localizzazione delle applicazioni di pubblicazione sullo stesso gestore code delle sottoscrizioni, quando possibile, riduce i sovraccarichi e massimizza le prestazioni. Per i cluster instradati dell'host argomento, considerare l'inserimento di publisher o sottoscrittori sui gestori code che ospitano l'argomento. Questa operazione rimuove eventuali "hop" supplementari tra i gestori code per passare una pubblicazione a un sottoscrittore. Questo approccio è particolarmente efficace nei casi in cui un argomento ha molti editori e pochi sottoscrittori, o molti sottoscrittori e pochi editori. Consultare, ad esempio, Instradamento host argomento utilizzando publisher o sottoscrittori centralizzati.

  Consultare anche“Clustering di pubblicazione / sottoscrizione: procedure ottimali” a pagina 93, in particolare le seguenti sezioni:

  – Come decidere quali argomenti raggruppare
  – Posizione di pubblicazione e sottoscrizione
- Considerare la dinamica dell'attività di rete.
  a) Stimare la frequenza con cui i sottoscrittori verranno aggiunti e rimossi su argomenti differenti.

  Ogni volta che una sottoscrizione viene aggiunta o rimossa da un gestore code ed è la prima o l'ultima sottoscrizione per quella specifica stringa di argomenti, tali informazioni vengono comunicate ad altri gestori code nella topologia. In un cluster instradato direttamente e in una

gerarchia, queste informazioni di sottoscrizione vengono propagate a tutti i gestori code nella topologia, indipendentemente dal fatto che abbiano o meno dei publisher sull'argomento. Se la topologia è composta da molti gestori code, questo potrebbe essere un sovraccarico delle prestazioni significativo. In un cluster instradato dell'host argomento, queste informazioni vengono propagate solo ai gestori code che ospitano un argomento del cluster associato alla stringa di argomenti della sottoscrizione.

Consultare anche la sezione Modifica della sottoscrizione e stringhe di argomenti dinamici di"Clustering di pubblicazione / sottoscrizione: procedure ottimali" a pagina 93.

Nota: In sistemi molto dinamici, in cui l'insieme di molte stringhe di argomenti univoci viene modificato in modo rapido e costante, potrebbe essere meglio passare alla modalità "pubblica ovunque" . Vedere Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione.

b) Considerare quanto sono dinamici i gestori code nella topologia.

Una gerarchia richiede che ogni modifica nel gestore code nella topologia venga inserita o rimossa manualmente dalla gerarchia, con attenzione quando si modificano i gestori code ai livelli più alti della gerarchia. I gestori code in una gerarchia generalmente utilizzano anche connessioni di canale configurate manualmente. È necessario mantenere queste connessioni, aggiungendo e rimuovendo canali quando i gestori code vengono aggiunti e rimossi dalla gerarchia.

In un cluster di pubblicazione / sottoscrizione, i gestori code vengono automaticamente connessi a qualsiasi altro gestore code richiesto quando si uniscono per la prima volta al cluster e diventano automaticamente consapevoli degli argomenti e delle sottoscrizioni.

- Considera la disponibilità del tuo instradamento e i requisiti di scalabilità del traffico di pubblicazione.
  a) Decidere se è necessario disporre sempre di un instradamento disponibile da un gestore code di pubblicazione a un gestore code di sottoscrizione, anche quando un gestore code non è disponibile.
  b) Considera la scalabilità di cui hai bisogno per la rete. Decidere se il livello di traffico di pubblicazione è troppo alto per essere instradato attraverso un singolo gestore code o canale e se tale livello di traffico di pubblicazione deve essere gestito da un singolo ramo di argomento o può essere distribuito su più rami di argomento.
  c) Considerare se è necessario mantenere l'ordine dei messaggi.

  Poiché un cluster instradato direttamente invia i messaggi direttamente dai gestori code di pubblicazione ai gestori code di sottoscrizione, non è necessario considerare la disponibilità dei gestori code intermedi lungo la rotta. Allo stesso modo, il ridimensionamento ai gestori code intermedi non è una considerazione. Tuttavia, come precedentemente menzionato, il sovraccarico di gestione automatica dei canali e dei flussi di informazioni tra tutti i gestori code nel cluster può influire in modo significativo sulle prestazioni, soprattutto in un ambiente di grandi dimensioni o dinamico.

  Un cluster di host argomento instradato può essere ottimizzato per singoli argomenti. È possibile garantire che ogni ramo della struttura ad albero degli argomenti che ha un considerevole carico di lavoro di pubblicazione sia definito su un gestore code differente e che ogni gestore code sia sufficientemente performante e disponibile per il carico di lavoro previsto per tale ramo della struttura ad albero degli argomenti. È inoltre possibile migliorare ulteriormente la disponibilità e il ridimensionamento orizzontale definendo ogni argomento su più gestori code. Ciò consente al sistema di instradare i gestori code dell'host argomento non disponibili e di bilanciare il traffico di pubblicazione tra di essi. Tuttavia, quando si definisce un determinato argomento su più gestori code, vengono introdotti anche i seguenti vincoli:

  – Si perde l'ordine dei messaggi tra le pubblicazioni.
  – Non è possibile utilizzare pubblicazioni conservate. Consultare"Considerazioni di progettazione per le pubblicazioni conservate nei cluster di pubblicazione / sottoscrizione" a pagina 106.

  Non è possibile configurare l'alta disponibilità o la scalabilità dell'instradamento in una gerarchia attraverso più instradamenti.

  Vedi anche la sezione Traffico di pubblicazione di"Clustering di pubblicazione / sottoscrizione: procedure ottimali" a pagina 93.

- In base a questi calcoli, utilizzare i link forniti per decidere se utilizzare un cluster instradato dell'host argomento, un cluster instradato direttamente, una gerarchia o una combinazione di queste topologie.

Operazioni successive

Si è ora pronti a configurare la rete di pubblicazione / sottoscrizione distribuita.

Attività correlate

Configurazione di un cluster di gestore code

Configurazione dell'accodamento distribuito

Configurazione di un cluster di pubblicazione / sottoscrizione

Connessione di un gestore code a una gerarchia di pubblicazione / sottoscrizione

Progettazione di cluster di pubblicazione / sottoscrizione

Esistono due topologie cluster di pubblicazione / sottoscrizione di base: *instradamento diretto* e *instradamento host argomento*. Ognuno ha vantaggi diversi. Quando si progetta il cluster di pubblicazione / sottoscrizione, scegliere la topologia che meglio si adatta ai requisiti di rete previsti.

Per una panoramica delle due topologie di cluster di pubblicazione / sottoscrizione, vedere Cluster di pubblicazione / sottoscrizione. Per aiutarti a valutare i tuoi requisiti di rete, vedi “Pianificazione della rete di pubblicazione / sottoscrizione distribuita” a pagina 72 e “Clustering di pubblicazione / sottoscrizione: procedure ottimali” a pagina 93.

In generale, entrambe le topologie di cluster forniscono i seguenti vantaggi:

- Configurazione semplice su una topologia di cluster point - to - point.
- Gestione automatica dei gestori code che si uniscono e escono dal cluster.
- Facilità di ridimensionamento per ulteriori sottoscrizioni e publisher, aggiungendo ulteriori gestori code e distribuendo le sottoscrizioni e i publisher aggiuntivi.

Tuttavia, le due topologie hanno vantaggi diversi quando i requisiti diventano più specifici.

Cluster di pubblicazione / sottoscrizione instradati diretti

Con l'instradamento diretto, tutti i gestori code nel cluster inviano le pubblicazioni dalle applicazioni connesse direttamente a qualsiasi altro gestore code nel cluster con una sottoscrizione corrispondente.

Un cluster di pubblicazione / sottoscrizione instradato direttamente fornisce i seguenti vantaggi:

- I messaggi destinati a una sottoscrizione su un gestore code specifico nello stesso cluster vengono trasportati direttamente a tale gestore code e non è necessario passare attraverso un gestore code intermedio. Ciò può migliorare le prestazioni rispetto a una topologia instradata dell'host argomento o a una topologia gerarchica.
- Poiché tutti i gestori code sono direttamente connessi tra loro, non esiste un singolo punto di errore nell'infrastruttura di instradamento di questa topologia. Se un gestore code non è disponibile, le sottoscrizioni su altri gestori code nel cluster sono ancora in grado di ricevere messaggi dai publisher sui gestori code disponibili.
- È molto semplice da configurare, soprattutto su un cluster esistente.

Elementi da considerare quando si utilizza un cluster di pubblicazione / sottoscrizione instradato direttamente:

- Tutti i gestori code nel cluster vengono a conoscenza di tutti gli altri gestori code nel cluster.
- I gestori code in un cluster che ospitano una o più sottoscrizioni a un argomento cluster, creano automaticamente i canali mittente del cluster per tutti gli altri gestori code nel cluster, anche quando tali gestori code non pubblicano messaggi su argomenti cluster.
- La prima sottoscrizione su un gestore code a una stringa di argomenti in un argomento cluster comporta l'invio di un messaggio a tutti gli altri gestori code nel cluster. Allo stesso modo, l'ultima sottoscrizione su una stringa di argomenti da eliminare risulta anche in un messaggio. Maggiore è il numero di stringhe

di argomenti individuali utilizzate in un argomento in cluster e maggiore è la frequenza di modifica delle sottoscrizioni, maggiore è la comunicazione tra gestori code.

- Ogni gestore code nel cluster conserva la conoscenza delle stringhe di argomenti sottoscritte di cui è informato, anche quando il gestore code non pubblica né sottoscrive tali argomenti.

Per questi motivi, tutti i gestori code in un cluster con un argomento di instradamento diretto definito subiranno un ulteriore sovraccarico. Maggiore è il numero di gestori code presenti nel cluster, maggiore è il sovraccarico. Allo stesso modo, maggiore è il numero di stringhe di argomenti sottoscritte e maggiore è il loro tasso di modifica, maggiore è il sovraccarico. Ciò può comportare un carico eccessivo sui gestori code in esecuzione su sistemi di piccole dimensioni in un cluster di pubblicazione / sottoscrizione instradato direttamente, di grandi dimensioni o dinamico. Per ulteriori informazioni, vedi Prestazioni di pubblicazione / sottoscrizione instradate in modo diretto .

Quando si sa che un cluster non è in grado di gestire i sovraccarichi della pubblicazione / sottoscrizione del cluster instradata diretta, è possibile utilizzare invece la pubblicazione / sottoscrizione instradata dell'host argomento. In alternativa, in situazioni estreme, è possibile disabilitare completamente la funzionalità di pubblicazione / sottoscrizione del cluster impostando l'attributo del gestore code **PSCLUS** su DISABLED su ogni gestore code nel cluster. Consultare “Blocco della pubblicazione / sottoscrizione in cluster” a pagina 103. Ciò impedisce la creazione di qualsiasi argomento in cluster e garantisce che la rete non subisca alcun sovraccarico associato alla pubblicazione / sottoscrizione in cluster.

Cluster di pubblicazione / sottoscrizione instradati host argomento

Con l'instradamento dell'host argomento, i gestori code in cui sono definiti amministrativamente gli argomenti del cluster diventano router per le pubblicazioni. Le pubblicazioni dei gestori code non host nel cluster vengono instradate tramite il gestore code host a qualsiasi gestore code nel cluster con una sottoscrizione corrispondente.

Un cluster di pubblicazione / sottoscrizione instradato dall'host dell'argomento fornisce i seguenti vantaggi aggiuntivi su un cluster di pubblicazione / sottoscrizione instradato direttamente:

- Solo i gestori code su cui sono definiti gli argomenti instradati dell'host argomento vengono informati di tutti gli altri gestori code nel cluster.
- Solo i gestori code dell'host argomento devono essere in grado di connettersi a tutti gli altri gestori code nel cluster e generalmente si connetteranno solo a quelli in cui esistono le sottoscrizioni. Pertanto, vi è un numero significativamente inferiore di canali in esecuzione tra i gestori code.
- I gestori code del cluster che ospitano una o più sottoscrizioni a un argomento in cluster creano automaticamente i canali mittente del cluster solo per i gestori code che ospitano un argomento del cluster associato alla stringa dell'argomento della sottoscrizione.
- La prima sottoscrizione su un gestore code a una stringa di argomenti in un argomento in cluster determina l'invio di un messaggio a un gestore code nel cluster che ospita l'argomento in cluster. Allo stesso modo, l'ultima sottoscrizione su una stringa di argomenti da eliminare risulta anche in un messaggio. Maggiore è il numero di stringhe di argomenti individuali utilizzate in un argomento in cluster e maggiore è la frequenza di modifica delle sottoscrizioni, maggiore è la comunicazione tra gestori code, ma solo tra host di sottoscrizioni e host di argomenti.
- Più controllo sulla configurazione fisica. Con l'instradamento diretto, tutti i gestori code devono partecipare al cluster di pubblicazione / sottoscrizione, aumentando i propri costi generali. Con l'instradamento dell'host argomento, solo i gestori code dell'host argomento sono a conoscenza di altri gestori code e delle relative sottoscrizioni. È possibile scegliere esplicitamente i gestori code dell'host argomento, quindi è possibile verificare che tali gestori code siano in esecuzione su un'apparecchiatura adeguata ed è possibile utilizzare sistemi meno potenti per gli altri gestori code.

Elementi da considerare quando si utilizza un cluster di pubblicazione / sottoscrizione instradato dell'host argomento:

- Un ulteriore "hop" tra un gestore code di pubblicazione e un gestore code di sottoscrizione viene introdotto quando il publisher o il sottoscrittore non si trova su un gestore code che ospita l'argomento. La latenza causata dall' "hop" aggiuntivo può significare che l'instradamento dell'host argomento è meno efficace di quello diretto.

- Su cluster di grandi dimensioni, l'instradamento dell'host argomento facilita le prestazioni significative e i problemi di ridimensionamento che è possibile ottenere con l'instradamento diretto.
- È possibile definire tutti gli argomenti su un singolo gestore code o su un numero molto piccolo di gestori code. In questo caso, verificare che i gestori code dell'host argomento siano ospitati su sistemi potenti con una buona connettività.
- È possibile definire lo stesso argomento su più di un gestore code. Ciò migliora la disponibilità dell'argomento e migliora anche la scalabilità perché il carico di lavoro IBM MQ bilancia le pubblicazioni per un argomento tra tutti gli host per tale argomento. Si noti, tuttavia, che la definizione dello stesso argomento su più di un gestore code perde l'ordine dei messaggi per tale argomento.
- Ospitando diversi argomenti su gestori code differenti, è possibile migliorare la scalabilità senza perdere l'ordine dei messaggi.

Attività correlate

Scenario: creazione di un cluster di pubblicazione / sottoscrizione

Configurazione di un cluster di pubblicazione / sottoscrizione

Ottimizzazione delle reti di pubblicazione / sottoscrizione distribuite

Risoluzione dei problemi di pubblicazione / sottoscrizione distribuiti

Instradamento diretto nei cluster di pubblicazione / sottoscrizione

Le pubblicazioni da qualsiasi gestore code di pubblicazione vengono instradate direttamente a qualsiasi altro gestore code nel cluster con una sottoscrizione corrispondente.

Per un'introduzione al modo in cui i messaggi vengono instradati tra i gestori code nelle gerarchie di pubblicazione / sottoscrizione e nei cluster, consultare Reti di pubblicazione / sottoscrizione distribuite.

Un cluster di pubblicazione / sottoscrizione instradato si comporta come segue:

- Tutti i gestori code conoscono automaticamente tutti gli altri gestori code.
- Tutti i gestori code con sottoscrizioni agli argomenti del cluster creano canali per tutti i gestori code nel cluster e li informano delle relative sottoscrizioni.
- I messaggi pubblicati da un'applicazione vengono instradati dal gestore code a cui è connessa, direttamente a ciascun gestore code in cui esiste una sottoscrizione corrispondente.

Il seguente diagramma mostra un cluster del gestore code che non è attualmente utilizzato per attività di pubblicazione / sottoscrizione o point - to - point. Tenere presente che ogni gestore code nel cluster si connette solo ai gestori code del repository completo.

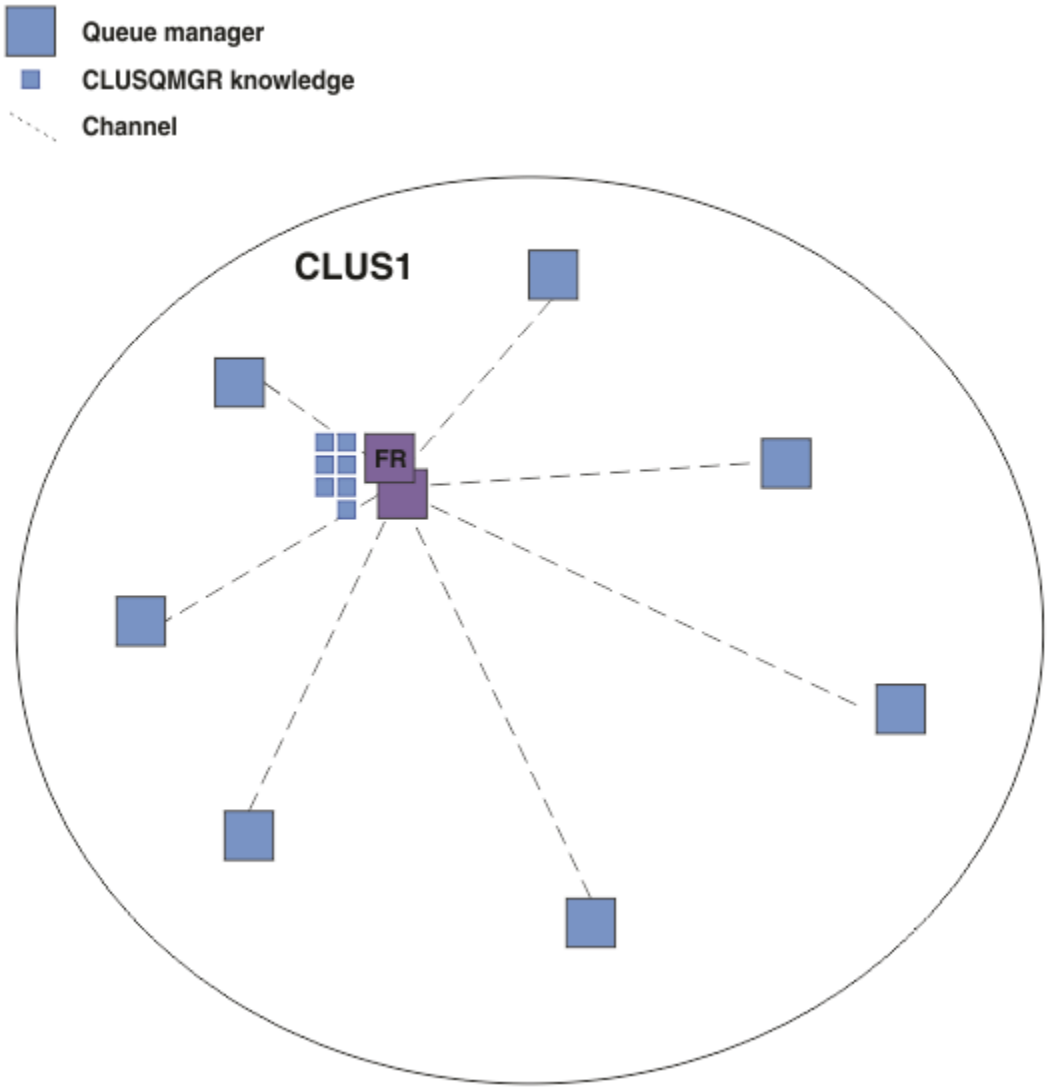


Figura 16. Un cluster di gestore code

Per consentire il flusso delle pubblicazioni tra gestori code in un cluster instradato direttamente, si crea il cluster di un ramo della struttura ad albero degli argomenti come descritto in Configurazione di un cluster di pubblicazione / sottoscrizionee si specifica *instradamento diretto* (impostazione predefinita).

In un cluster di pubblicazione / sottoscrizione instradato direttamente, si definisce l'oggetto argomento su qualsiasi gestore code nel cluster. Quando si esegue questa operazione, la conoscenza dell'oggetto e di tutti gli altri gestori code nel cluster viene automaticamente inviata a tutti i gestori code nel cluster dai gestori code del repository completo. Ciò si verifica prima che un gestore code faccia riferimento all'argomento:

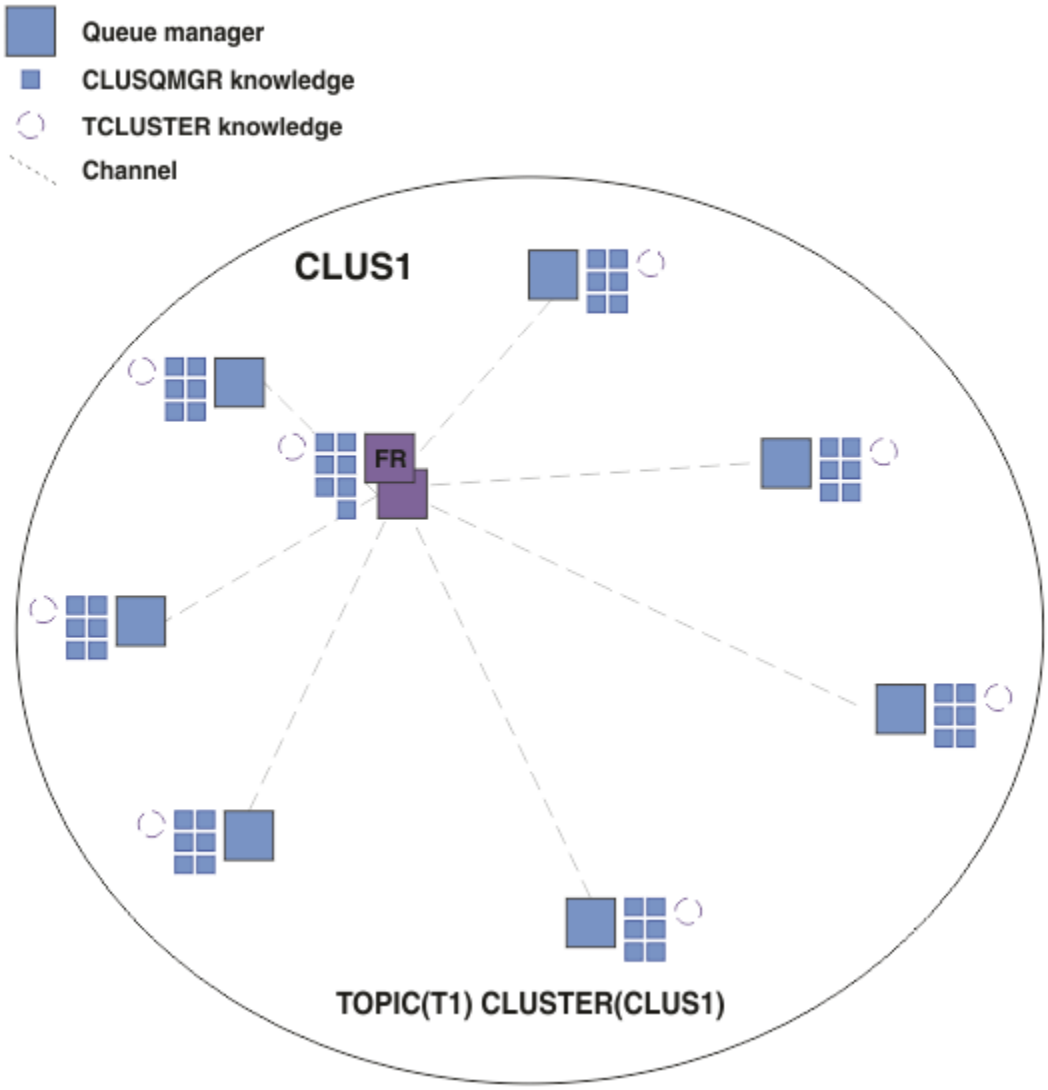


Figura 17. Un cluster di pubblicazione / sottoscrizione instradato direttamente

Quando viene creata una sottoscrizione, il gestore code che ospita la sottoscrizione stabilisce un canale per ogni gestore code nel cluster e invia i dettagli della sottoscrizione. Questa conoscenza della sottoscrizione distribuita è rappresentata da una sottoscrizione proxy su ciascun gestore code. Quando una pubblicazione viene prodotta su un qualsiasi gestore code nel cluster che corrisponde alla stringa di argomenti della sottoscrizione proxy, viene stabilito un canale cluster dal gestore code del publisher a ciascun gestore code che ospita una sottoscrizione e il messaggio viene inviato a ognuno di essi.

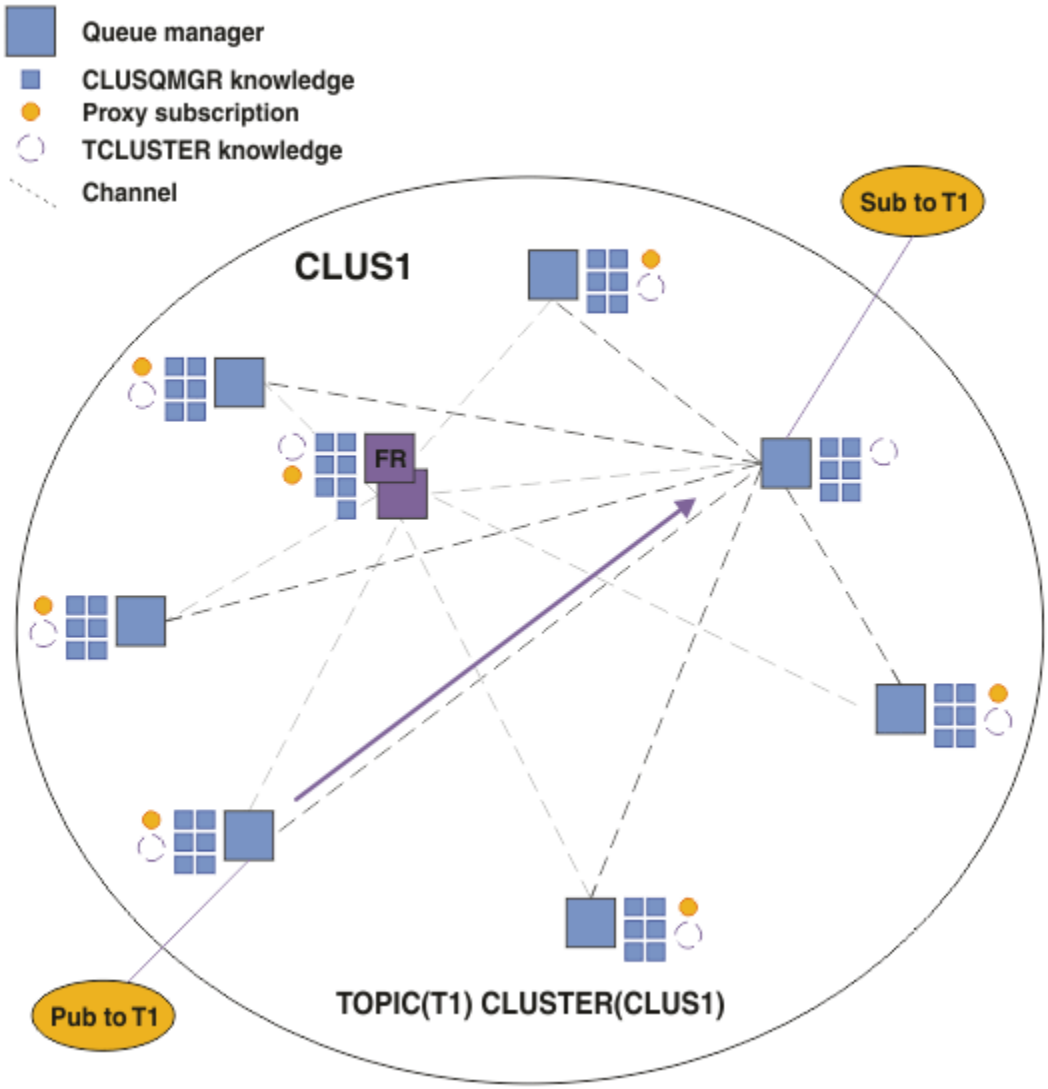


Figura 18. Un cluster di pubblicazione / sottoscrizione instradato direttamente con un publisher e un sottoscrittore a un argomento in cluster

L'instradamento diretto delle pubblicazioni ai gestori code che ospitano le sottoscrizioni semplifica la configurazione e riduce la latenza nella consegna delle pubblicazioni alle sottoscrizioni.

Tuttavia, a seconda dell'ubicazione delle sottoscrizioni e dei publisher, il cluster può diventare rapidamente completamente interconnesso, con ogni gestore code che ha una connessione diretta a ogni altro gestore code. Ciò potrebbe essere o meno accettabile nel proprio ambiente. Allo stesso modo, se la serie di stringhe di argomenti sottoscritte viene modificata di frequente, anche il sovraccarico di propagazione di tali informazioni tra tutti i gestori code può diventare significativo. Tutti i gestori code in un cluster di pubblicazione / sottoscrizione instradato devono essere in grado di far fronte a questi costi generali.

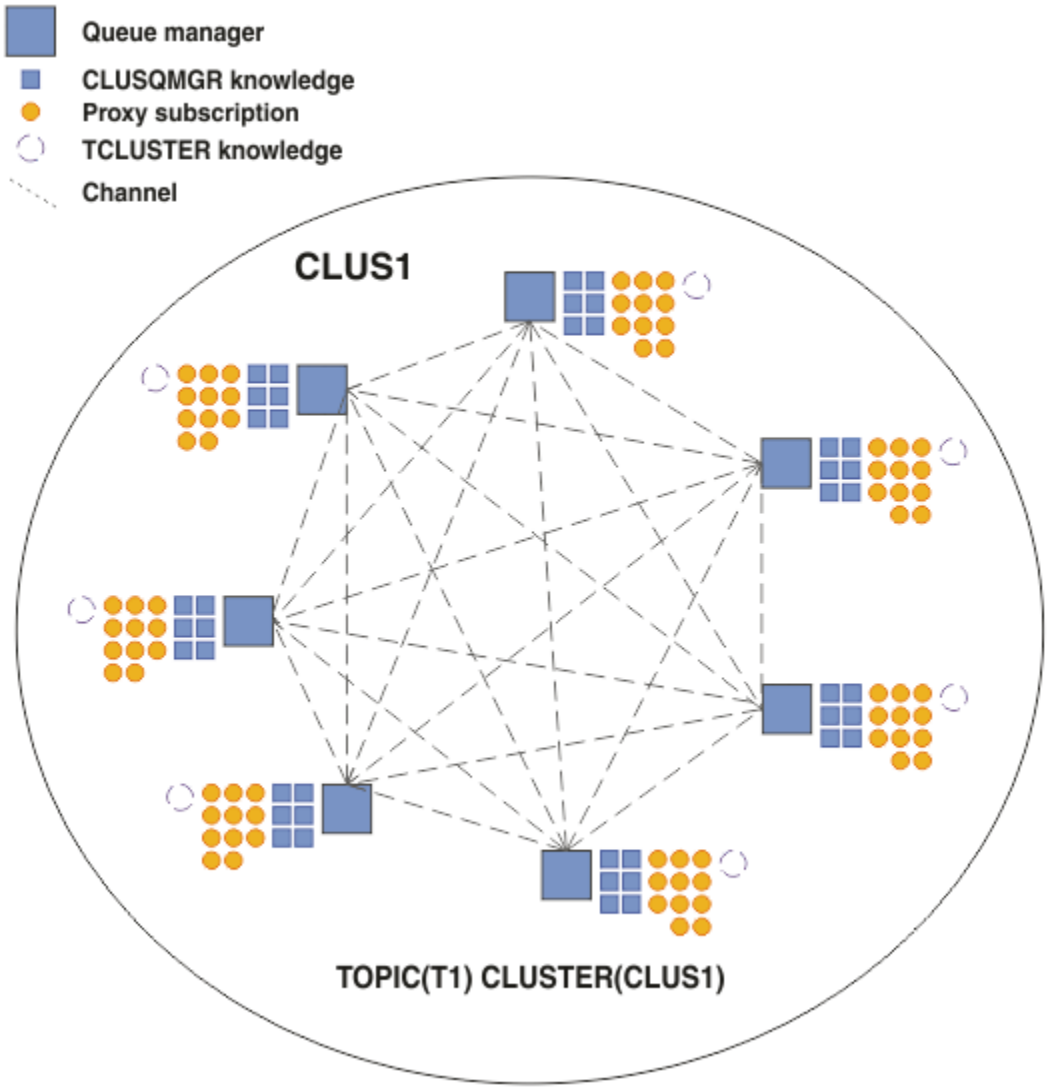


Figura 19. Un cluster di pubblicazione / sottoscrizione instradato che è completamente interconnesso

Sintesi e considerazioni aggiuntive

Un cluster di pubblicazione / sottoscrizione diretto ha bisogno di un piccolo intervento manuale per creare o amministrare e fornisce un instradamento diretto tra i publisher e i sottoscrittori. Per alcune configurazioni, di solito è la topologia più appropriata, in particolare i cluster con pochi gestori code o dove la connettività elevata dei gestori code è accettabile e le sottoscrizioni cambiano raramente. Tuttavia, impone anche alcuni vincoli al sistema:

- Il carico su ciascun gestore code è proporzionale al numero totale di gestori code nel cluster. Pertanto, in cluster più grandi, i singoli gestori code e il sistema nel suo complesso possono riscontrare problemi di prestazioni.
- Per impostazione predefinita, tutte le stringhe di argomenti in cluster sottoscritte vengono propagate in tutto il cluster e le pubblicazioni sono propagate solo ai gestori code remoti che hanno una sottoscrizione all'argomento associato. Pertanto, le rapide modifiche all'insieme delle sottoscrizioni possono diventare un fattore limitante. È possibile modificare questo comportamento predefinito e fare in modo che tutte le pubblicazioni vengano propagate a tutti i gestori code, il che elimina la necessità di sottoscrizioni proxy. 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.

  Nota: Una restrizione simile si applica anche alle gerarchie.

- A causa della natura interconnessa dei gestori code di pubblicazione / sottoscrizione, la propagazione delle sottoscrizioni proxy su tutti i nodi nella rete richiede tempo. Le pubblicazioni remote non iniziano necessariamente ad essere sottoscritte immediatamente, quindi le pubblicazioni iniziali potrebbero non essere inviate in seguito ad una sottoscrizione ad una nuova stringa di argomenti. È possibile rimuovere i problemi causati dal ritardo della sottoscrizione facendo in modo che tutte le pubblicazioni vengano propagate a tutti i gestori code, eliminando la necessità di sottoscrizioni proxy. Consultare Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione.

  Nota: Questa limitazione si applica anche alle gerarchie.

Prima di utilizzare l'instradamento diretto, esplorare gli approcci alternativi descritti in “Instradamento host argomento nei cluster di pubblicazione / sottoscrizione” a pagina 83e “Instradamento nelle gerarchie di pubblicazione / sottoscrizione” a pagina 108.

Instradamento host argomento nei cluster di pubblicazione / sottoscrizione

Le pubblicazioni dei gestori code non host nel cluster vengono instradate tramite il gestore code host a qualsiasi gestore code nel cluster con una sottoscrizione corrispondente.

Per un'introduzione al modo in cui i messaggi vengono instradati tra i gestori code nelle gerarchie di pubblicazione / sottoscrizione e nei cluster, consultare Reti di pubblicazione / sottoscrizione distribuite.

Per comprendere il comportamento e i vantaggi dell'instradamento dell'host argomento, è preferibile prima comprendere “Instradamento diretto nei cluster di pubblicazione / sottoscrizione” a pagina 78.

Un cluster di pubblicazione / sottoscrizione instradato dell'host argomento si comporta come segue:

- Gli oggetti argomento gestiti in cluster vengono definiti manualmente sui singoli gestori code nel cluster. A questi si fa riferimento come *gestori code host argomento*.
- Quando viene effettuata una sottoscrizione su un gestore code cluster, i canali vengono creati dal gestore code dell'host di sottoscrizione ai gestori code dell'host argomento e le sottoscrizioni proxy vengono create solo sui gestori code che ospitano l'argomento.
- Quando un'applicazione pubblica le informazioni su un argomento, il gestore code connesso inoltra sempre la pubblicazione a un gestore code su cui è presente l'argomento, che le trasmette a tutti i gestori code nel cluster che hanno sottoscrizioni corrispondenti all'argomento.

Questo processo è spiegato in modo più dettagliato nei seguenti esempi.

Instradamento host argomento utilizzando un singolo host argomento

Per il flusso delle pubblicazioni tra i gestori code in un cluster instradato di host argomento, si crea un cluster di una sezione della struttura ad albero degli argomenti come descritto in Configurazione di un cluster di pubblicazione / sottoscrizionee si specifica *Instradamento host argomento*.

Esistono diversi motivi per definire un oggetto argomento instradato dell'host argomento su più gestori code in un cluster. Tuttavia, per semplicità iniziamo con un singolo host di argomento.

Il seguente diagramma mostra un cluster del gestore code che non è attualmente utilizzato per attività di pubblicazione / sottoscrizione o point - to - point. Tenere presente che ogni gestore code nel cluster si connette solo ai gestori code del repository completo.

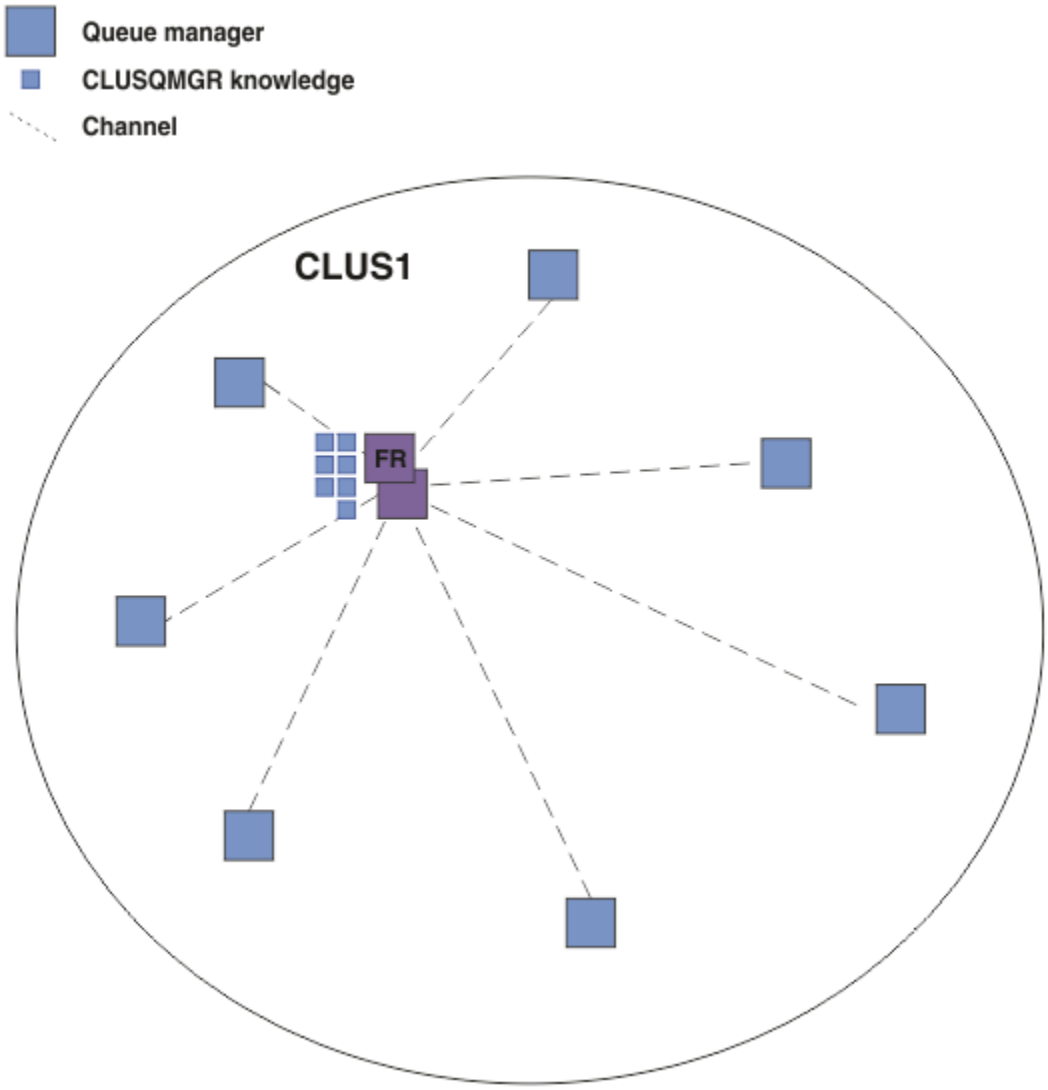


Figura 20. Un cluster di gestore code

In un cluster di pubblicazione / sottoscrizione instradato dell'host argomento, si definisce l'oggetto argomento su un gestore code specifico nel cluster. Il traffico di pubblicazione / sottoscrizione passa quindi attraverso tale gestore code, rendendolo un gestore code critico nel cluster e aumentandone il carico di lavoro. Per questi motivi si consiglia di non utilizzare un gestore code del repository completo, ma un altro gestore code nel cluster. Quando si definisce l'oggetto argomento sul gestore code host, la conoscenza dell'oggetto e del suo host viene automaticamente inviata, dai gestori code del repository completo, a tutti gli altri gestori code nel cluster. Notare che, a differenza dell' *instradamento diretto*, a ciascun gestore code non viene comunicato alcun altro gestore code nel cluster.

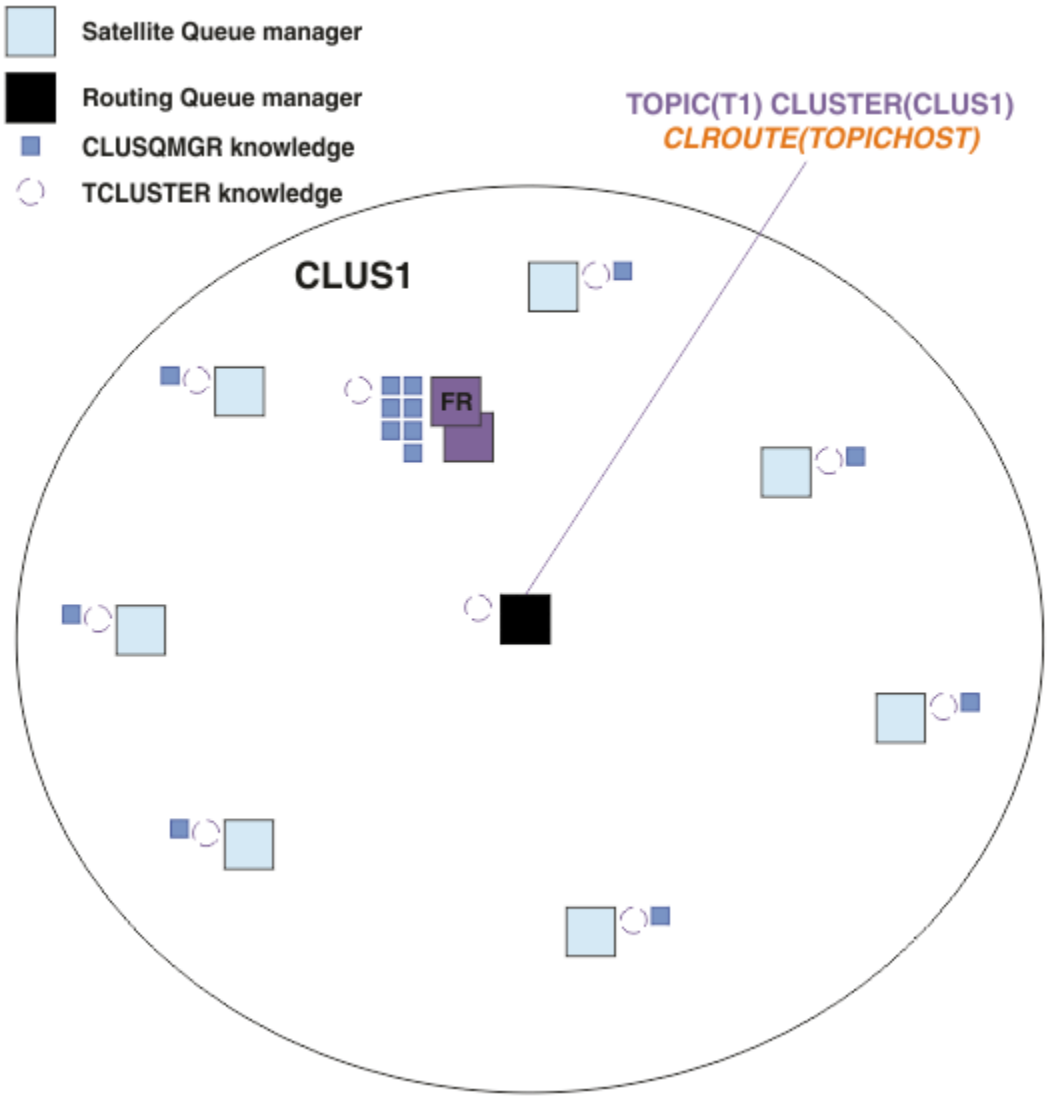


Figura 21. Un cluster di pubblicazione / sottoscrizione instradato dell'host argomento con un argomento definito su un host argomento

Quando una sottoscrizione viene creata su un gestore code, viene creato un canale tra il gestore code di sottoscrizione e il gestore code dell'host argomento. Il gestore code di sottoscrizione si connette solo al gestore code dell'host dell'argomento e invia i dettagli della sottoscrizione (sotto forma di *sottoscrizione proxy*). Il gestore code dell'host argomento non inoltra queste informazioni di sottoscrizione ad altri gestori code nel cluster.

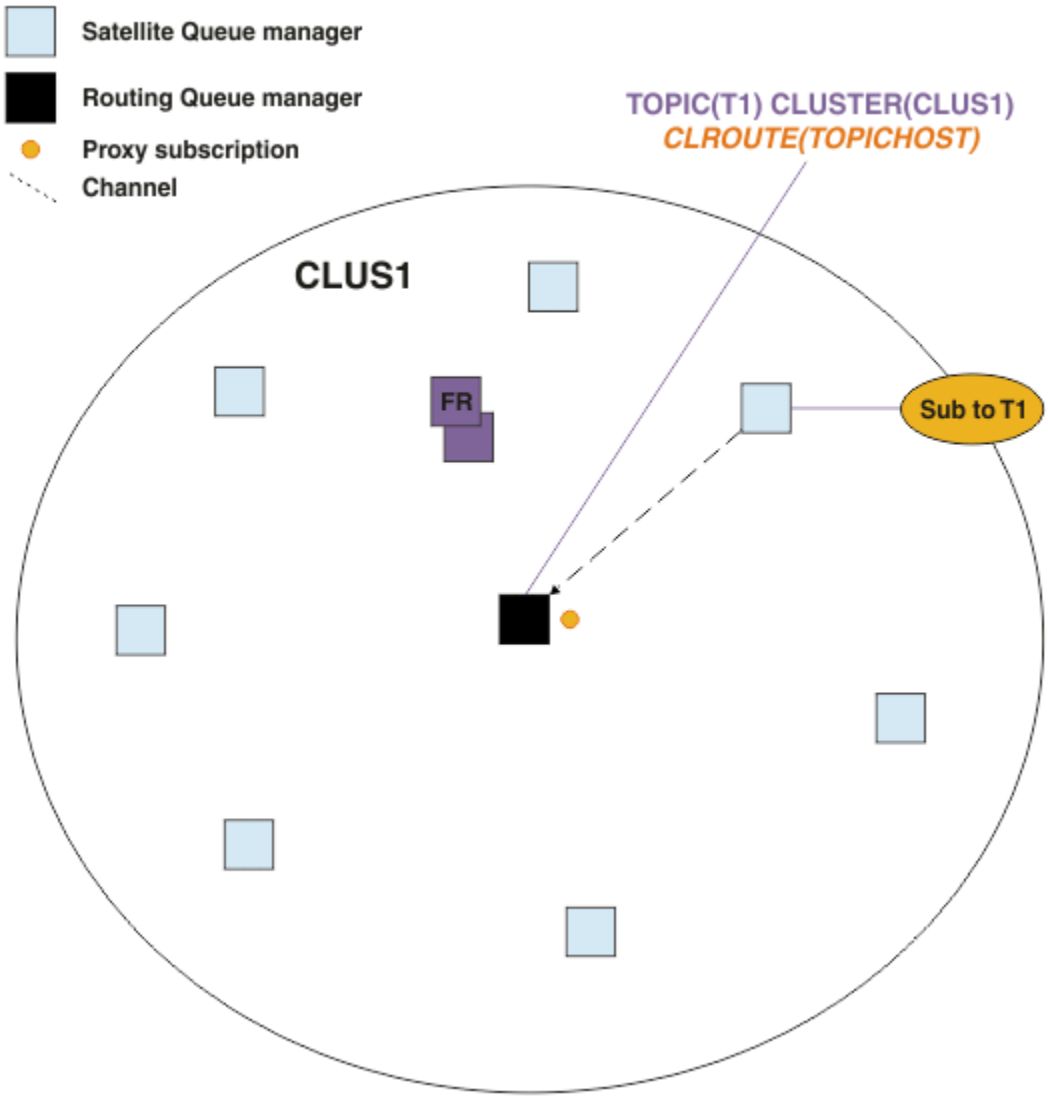


Figura 22. Un cluster di pubblicazione / sottoscrizione instradato dell'host argomento con un argomento definito su un host argomento e un sottoscrittore

Quando un'applicazione di pubblicazione si connette a un'altra gestore code e viene pubblicato un messaggio, viene creato un canale tra il gestore code di pubblicazione e il gestore code dell'host argomento e il messaggio viene inoltrato a tale gestore code. Il gestore code di pubblicazione non è a conoscenza di eventuali sottoscrizioni su altri gestori code nel cluster, quindi il messaggio viene inoltrato al gestore code dell'host argomento anche se non vi sono sottoscrittori a tale argomento nel cluster. Il gestore code di pubblicazione si collega solo al gestore code dell'host argomento. Le pubblicazioni vengono instradate tramite l'host argomento ai gestori code di sottoscrizione, se presenti.

Le sottoscrizioni sullo stesso gestore code del publisher vengono soddisfatte direttamente, senza inviare prima i messaggi a un gestore code dell'host argomento.

Tenere presente che, a causa del ruolo critico svolto da ciascun gestore code dell'host argomento, è necessario scegliere i gestori code che possono gestire i requisiti di carico, disponibilità e connettività dell'hosting dell'argomento.

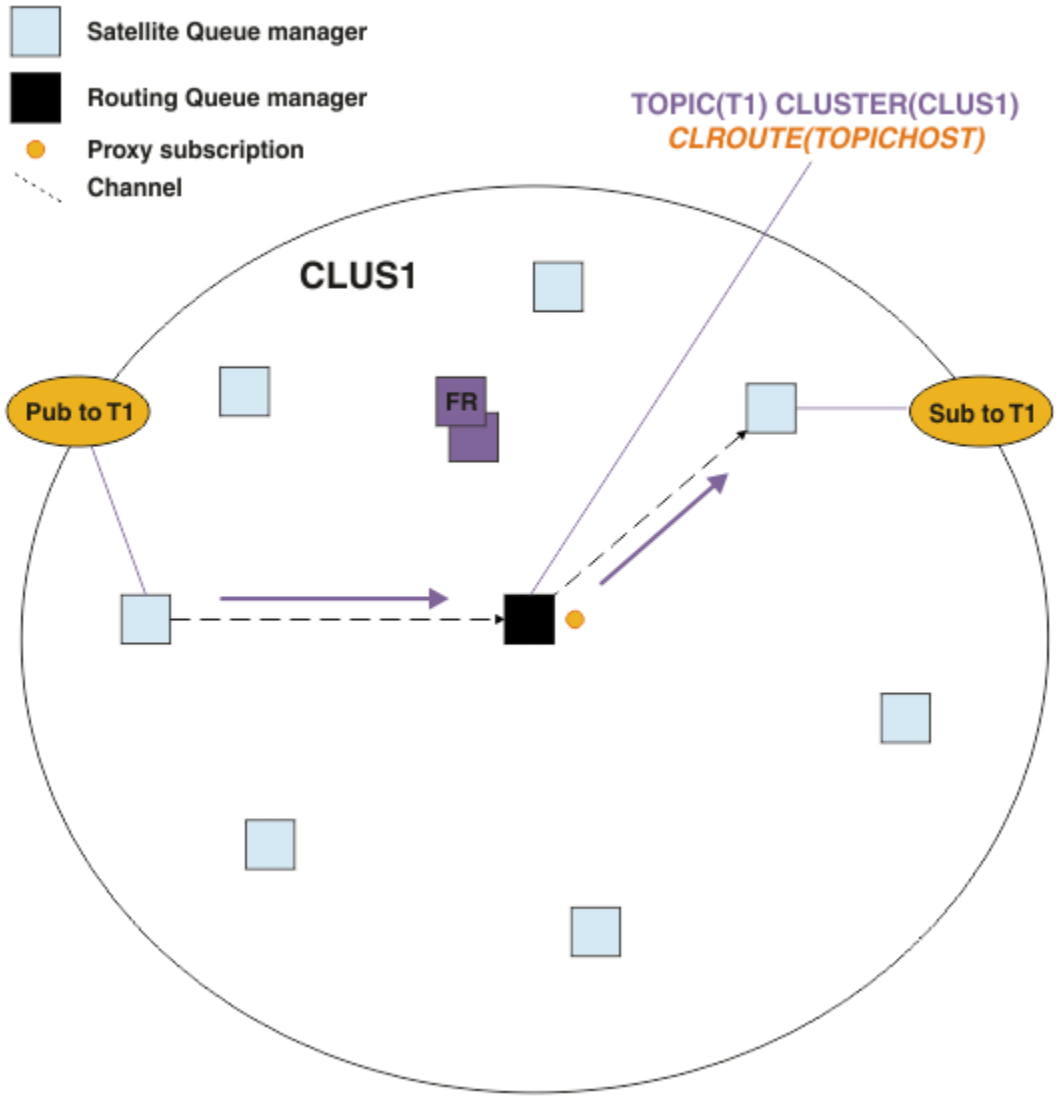


Figura 23. Un cluster di pubblicazione / sottoscrizione instradato dell'host argomento con un argomento, un sottoscrittore e un publisher

Divisione della struttura ad albero degli argomenti tra più gestori code

Un gestore code che ospita un argomento instradato è responsabile solo della conoscenza della sottoscrizione e dei messaggi di pubblicazione relativi al ramo della struttura ad albero dell'argomento per cui è configurato il suo oggetto argomento gestito. Se argomenti differenti vengono utilizzati da applicazioni di pubblicazione / sottoscrizione differenti nel cluster, è possibile configurare gestori code differenti per ospitare rami cluster differenti della struttura ad albero degli argomenti. Ciò consente la scalabilità riducendo il traffico di pubblicazione, la conoscenza della sottoscrizione e i canali su ciascun gestore code dell'host argomento nel cluster. Utilizzare questo metodo per i rami di volumi elevati distinti della struttura ad albero degli argomenti:

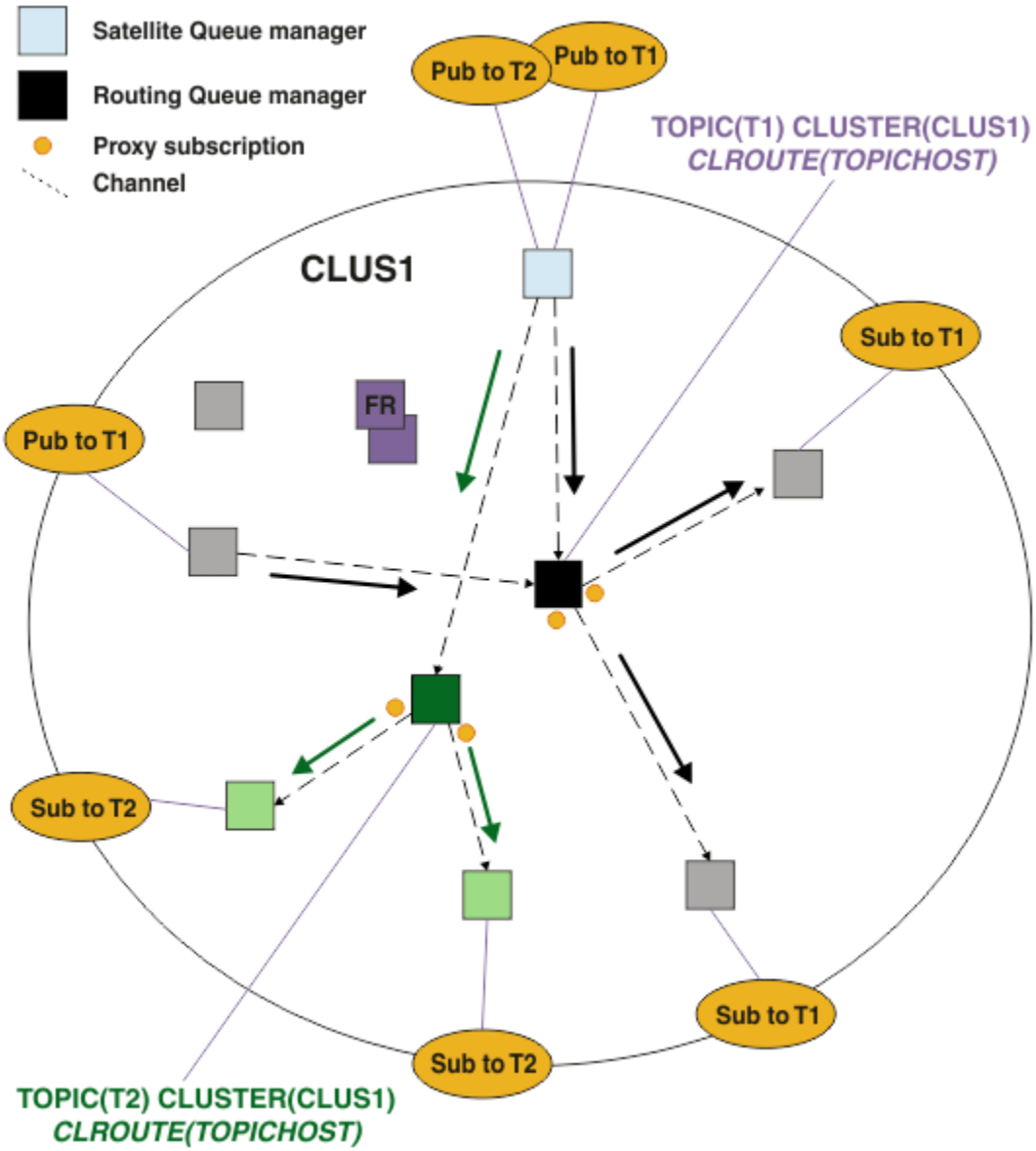


Figura 24. Un cluster di pubblicazione / sottoscrizione instradato dell'host argomento con due argomenti, ciascuno definito su un host argomento

Ad esempio, utilizzando gli argomenti descritti in Alberi argomenti, se l'argomento T1 è stato configurato con una stringa di argomenti `/USA/Alabamae` l'argomento T2 è stato configurato con una stringa di argomenti `/USA/Alaska`, un messaggio pubblicato in `/USA/Alabama/Mobile` verrà instradato tramite il gestore code che ospita T1, e un messaggio pubblicato in `/USA/Alaska/Juneau` verrà instradato tramite il gestore code su cui si trova T2.

Nota: Non è possibile eseguire una singola sottoscrizione su più rami cluster della struttura ad albero degli argomenti utilizzando un carattere jolly più in alto nella struttura ad albero degli argomenti rispetto ai punti raggruppati. Vedere Sottoscrizioni jolly.

Instradamento host argomento utilizzando più host argomento per un singolo argomento

Se un singolo gestore code ha la responsabilità dell'instradamento di un argomento e tale gestore code diventa non disponibile o non è in grado di gestire il carico di lavoro, le pubblicazioni non fluiranno prontamente nelle sottoscrizioni.

Se hai bisogno di maggiore resilienza, scalabilità e bilanciamento del carico di lavoro rispetto a quando definisci un argomento su un solo gestore code, puoi definire un argomento su più di un gestore code. Ogni singolo messaggio pubblicato viene instradato attraverso un unico host argomento. Quando esistono più definizioni host argomento corrispondenti, viene scelto uno degli host argomento. La scelta viene effettuata nello stesso modo delle code cluster. Ciò consente l'instradamento dei messaggi agli host argomento disponibili, evitando quelli non disponibili, e consente il bilanciamento del carico di lavoro del carico di lavoro tra più gestori code e canali dell'host argomento. Tuttavia, l'ordinamento tra più messaggi non viene mantenuto quando si utilizzano più host argomento per lo stesso argomento nel cluster.

Il seguente diagramma mostra un cluster di host argomento instradato in cui lo stesso argomento è stato definito su due gestori code. In questo esempio, i gestori code di sottoscrizione inviano informazioni sull'argomento sottoscritto a entrambi i gestori code dell'host argomento sotto forma di sottoscrizione proxy:

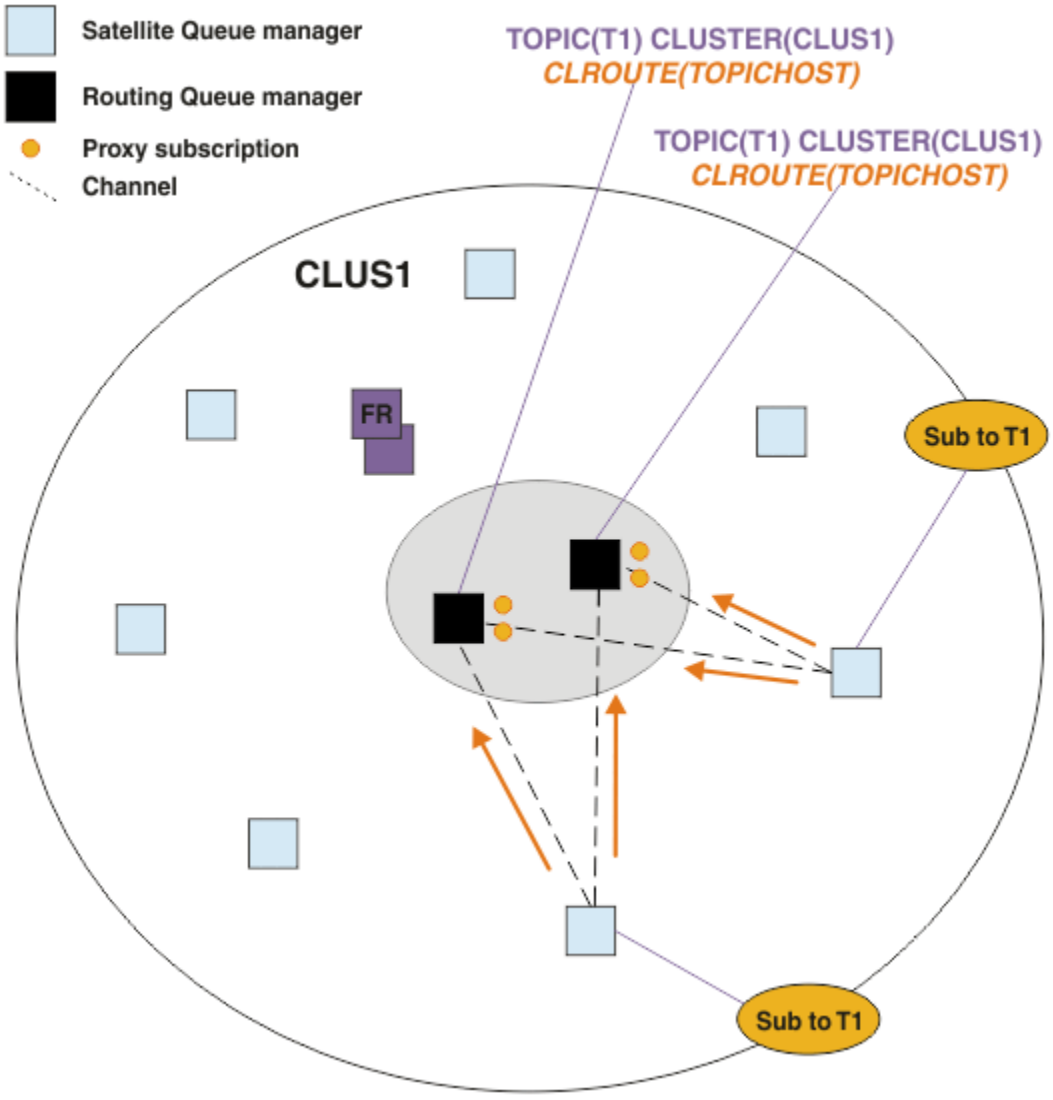


Figura 25. Creazione di sottoscrizioni proxy in un cluster di pubblicazione / sottoscrizione di più host argomento

Quando una pubblicazione viene effettuata da un gestore code non host, il gestore code invia una copia della pubblicazione a *uno* dei gestori code dell'host argomento per tale argomento. Il sistema sceglie l'host in base al comportamento predefinito dell' algoritmo di gestione del carico di lavoro del cluster. In un sistema tipico, questo valore si avvicina a una distribuzione round robin su ciascun gestore code dell'host argomento. Non esiste alcuna affinità tra i messaggi dalla stessa applicazione di pubblicazione; ciò equivale all'utilizzo di un bind cluster di tipo `NOTFIXED`.

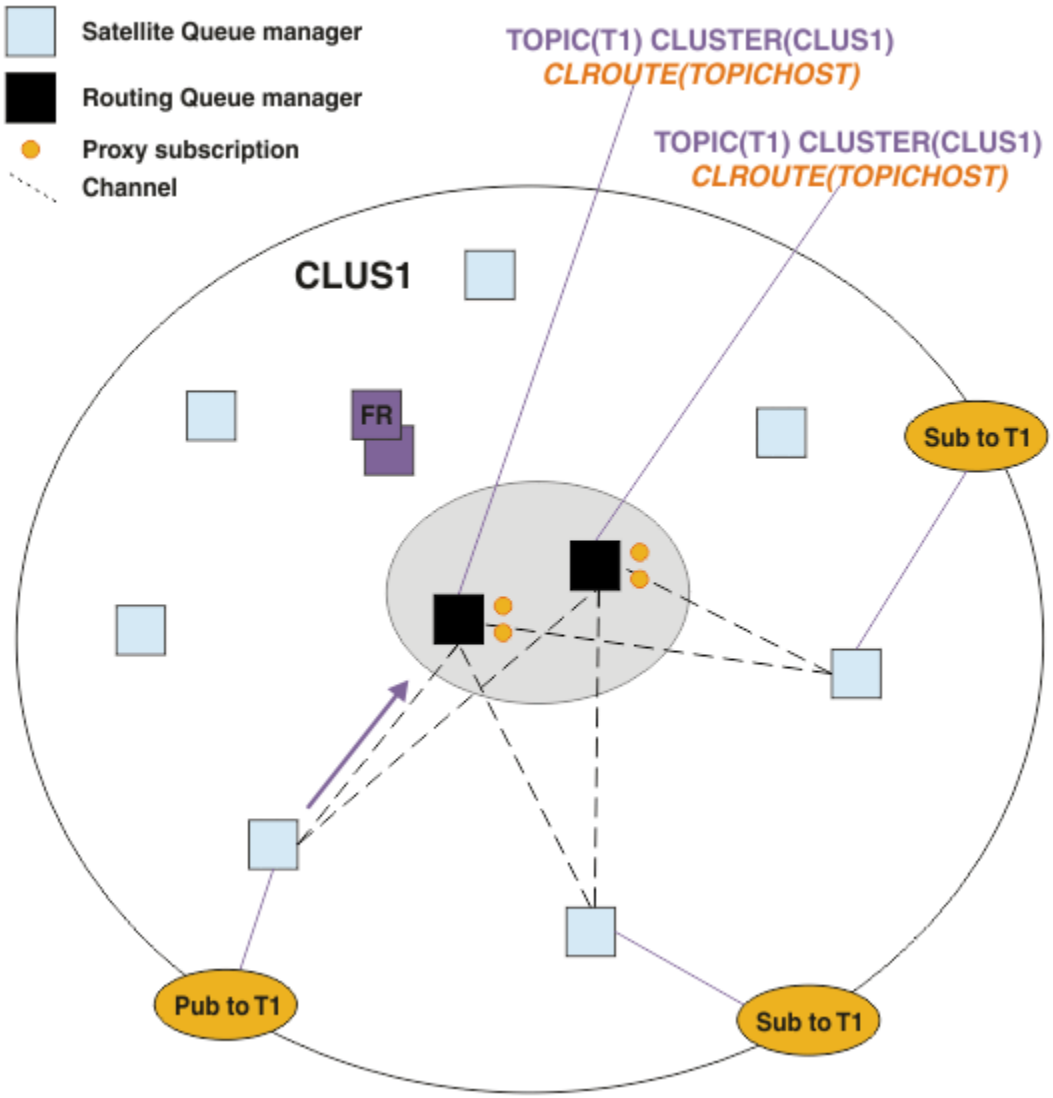


Figura 26. Ricezione di pubblicazioni in un cluster di pubblicazione / sottoscrizione di più host argomento

Le pubblicazioni in entrata per il gestore code dell'host argomento scelto vengono quindi inoltrate a tutti i gestori code che hanno registrato una sottoscrizione proxy corrispondente:

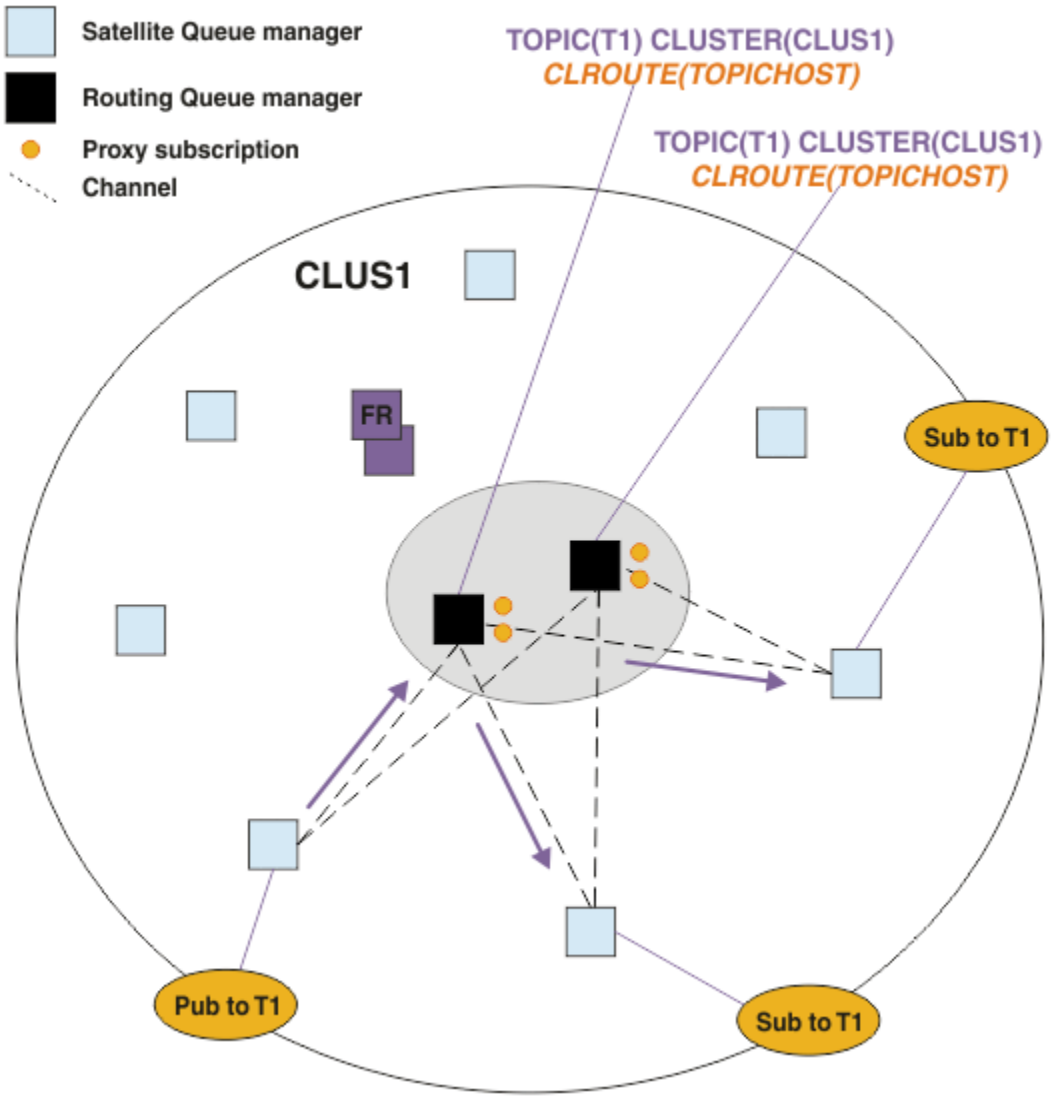


Figura 27. Instradamento delle pubblicazioni ai sottoscrittori in un cluster di pubblicazione / sottoscrizione di più host argomento

Come rendere le sottoscrizioni e i publisher locali per un gestore code dell'host argomento

Gli esempi precedenti mostrano l'instradamento tra i publisher e i sottoscrittori sui gestori code che non ospitano gli oggetti argomento instradati gestiti. In queste topologie, i messaggi richiedono più *hop* per raggiungere le sottoscrizioni.

Laddove l'hop aggiuntivo non è desiderato, potrebbe essere appropriato connettere i publisher di chiavi ai gestori code che ospitano gli argomenti. Tuttavia, se sono presenti più host argomento per un argomento e un solo publisher, tutto il traffico di pubblicazione verrà instradato attraverso il gestore code dell'host argomento a cui è connesso il publisher.

Allo stesso modo, se sono presenti sottoscrizioni chiave, queste potrebbero essere ubicate su un gestore code dell'host argomento. Tuttavia, se vi sono più host dell'argomento instradato, solo una parte delle pubblicazioni eviterà l'hop aggiuntivo, mentre il resto verrà instradato prima attraverso gli altri gestori code dell'host argomento.

Topologie come queste sono descritte più avanti: Instradamento host argomento utilizzando publisher o sottoscrittori centralizzati.

Nota: Una pianificazione speciale è necessaria se si modifica la configurazione dell'argomento instradato quando si co - localizzano i publisher o le sottoscrizioni con gli host dell'argomento instradati. Ad esempio, consultare Aggiunta di ulteriori host argomento a un cluster instradato host argomento.

Sintesi e considerazioni aggiuntive

Un cluster di pubblicazione / sottoscrizione instradato dell'host argomento fornisce un controllo preciso su quali gestori code ospitano ciascun argomento e tali gestori code diventano i gestori code di *instradamento* per quel ramo della struttura ad albero degli argomenti. Inoltre, i gestori code senza sottoscrizioni o publisher non hanno bisogno di connettersi ai gestori code dell'host argomento e i gestori code con sottoscrizioni non hanno bisogno di connettersi ai gestori code che non ospitano un argomento. Questa configurazione può ridurre in modo significativo il numero di connessioni tra gestori code nel cluster e la quantità di informazioni trasmesse tra gestori code. Ciò è particolarmente vero nei cluster di grandi dimensioni in cui solo un sottoinsieme di gestori code sta eseguendo attività di pubblicazione / sottoscrizione. Questa configurazione fornisce anche un certo controllo sul carico sui singoli gestori code nel cluster, per cui (ad esempio) è possibile scegliere di ospitare argomenti molto attivi su sistemi più potenti e resilienti. Per alcune configurazioni, in particolare per i cluster più grandi, di solito si tratta di una topologia più appropriata rispetto all' *instradamento diretto*.

Tuttavia, l'instradamento all'host argomento impone anche alcuni vincoli sul sistema:

- La configurazione e manutenzione del sistema richiedono una maggiore pianificazione rispetto all'instradamento diretto. È necessario decidere quali punti raggruppare nella struttura ad albero degli argomenti e la posizione delle definizioni di argomento nel cluster.
- Così come per gli argomenti con instradamento diretto, quando si definisce un nuovo argomento instradato all'host argomento, le informazioni vengono inviate ai gestori code del repository completo e da lì indirizzate a tutti i membri del cluster. Questo evento fa sì che i canali vengano avviati in ciascun membro del cluster da tutti i repository (se non già avviati).
- Le pubblicazioni vengono sempre inviate al gestore code host da un gestore code non host, anche se non sono presenti sottoscrizioni nel cluster. Pertanto, occorre utilizzare gli argomenti instradati quando è prevista l'esistenza di sottoscrizioni o quando il sovraccarico di connettività globale e conoscenza è maggiore del rischio di traffico di pubblicazione supplementare.

  Nota: Come descritto in precedenza, rendere i publisher locali per un host argomento può ridurre questo rischio.
- I messaggi che vengono pubblicati sui gestori code non host non vengono indirizzati direttamente al gestore code che ospita la sottoscrizione, ma vengono instradati sempre attraverso un gestore code dell'host argomento. Questo approccio può aumentare il sovraccarico totale nel cluster, nonché aumentare la latenza dei messaggi e ridurre le prestazioni.

  Nota: Come descritto in precedenza, rendere le sottoscrizioni o i publisher locali per un host argomento può ridurre questo rischio.
- L'utilizzo di un unico gestore code dell'host argomento introduce un singolo punto di errore per tutti i messaggi che vengono pubblicati in un argomento. È possibile rimuovere questo singolo punto di errore definendo più host argomento. Tuttavia, la presenza di più host influisce sull'ordine dei messaggi pubblicati man mano che vengono ricevuti dalle sottoscrizioni.
- Il carico di messaggi supplementare è sostenuto dai gestori code dell'host argomento, in quanto questi dovranno elaborare il traffico di pubblicazione da più gestori code. Questo carico può essere ridotto: utilizzare più host argomento per un singolo argomento (nel qual caso l'ordine dei messaggi non viene mantenuto) o utilizzare gestori code differenti per ospitare gli argomenti instradati per i diversi rami di una struttura ad albero degli argomenti.

Prima di utilizzare l'instradamento dell'host argomento, esplorare gli approcci alternativi descritti in “Instradamento diretto nei cluster di pubblicazione / sottoscrizione” a pagina 78e “Instradamento nelle gerarchie di pubblicazione / sottoscrizione” a pagina 108.

Clustering di pubblicazione / sottoscrizione: procedure ottimali

L'utilizzo di argomenti raggruppati rende semplice l'estensione del dominio di pubblicazione / sottoscrizione tra gestori code, ma può portare a problemi se i meccanismi e le implicazioni non sono completamente compresi. Esistono due modelli per la condivisione delle informazioni e l'instradamento delle pubblicazioni. Implementare il modello che meglio soddisfa le singole esigenze di business e si comporta al meglio sul cluster scelto.

Le informazioni sulle migliori pratiche riportate nelle seguenti sezioni non forniscono una soluzione unica per tutte le soluzioni, ma condividono piuttosto approcci comuni per risolvere problemi comuni. Si presume che si abbia una conoscenza di base dei cluster IBM MQ e della messaggistica di pubblicazione / sottoscrizione e che si abbia familiarità con le informazioni nelle reti di pubblicazione / sottoscrizione distribuite e “Progettazione di cluster di pubblicazione / sottoscrizione” a pagina 76.

Quando si utilizza un cluster per la messaggistica point - to - point, ogni gestore code nel cluster funziona in base alla necessità di sapere. In altre parole, rileva solo altre risorse del cluster, ad esempio altri gestori code nel cluster e code cluster, quando le applicazioni che si collegano ad essi richiedono di utilizzarle. Quando si aggiunge la messaggistica di pubblicazione / sottoscrizione a un cluster, viene introdotto un livello maggiore di condivisione delle informazioni e di connettività tra i gestori code del cluster. Per poter seguire le procedure ottimali per i cluster di pubblicazione / sottoscrizione, è necessario comprendere appieno le implicazioni di questa modifica del comportamento.

Per consentirti di creare l'architettura migliore, in base alle tue precise esigenze, ci sono due modelli per la condivisione delle informazioni e l'instradamento della pubblicazione nei cluster di pubblicazione / sottoscrizione: *instradamento diretto* e *instradamento dell'host argomento*. Per fare la giusta scelta, è necessario comprendere entrambi i modelli e i diversi requisiti che ogni modello soddisfa. Questi requisiti sono discussi nelle sezioni seguenti, insieme a “Pianificazione della rete di pubblicazione / sottoscrizione distribuita” a pagina 72:

- “Motivi per limitare il numero di gestori code del cluster coinvolti nell'attività di pubblicazione / sottoscrizione” a pagina 93
- “Come decidere quali argomenti raggruppare” a pagina 94
- “Come dimensionare il sistema” a pagina 94
- “Ubicazione di pubblicazione e sottoscrizione” a pagina 95
- “Traffico di pubblicazione” a pagina 96
- “Modifica della sottoscrizione e stringhe di argomenti dinamici” a pagina 96

Motivi per limitare il numero di gestori code del cluster coinvolti nell'attività di pubblicazione / sottoscrizione

Esistono considerazioni sulla capacità e sulle prestazioni quando si utilizza la messaggistica di pubblicazione / sottoscrizione in un cluster. Di conseguenza, è consigliabile considerare attentamente la necessità di attività di pubblicazione / sottoscrizione tra i gestori code e limitarla solo al numero di gestori code che lo richiedono. Una volta identificata la serie minima di gestori code che devono pubblicare e sottoscrivere gli argomenti, è possibile renderli membri di un cluster che contiene solo loro e nessun altro gestore code.

Questo approccio è particolarmente utile se si dispone di un cluster stabilito che funziona già bene per la messaggistica point - to - point. Quando si trasforma un cluster di grandi dimensioni esistente in un cluster di pubblicazione / sottoscrizione, si consiglia di creare inizialmente un cluster separato per il lavoro di pubblicazione / sottoscrizione in cui è possibile provare le applicazioni, piuttosto che utilizzare il cluster corrente. È possibile utilizzare un sottoinsieme di gestori code esistenti che si trovano già in uno o più cluster point - to - point e rendere questo sottoinsieme membro del nuovo cluster di pubblicazione / sottoscrizione. Tuttavia, i gestori code del repository completo per il nuovo cluster non devono essere membri di nessun altro cluster; ciò isola il carico aggiuntivo dai repository completi del cluster esistenti.

Se non è possibile creare un nuovo cluster e si deve trasformare un cluster di grandi dimensioni esistente in un cluster di pubblicazione / sottoscrizione, non utilizzare un modello instradato direttamente. Il modello di host argomento instradato di solito funziona meglio in cluster più grandi, perché in genere

limita la condivisione delle informazioni di pubblicazione / sottoscrizione e la connettività alla serie di gestori code che stanno eseguendo attivamente il lavoro di pubblicazione / sottoscrizione, concentrandosi sui gestori code che ospitano gli argomenti. L'eccezione è rappresentata dal fatto che un aggiornamento manuale delle informazioni di sottoscrizione viene richiamato su un gestore code che ospita una definizione di argomento, a quel punto il gestore code dell'host argomento si connetterà a ogni gestore code nel cluster. Consultare Risincronizzazione delle sottoscrizioni proxy.

Se si stabilisce che un cluster non può essere utilizzato per la pubblicazione / sottoscrizione a causa della sua dimensione o del carico corrente, si consiglia di evitare che questo cluster venga inaspettatamente trasformato in un cluster di pubblicazione / sottoscrizione. Utilizzare la proprietà del gestore code **PSCLUS** per arrestare gli utenti che aggiungono un argomento in cluster su qualsiasi gestore code nel cluster. Consultare “Blocco della pubblicazione / sottoscrizione in cluster” a pagina 103.

Come decidere quali argomenti raggruppare

È importante scegliere attentamente quali argomenti vengono aggiunti al cluster: più in alto nella struttura ad albero degli argomenti si trovano questi argomenti, più diventa diffuso il loro utilizzo. Ciò può causare la propagazione di un numero maggiore di informazioni di sottoscrizione e di pubblicazioni rispetto al necessario. Se ci sono più rami distinti della struttura ad albero degli argomenti, dove alcuni devono essere raggruppati in cluster e altri no, creare oggetti argomento gestiti alla radice di ogni ramo che necessita di cluster e aggiungerli al cluster. Ad esempio, se i rami `/A`, `/B` e `/C` necessitano di cluster, definire un oggetto argomento cluster separato per ogni ramo.

Nota: Il sistema impedisce di nidificare definizioni di argomenti in cluster nella struttura ad albero degli argomenti. È possibile raggruppare gli argomenti solo in un punto della struttura ad albero degli argomenti per ogni ramo secondario. Ad esempio, non è possibile definire oggetti argomento in cluster per `/A` e per `/A/B`. La nidificazione degli argomenti del cluster può creare confusione su quale oggetto del cluster si applica a quale sottoscrizione, specialmente quando le sottoscrizioni utilizzano caratteri jolly. Ciò è ancora più importante quando si utilizza l'instradamento dell'host argomento, dove le decisioni di instradamento sono definite in modo preciso dall'assegnazione degli host argomento.

Se gli argomenti in cluster devono essere aggiunti in alto nella struttura ad albero degli argomenti, ma alcuni rami della struttura ad albero al di sotto del punto in cluster non richiedono il funzionamento in cluster, è possibile utilizzare gli attributi dell'ambito della sottoscrizione e della pubblicazione per ridurre il livello di condivisione della sottoscrizione e della pubblicazione per ulteriori argomenti.

Non inserire il nodo root dell'argomento nel cluster senza considerare il comportamento visualizzato. Rendere gli argomenti globali ovvi laddove possibile, ad esempio utilizzando un qualificatore di alto livello nella stringa di argomenti: `/global` o `/cluster`.

Vi è un ulteriore motivo per non voler creare il nodo dell'argomento root in cluster. Ciò è dovuto al fatto che ogni gestore code dispone di una definizione locale per il nodo root, l'oggetto argomento `SYSTEM.BASE.TOPIC` . Quando questo oggetto viene raggruppato in cluster su un gestore code nel cluster, tutti gli altri gestori code ne vengono informati. Tuttavia, quando esiste una definizione locale dello stesso oggetto, le relative proprietà sovrascrivono l'oggetto cluster. Ciò si traduce in quei gestori code che agiscono come se l'argomento non fosse raggruppato in cluster. Per risolvere questo problema, è necessario raggruppare ogni definizione di `SYSTEM.BASE.TOPIC`. È possibile eseguire questa operazione per le definizioni di instradamento diretto, ma non per le definizioni di instradamento dell'host argomento, poiché ogni gestore code diventa un host argomento.

Come dimensionare il sistema

I cluster di pubblicazione / sottoscrizione generalmente risultano in un modello diverso di canali cluster per la messaggistica point - to - point in un cluster. Il modello point - to - point è un modello 'opt in', ma i cluster di pubblicazione / sottoscrizione hanno una natura più indiscriminata con fan - out di sottoscrizione, specialmente quando si utilizzano argomenti instradati direttamente. Pertanto, è importante identificare quali gestori code in un cluster di pubblicazione / sottoscrizione utilizzeranno i canali cluster per connettersi ad altri gestori code e in quali circostanze.

La seguente tabella elenca la serie tipica di canali mittente e ricevente del cluster prevista per ogni gestore code in un cluster di pubblicazione / sottoscrizione in esecuzione normale, in base al ruolo del gestore code nel cluster di pubblicazione / sottoscrizione.

Tabella 5. Canali mittente e destinatario cluster per ogni metodo di instradamento.

Ruolo gestore code	Ricevitori cluster diretti	Mittenti cluster diretti	Ricevitori cluster argomento	Mittenti cluster argomento
Repository completo	AllQmgrs	AllQmgrs	AllQmgrs	AllQMgrs
Host della definizione argomento	n/a	n/a	AllSubs+AllPubs (1)	AllSubs (1)
Sottoscrizioni create	AllPubs (1)	AllQMgrs	AllHosts	AllHosts
Publisher connessi	AllSubs (1)	AllSubs (1)	AllHosts	AllHosts
Nessun publisher o sottoscrittore	AllSubs (1)	Nessuno (1)	Nessuno (2)	Nessuno (2)

Chiave:

AllQmgrs
Un canale da e verso ogni gestore code nel cluster.

AllSubs
Un canale da e verso ogni gestore code in cui è stata creata una sottoscrizione.

AllPubs
Un canale da e verso ogni gestore code a cui è stata connessa un'applicazione di pubblicazione.

AllHosts
Un canale da e verso ogni gestore code in cui è stata configurata una definizione dell'oggetto argomento con cluster.

Nessuna
Nessun canale verso o da altri gestori code nel cluster per il solo scopo della messaggistica di pubblicazione / sottoscrizione.

Note:

1. Se un aggiornamento del gestore code delle sottoscrizioni proxy viene effettuato da questo gestore code, un canale da e verso tutti gli altri gestori code nel cluster potrebbe essere creato automaticamente.
2. Se un aggiornamento del gestore code delle sottoscrizioni proxy viene effettuato da questo gestore code, un canale verso e da qualsiasi altro gestore code nel cluster che ospita una definizione di un argomento del cluster potrebbe essere creato automaticamente.

La tabella precedente mostra che l'instradamento host argomento in genere utilizza meno canali mittente e ricevente del cluster rispetto all'instradamento diretto. Se la connettività del canale è un problema per alcuni gestori code in un cluster, per motivi di capacità o capacità di stabilire determinati canali (ad esempio, tramite i firewall), l'instradamento dell'host argomento è pertanto una soluzione preferita.

Ubicazione di pubblicazione e sottoscrizione

La pubblicazione / sottoscrizione in cluster consente ai messaggi pubblicati su un gestore code di essere consegnati alle sottoscrizioni su qualsiasi altro gestore code nel cluster. Per quanto riguarda la messaggistica point - to - point, il costo di trasmissione dei messaggi tra gestori code può essere dannoso per le prestazioni. Pertanto, è necessario considerare la possibilità di creare sottoscrizioni agli argomenti sugli stessi gestori code in cui vengono pubblicati i messaggi.

Quando si utilizza l'instradamento dell'host argomento in un cluster, è importante considerare anche l'ubicazione delle sottoscrizioni e dei publisher rispetto ai gestori code che ospitano l'argomento. Quando

il publisher non è connesso a un gestore code che è un host dell'argomento cluster, i messaggi pubblicati vengono sempre inviati a un gestore code che ospita l'argomento. Allo stesso modo, quando una sottoscrizione viene creata su un gestore code che non è un host argomento per un argomento cluster, i messaggi pubblicati da altri gestori code nel cluster vengono sempre inviati prima a un gestore code che ospita un argomento. In modo più specifico, se la sottoscrizione si trova su un gestore code che ospita l'argomento, ma sono presenti uno o più altri gestori code che ospitano lo stesso argomento, una proporzione di pubblicazioni provenienti da altri gestori code viene instradata attraverso gli altri gestori code che ospitano l'argomento. Per ulteriori informazioni sulla progettazione di un cluster di pubblicazione / sottoscrizione instradato dell'host argomento per ridurre la distanza tra i publisher e le sottoscrizioni, consultare Instradamento dell'host argomento utilizzando i publisher o i sottoscrittori centralizzati .

Traffico di pubblicazione

I messaggi pubblicati da un'applicazione connessa a un gestore code in un cluster vengono trasmessi alle sottoscrizioni su altri gestori code utilizzando i canali mittente del cluster.

Quando si utilizza l'instradamento diretto, i messaggi pubblicati utilizzano il percorso più breve tra i gestori code. In altre parole, passano direttamente dal gestore code di pubblicazione a ciascuno dei gestori code con sottoscrizioni. I messaggi non vengono trasmessi ai gestori code che non hanno sottoscrizioni per l'argomento. Consultare Sottoscrizioni proxy in una rete di pubblicazione / sottoscrizione.

Quando la frequenza dei messaggi di pubblicazione tra un gestore code e un altro nel cluster è elevata, l'infrastruttura del canale cluster tra questi due punti deve essere in grado di mantenere la frequenza. Ciò potrebbe implicare l'ottimizzazione dei canali e della coda di trasmissione utilizzati.

Quando si utilizza l'instradamento dell'host argomento, ogni messaggio pubblicato su un gestore code che non è un host argomento viene trasmesso a un gestore code dell'host argomento. Ciò è indipendente dal fatto che una o più sottoscrizioni siano presenti in qualsiasi altro punto del cluster. Ciò introduce ulteriori fattori da considerare nella pianificazione:

- La latenza aggiuntiva dell'invio di ciascuna pubblicazione a un gestore code dell'host argomento è accettabile?
- Ciascun gestore code dell'host argomento può sostenere la frequenza di pubblicazione in entrata e in uscita? Considerare un sistema con publisher su molti gestori code differenti. Se tutti inviano i messaggi a una serie molto piccola di gestori code che ospitano argomenti, tali host argomento potrebbero diventare un collo di bottiglia nell'elaborazione di tali messaggi e nell'indirizzarli ai gestori code di sottoscrizione.
- Si prevede che una percentuale significativa dei messaggi pubblicati non avrà un sottoscrittore corrispondente? In tal caso, e la frequenza di pubblicazione di tali messaggi è elevata, potrebbe essere preferibile rendere il gestore code del publisher un host argomento. In questa situazione, qualsiasi messaggio pubblicato in cui non esistono sottoscrizioni nel cluster non verrà trasmesso ad altri gestori code.

Questi problemi potrebbero anche essere risolti introducendo più host di argomenti, per distribuire il carico di pubblicazione su di essi:

- In presenza di più argomenti distinti, ciascuno con una parte del traffico di pubblicazione, considerare la possibilità di ospitarli su gestori code differenti.
- Se gli argomenti non possono essere separati su host di argomenti differenti, considerare la possibilità di definire lo stesso oggetto argomento su più gestori code. Ciò si traduce in un bilanciamento del carico di lavoro delle pubblicazioni per l'instradamento. Tuttavia, ciò è appropriato solo quando non è richiesto l'ordinamento dei messaggi di pubblicazione.

Modifica della sottoscrizione e stringhe di argomenti dinamici

Un'altra considerazione è l'effetto sulle prestazioni del sistema per la propagazione delle sottoscrizioni proxy. In genere, un gestore code invia un messaggio di sottoscrizione proxy ad alcuni altri gestori code nel cluster quando viene creata la prima sottoscrizione per una specifica stringa di argomenti in cluster

(non solo un oggetto argomento configurato) su tale gestore code. Allo stesso modo, un messaggio di eliminazione della sottoscrizione proxy viene inviato quando viene eliminata l'ultima sottoscrizione per una specifica stringa di argomenti in cluster.

Per l'instradamento diretto, ogni gestore code con sottoscrizioni invia tali sottoscrizioni proxy a ogni altro gestore code nel cluster. Per l'instradamento dell'host argomento, ogni gestore code con sottoscrizioni invia solo le sottoscrizioni proxy a ciascun gestore code che ospita una definizione per tale argomento del cluster. Pertanto, con l'instradamento diretto, maggiore è il numero di gestori code presenti nel cluster, maggiore è il sovraccarico di gestione delle sottoscrizioni proxy. Mentre, con l'instradamento dell'host argomento, il numero di gestori code nel cluster non è un fattore.

In entrambi i modelli di instradamento, se una soluzione di pubblicazione / sottoscrizione è composta da molte stringhe di argomenti univoche sottoscritte, o gli argomenti su un gestore code nel cluster sono frequentemente sottoscritti e annullati, verrà visualizzato un sovraccarico significativo su tale gestore code, causato dalla costante generazione di messaggi che distribuiscono ed eliminano le sottoscrizioni proxy. Con l'instradamento diretto, a ciò si aggiunge la necessità di inviare questi messaggi a ogni gestore code nel cluster.

Se la frequenza di modifica delle sottoscrizioni è troppo elevata per essere adattata, anche all'interno di un sistema instradato dell'host argomento, consultare Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione per informazioni sui modi per ridurre il sovraccarico della sottoscrizione proxy.

Definizione degli argomenti del 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.

Quando si utilizza l'instradamento diretto, l'ubicazione della definizione dell'argomento in cluster non influenza direttamente il comportamento del sistema, poiché tutti i gestori code nel cluster utilizzano la definizione dell'argomento nello stesso modo. Pertanto, è necessario definire l'argomento su qualsiasi gestore code che sarà membro del cluster per tutto il tempo necessario e che si trova su un sistema sufficientemente affidabile da essere regolarmente in contatto con i gestori code del repository completo.

Quando si utilizza l'instradamento dell'host argomento, l'ubicazione della definizione dell'argomento in cluster è molto importante, poiché altri gestori code nel cluster creano canali a questo gestore code e inviano ad esso informazioni di sottoscrizione e pubblicazioni. Per scegliere il gestore code migliore per ospitare la definizione dell'argomento, è necessario comprendere l'instradamento dell'host dell'argomento. Consultare “Instradamento host argomento nei cluster di pubblicazione / sottoscrizione” a pagina 83.

Se si dispone di un argomento in cluster e di un oggetto argomento locale, l'argomento locale ha la precedenza. Consultare “Più definizioni di argomenti cluster con lo stesso nome” a pagina 100.

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

Eredità argomenti raggruppati

Generalmente, le applicazioni di pubblicazione e sottoscrizione in una topologia di pubblicazione / sottoscrizione in cluster si aspettano di funzionare allo stesso modo, indipendentemente dal gestore

code nel cluster a cui sono connesse. Questo è il motivo per cui gli oggetti argomento gestiti in cluster vengono propagati a ogni gestore code nel cluster.

Un oggetto argomento gestito eredita il suo comportamento da altri oggetti argomento gestiti più in alto nella struttura ad albero degli argomenti. Questa eredità si verifica quando non è stato impostato un valore esplicito per un parametro dell'argomento.

Nel caso di pubblicazione / sottoscrizione in cluster, è importante considerare tale eredità perché introduce la possibilità che i publisher e i sottoscrittori si comportino in modo diverso a seconda del gestore code a cui si connettono. Se un oggetto argomento in cluster lascia dei parametri per ereditare da oggetti argomento superiori, l'argomento potrebbe comportarsi in modo diverso su gestori code differenti nel cluster. Allo stesso modo, gli oggetti argomento definiti localmente definiti al di sotto di un oggetto argomento in cluster nella struttura ad albero degli argomenti indicano che tali argomenti inferiori sono ancora in cluster, ma l'oggetto locale potrebbe modificarne il funzionamento in modo diverso rispetto ad altri gestori code nel cluster.

Sottoscrizioni jolly

Le sottoscrizioni proxy vengono create quando le sottoscrizioni locali vengono effettuate a una stringa di argomenti che si risolve in un oggetto argomento del cluster o al di sotto di esso. Se una sottoscrizione con caratteri jolly viene effettuata più in alto nella gerarchia di argomenti rispetto a qualsiasi argomento cluster, non dispone di sottoscrizioni proxy inviate intorno al cluster per l'argomento cluster corrispondente e pertanto non riceve alcuna pubblicazione da altri membri del cluster. Tuttavia, riceve pubblicazioni dal gestore code locale.

Tuttavia, se un'altra applicazione effettua la sottoscrizione a una stringa di argomenti che si risolve in o al di sotto dell'argomento del cluster, le sottoscrizioni proxy vengono generate e le pubblicazioni vengono propagate a questo gestore code. All'arrivo, la sottoscrizione di un carattere jolly originale e superiore è considerata un destinatario legittimo di tali pubblicazioni e riceve una copia. Se questo comportamento non è richiesto, impostare **`WILDCARD(BLOCK)`** sull'argomento del cluster. Ciò rende il carattere jolly originale non considerato una sottoscrizione legittima e arresta la ricezione di qualsiasi pubblicazione (locale o da un altro punto del cluster) sull'argomento del cluster o sui relativi argomenti secondari.

Concetti correlati

Utilizzo degli argomenti di gestione

Utilizzo delle sottoscrizioni

Riferimenti correlati

VISUALIZZAZIONEARGOMENTO

STATOVISUALIZZAZIONEP

VISUALIZZAZIONESUB

Attributi argomento cluster

Quando un oggetto argomento ha l'attributo del nome cluster impostato, la definizione dell'argomento viene propagata a tutti i gestori code nel cluster. Ogni gestore code utilizza gli attributi dell'argomento propagato per controllare il comportamento delle applicazioni di pubblicazione / sottoscrizione.

Un oggetto argomento ha diversi attributi che si applicano ai cluster di pubblicazione / sottoscrizione. Alcuni controllano il comportamento generale delle applicazioni di pubblicazione e sottoscrizione e altri il modo in cui l'argomento viene utilizzato nel cluster.

Una definizione di oggetto argomento con cluster deve essere configurata in modo che tutti i gestori code nel cluster possano utilizzarla correttamente.

Ad esempio, se le code modello da utilizzare per le sottoscrizioni gestite (`MDURMDL` e `MNDURMDL`) sono impostati su un nome coda non predefinito, tale coda modello denominata deve essere definita su tutti i gestori code in cui verranno create le sottoscrizioni gestite.

Allo stesso modo, se un attributo è impostato su ASPARENT, il funzionamento dell'argomento dipenderà dai nodi più elevati nella struttura ad albero degli argomenti (consultare Oggetti argomento di gestione) su ogni singolo gestore code nel cluster. Ciò potrebbe determinare un comportamento diverso durante la pubblicazione o la sottoscrizione da gestori code differenti.

Gli attributi principali direttamente correlati al comportamento di pubblicazione / sottoscrizione nel cluster sono i seguenti:

CLROUTE
Questo parametro controlla l'instradamento dei messaggi tra i gestori code in cui sono connessi i publisher e i gestori code in cui esistono sottoscrizioni corrispondenti.

- L'instradamento viene configurato in modo che sia diretto tra questi gestori code o tramite un gestore code su cui è presente una definizione dell'argomento cluster. Consultare Cluster di pubblicazione / sottoscrizione per ulteriori dettagli.
- Non è possibile modificare **CLROUTE** mentre è impostato il parametro **CLUSTER** . Per modificare **CLROUTE**, impostare prima la proprietà **CLUSTER** su un valore vuoto. Ciò impedisce alle applicazioni che utilizzano l'argomento di comportarsi in modo cluster. Ciò a sua volta determina un'interruzione delle pubblicazioni consegnate alle sottoscrizioni, pertanto è necessario disattivare anche la messaggistica di pubblicazione / sottoscrizione durante l'esecuzione della modifica.

PROXYSUB
Questo parametro controlla quando vengono effettuate le sottoscrizioni proxy.

- FIRSTUSE è il valore predefinito e fa sì che le sottoscrizioni proxy vengano inviate in risposta alle sottoscrizioni locali su un gestore code in una topologia di pubblicazione / sottoscrizione distribuita e annullate quando non più richieste. Per i dettagli sul motivo per cui si potrebbe voler modificare questo attributo dal valore predefinito di FIRSTUSE, consultare Inoltro sottoscrizione proxy individuale e *pubblicazione ovunque* .
- Per abilitare la *pubblicazione ovunque*, impostare il parametro **PROXYSUB** su FORCE per un oggetto argomento di alto livello. Ne risulta una singola sottoscrizione proxy con caratteri jolly che corrisponde a tutti gli argomenti al di sotto di questo oggetto argomento nella struttura ad albero degli argomenti.

Nota: L'impostazione dell'attributo **PROXYSUB(FORCE)** in un cluster di pubblicazione / sottoscrizione di grandi dimensioni o occupato può causare un carico eccessivo sulle risorse di sistema. L'attributo **PROXYSUB(FORCE)** viene propagato a ogni gestore code, non solo al gestore code su cui è stato definito l'argomento. Ciò fa sì che ogni gestore code nel cluster crei una sottoscrizione proxy con caratteri jolly.

Una copia di un messaggio per questo argomento, pubblicato su qualsiasi gestore code del cluster, viene inviata a ogni gestore code del cluster, direttamente o tramite un gestore code dell'host argomento, a seconda dell'impostazione **CLROUTE** .

Quando l'argomento viene instradato direttamente, ogni gestore code crea canali mittente del cluster a ogni altro gestore code. Quando l'argomento è un host argomento instradato, i canali per ciascun gestore code dell'host argomento vengono creati da ogni gestore code del cluster.

Per ulteriori informazioni sul parametro **PROXYSUB** quando utilizzato nei cluster, vedi Direct routed publish / subscribe performance.

PUBSCOBE e SUBSCOPE
Questi parametri determinano se questo gestore code propaga le pubblicazioni ai gestori code nella topologia (cluster di pubblicazione / sottoscrizione o gerarchia) o limita l'ambito al solo gestore code locale. È possibile eseguire il lavoro equivalente in modo programmatico utilizzando MQPMO_SCOPE_QMGR e MQSO_SCOPE_QMGR.

PUBSCOPE
Se un oggetto argomento del cluster è definito con **PUBSCOPE(QMGR)**, la definizione viene condivisa con il cluster, ma l'ambito delle pubblicazioni basate su tale argomento è solo locale e non vengono inviate ad altri gestori code nel cluster.

SUBSCOPE
Se un oggetto argomento del cluster è definito con **SUBSCOPE(QMGR)**, la definizione viene condivisa con il cluster, ma l'ambito delle sottoscrizioni che si basano su tale argomento è solo locale, quindi nessuna sottoscrizione proxy viene inviata ad altri gestori code nel cluster.

Questi due attributi vengono comunemente utilizzati insieme per isolare un gestore code dall'interazione con altri membri del cluster su argomenti particolari. Il gestore code non pubblica o riceve pubblicazioni su tali argomenti da e verso altri membri del cluster. Questa situazione non impedisce la pubblicazione o la sottoscrizione se gli oggetti argomento sono definiti su argomenti secondari.

L'impostazione di **SUBSCOPE** su QMGR su una definizione locale di un argomento non impedisce ad altri gestori code nel cluster di propagare le relative sottoscrizioni proxy al gestore code se utilizzano una versione cluster dell'argomento, con **SUBSCOPE(ALL)**. Tuttavia, se la definizione locale imposta anche **PUBSCOPE** su QMGR , le sottoscrizioni proxy non vengono inviate alle pubblicazioni da questo gestore code.

Concetti correlati

Ambito della pubblicazione

Ambito della sottoscrizione

Più definizioni di argomenti cluster con lo stesso nome

È possibile definire lo stesso oggetto argomento cluster denominato su più di un gestore code nel cluster e in determinati scenari ciò abilita un comportamento specifico. Quando esistono più definizioni di argomenti cluster con lo stesso nome, la maggior parte delle proprietà deve corrispondere. In caso contrario, vengono riportati errori o avvertenze in base alla significatività della mancata corrispondenza.

In generale, se si verifica una mancata corrispondenza nelle proprietà di più definizioni di argomenti cluster, vengono emesse delle avvertenze e una delle definizioni di oggetti argomento viene utilizzata da ciascun gestore code nel cluster. La definizione utilizzata da ciascun gestore code non è deterministica o congruente tra i gestori code nel cluster. Tali disallineamenti dovrebbero essere risolti il più rapidamente possibile.

Durante l'impostazione o la manutenzione del cluster, a volte è necessario creare più definizioni di argomenti del cluster che non sono identiche. Tuttavia, ciò è sempre utile solo come misura temporanea, e viene quindi trattato come una potenziale condizione di errore.

Quando vengono rilevate delle mancate corrispondenze, i seguenti messaggi di avviso vengono scritti nel log degli errori di ciascun gestore code:

- Multi Su Multipiattaforme, AMQ9465 e AMQ9466.
- z/OS Su z/OS, CSQX465I e CSQX466I.

Le proprietà scelte per qualsiasi stringa di argomenti su ciascun gestore code possono essere determinate visualizzando lo stato dell'argomento piuttosto che le definizioni degli oggetti argomento, ad esempio utilizzando **DISPLAY TPSTATUS**.

In alcune situazioni, un conflitto nelle proprietà di configurazione è abbastanza grave da interrompere la creazione dell'oggetto argomento o da far sì che gli oggetti non corrispondenti vengano contrassegnati come non validi e non propagati nel cluster (consultare **CLSTATE** in DISPLAY TOPIC). Queste situazioni si verificano quando si verifica un conflitto nella proprietà di instradamento cluster (**CLROUTE**) delle definizioni argomento. Inoltre, a causa dell'importanza della coerenza tra le definizioni instradate dell'host argomento, ulteriori incongruenze vengono respinte come descritto in dettaglio nelle sezioni successive di questo articolo.

Se il conflitto viene rilevato al momento della definizione dell'oggetto, la modifica della configurazione viene rifiutata. Se successivamente vengono rilevati dai gestori code del repository completo, i seguenti messaggi di avvertenza vengono scritti nei log degli errori dei gestori code:

- Multi Su Multipiattaforme: AMQ9879
- z/OS Su z/OS CSQX879E.

Quando più definizioni dello stesso oggetto argomento sono definite nel cluster, una definizione definita localmente ha la precedenza su qualsiasi definizione definita in remoto. Pertanto, se esistono differenze nelle definizioni, i gestori code che ospitano le definizioni multiple si comportano in modo diverso.

L'effetto della definizione di un argomento non cluster con lo stesso nome di un argomento cluster da un altro gestore code

È possibile definire un oggetto argomento gestito che non è in cluster su un gestore code che si trova in un cluster e definire simultaneamente lo stesso oggetto argomento denominato come definizione di argomento in cluster su un gestore code differente. In tal modo, l'oggetto argomento definito localmente ha la precedenza su tutte le definizioni remote dello stesso nome.

Ciò ha l'effetto di impedire il funzionamento del cluster dell'argomento quando viene utilizzato da questo gestore code. In altre parole, le sottoscrizioni potrebbero non ricevere pubblicazioni dai publisher remoti e i messaggi dai publisher potrebbero non essere propagati alle sottoscrizioni remote nel cluster.

Prima di configurare un sistema di questo tipo, è necessario prestare particolare attenzione, poiché ciò può creare confusione.

Nota: Se un singolo gestore code deve impedire che le pubblicazioni e le sottoscrizioni si propaghino intorno al cluster, anche quando l'argomento è stato raggruppato altrove, un approccio alternativo consiste nell'impostare gli ambiti di pubblicazione e sottoscrizione solo sul gestore code locale. Consultare “Attributi argomento cluster” a pagina 98.

Più definizioni dell'argomento cluster in un cluster con instradamento diretto

Per l'instradamento diretto, di solito non si definisce lo stesso argomento cluster su più di un gestore code cluster. Questo perché l'instradamento diretto rende l'argomento disponibile in tutti i gestori code nel cluster, indipendentemente dal gestore code su cui è stato definito. Inoltre, l'aggiunta di più definizioni di argomenti cluster aumenta in modo significativo l'attività del sistema e la complessità amministrativa, e con l'aumento della complessità si ha una maggiore probabilità di errore umano:

- Ciascuna definizione risulta in un oggetto argomento cluster aggiuntivo che viene inviato agli altri gestori code nel cluster, inclusi gli altri gestori code dell'host argomento cluster.
- Tutte le definizioni per un determinato argomento in un cluster devono essere identiche, altrimenti è difficile stabilire quale definizione argomento viene utilizzata da un gestore code.

Non è inoltre essenziale che il solo gestore code host sia continuamente disponibile per il corretto funzionamento dell'argomento nel cluster, poiché la definizione dell'argomento del cluster viene memorizzata nella cache dai gestori code del repository completo e da tutti gli altri gestori code nei repository del cluster parziali. Per ulteriori informazioni, consultare Disponibilità dei gestori code dell'host argomento che utilizzano l'instradamento diretto.

Per una situazione in cui potrebbe essere necessario definire temporaneamente un argomento del cluster su un secondo gestore code, ad esempio quando l'host esistente dell'argomento deve essere rimosso dal cluster, fare riferimento a Spostamento di una definizione dell'argomento del cluster in un gestore code differente.

Se occorre modificare una definizione dell'argomento cluster, effettuare la modifica nello stesso gestore code in cui era stata definita. Il tentativo di modificarla da un altro gestore code potrebbe accidentalmente creare una seconda definizione dell'argomento con attributi dell'argomento in conflitto.

Più definizioni dell'argomento cluster in un cluster instradato all'host argomento

Quando un argomento cluster viene definito con un instradamento cluster di *host argomento*, l'argomento viene propagato su tutti i gestori code nel cluster proprio come per gli argomenti instradati *diretti* . Inoltre, tutta la messaggistica di pubblicazione / sottoscrizione per tale argomento viene instradata attraverso i gestori code in cui è definito tale argomento. Pertanto, l'ubicazione e il numero di definizioni dell'argomento nel cluster diventano importanti (consultare “Instradamento host argomento nei cluster di pubblicazione / sottoscrizione” a pagina 83).

Per garantire una disponibilità e una scalabilità adeguate, è utile, se possibile, disporre di più definizioni di argomenti. Consultare Disponibilità dei gestori code dell'host argomento che utilizzano l'instradamento dell'host argomento.

Quando si aggiungono o si rimuovono ulteriori definizioni di un argomento instradato *host argomento* in un cluster, è necessario considerare il flusso di messaggi al momento della modifica della configurazione. Se i messaggi vengono pubblicati nel cluster nell'argomento al momento della modifica, è richiesto un processo a fasi per aggiungere o rimuovere una definizione di argomento. Fare riferimento a Spostamento di una definizione di argomento del cluster in un gestore code differente e a Aggiunta di ulteriori host argomento a un cluster instradato dell'host argomento.

Come precedentemente spiegato, le proprietà delle definizioni multiple devono corrispondere, con la possibile eccezione del parametro **PUB** , come descritto nella sezione successiva. Quando le pubblicazioni vengono instradate tramite i gestori code dell'host argomento, è ancora più importante che più definizioni siano congruenti. Pertanto, un'incongruenza rilevata nella stringa di argomenti o nel nome cluster viene rifiutata se una o più definizioni di argomenti sono state configurate per l'instradamento del cluster di host di argomenti.

Nota: Le definizioni di argomenti del cluster vengono rifiutate anche se viene effettuato un tentativo di configurarle sopra o sotto un altro argomento nella struttura ad albero degli argomenti, dove la definizione di argomenti del cluster esistente è configurata per l'instradamento dell'host dell'argomento. Ciò evita ambiguità nell'instradamento delle pubblicazioni rispetto alle sottoscrizioni con caratteri jolly.

Gestione speciale per il parametro PUB

Il parametro di **PUB** viene utilizzato per controllare quando le applicazioni possono pubblicare un argomento. Nel caso dell'instradamento dell'host argomento in un cluster, può anche controllare quali gestori code dell'host argomento vengono utilizzati per instradare pubblicazioni. Per questo motivo è consentito avere più definizioni dello stesso oggetto argomento nel cluster, con impostazioni diverse per il parametro PUB .

Se più definizioni cluster remote di un argomento hanno impostazioni differenti per questo parametro, l'argomento consente l'invio e la consegna delle pubblicazioni alle sottoscrizioni se vengono soddisfatte le condizioni riportate di seguito:

- Non c'è un oggetto argomento corrispondente definito sul gestore code a cui è connesso il publisher impostato su `PUB(DISABLED)`.
- Una o più definizioni di argomenti multipli nel cluster sono impostate su `PUB(ENABLED)`, oppure una o più definizioni di argomenti multipli sono impostate su `PUB(ASPARENT)` e i gestori code locali in cui il publisher è connesso e la sottoscrizione definita sono impostati su `PUB(ENABLED)` in un punto più alto nella struttura ad albero degli argomenti.

Per l'instradamento dell'host argomento, quando i messaggi vengono pubblicati dalle applicazioni connesse ai gestori code che non sono host argomento, i messaggi vengono instradati solo ai gestori code che ospitano l'argomento in cui il parametro **PUB** non è stato esplicitamente impostato su `DISABLED`. È quindi possibile utilizzare l'impostazione `PUB(DISABLED)` per disattivare il traffico di messaggi attraverso determinati host argomento. È possibile eseguire questa operazione per preparare la manutenzione o la rimozione di un gestore code o per i motivi descritti in Aggiunta di ulteriori host argomento a un cluster instradato di host argomento.

Disponibilità dei gestori code dell'host argomento del cluster
Progettare il cluster di pubblicazione / sottoscrizione per ridurre al minimo il rischio che, se un gestore code dell'host argomento diventa non disponibile, il cluster non sarà più in grado di elaborare il traffico per l'argomento. L'effetto di un gestore code dell'host argomento che diventa non disponibile dipende dal fatto che il cluster stia utilizzando l'instradamento dell'host argomento o l'instradamento diretto.

Disponibilità dei gestori code dell'host argomento che utilizzano l'instradamento diretto

Per l'instradamento diretto, di solito non si definisce lo stesso argomento cluster su più di un gestore code cluster. Questo perché l'instradamento diretto rende l'argomento disponibile in tutti i gestori code nel cluster, indipendentemente dal gestore code su cui è stato definito.Consultare Definizioni di più argomenti cluster in un cluster instradato direttamente.

In un cluster, ogni volta che l'host di un oggetto in cluster (ad esempio una coda in cluster o un argomento in cluster) diventa non disponibile per un periodo di tempo prolungato, gli altri membri del cluster alla fine annulleranno la conoscenza di tali oggetti. Nel caso di un argomento con cluster, se il gestore code dell'host dell'argomento del cluster diventa non disponibile, gli altri gestori code continuano a elaborare le richieste di pubblicazione / sottoscrizione per l'argomento in un cluster diretto (ovvero, inviando pubblicazioni a sottoscrizioni su gestori code remoti) per almeno 60 giorni dall'ultima volta che il gestore code che ospita l'argomento è stato in comunicazione con i gestori code del repository completo. Se il gestore code su cui è stato definito l'oggetto argomento cluster non viene mai più reso disponibile, alla fine gli oggetti argomento memorizzati nella cache sugli altri gestori code vengono eliminati e l'argomento ritorna a un argomento locale, nel qual caso le sottoscrizioni cessano di ricevere le pubblicazioni dalle applicazioni connesse ai gestori code remoti.

Con un periodo di 60 giorni per ripristinare il gestore code su cui si definisce un oggetto argomento del cluster, non è necessario adottare misure speciali per garantire che l'host argomento del cluster rimanga disponibile (notare, tuttavia, che le sottoscrizioni definite sull'host argomento del cluster non disponibile non rimangono disponibili). Il periodo di 60 giorni è sufficiente per far fronte a problemi tecnici ed è probabile che venga superato solo a causa di errori amministrativi. Per attenuare questa possibilità, se l'host dell'argomento del cluster non è disponibile, tutti i membri del cluster scrivono ogni ora i messaggi di log degli errori, indicando che il relativo oggetto dell'argomento del cluster memorizzato nella cache non è stato aggiornato. Rispondere a questi messaggi verificando che il gestore code su cui è definito l'oggetto argomento cluster sia in esecuzione. Se non è possibile rendere nuovamente disponibile il gestore code dell'host argomento del cluster, definire la stessa definizione di argomento del cluster, con esattamente gli stessi attributi, su un altro gestore code del cluster.

Disponibilità dei gestori code dell'host argomento che utilizzano l'instradamento dell'host argomento

Per l'instradamento dell'host dell'argomento, tutta la messaggistica di pubblicazione / sottoscrizione per un argomento viene instradata attraverso i gestori code in cui è definito tale argomento. Per questo motivo, è molto importante considerare la disponibilità continua di questi gestori code nel cluster.
Se un host dell'argomento diventa non disponibile e non esiste alcun altro host per l'argomento, il traffico dai publisher ai sottoscrittori su gestori code differenti nel cluster si arresta immediatamente per l'argomento. Se sono disponibili ulteriori host argomento, i gestori code del cluster instradano il nuovo traffico di pubblicazione attraverso questi host argomento, fornendo la disponibilità continua delle rotte dei messaggi.

Per quanto riguarda gli argomenti diretti, dopo 60 giorni, se il primo host argomento non è ancora disponibile, la conoscenza dell'argomento di tale host argomento viene rimossa dal cluster. Se questa è l'ultima definizione rimanente per questo argomento nel cluster, tutti gli altri gestori code cessano di inoltrare le pubblicazioni a qualsiasi host argomento per l'instradamento.

Per garantire una disponibilità e una scalabilità adeguate, è quindi utile, se possibile, definire ogni argomento su almeno due gestori code cluster. Ciò fornisce una protezione contro qualsiasi gestore code dell'host argomento specificato che diventa non disponibile. Consultare anche Più definizioni di argomenti del cluster in un cluster instradato dell'host argomento.

Se non è possibile configurare più host di argomenti (ad esempio, perché è necessario conservare l'ordine dei messaggi) e non è possibile configurare solo un host di argomenti (poiché la disponibilità di un singolo gestore code non deve influenzare il flusso di pubblicazioni per le sottoscrizioni in tutti i gestori code nel cluster), considerare la configurazione dell'argomento come un argomento instradato direttamente. In questo modo si evita di fare affidamento su un singolo gestore code per l'intero cluster, ma è comunque necessario che ogni singolo gestore code sia disponibile per elaborare le sottoscrizioni e i publisher ospitati localmente.

Blocco della pubblicazione / sottoscrizione in cluster

L'introduzione del primo argomento del cluster instradato direttamente in un cluster forza ogni gestore code nel cluster a rendersi conto di ogni altro gestore code e potenzialmente fa sì che creino canali l'uno per l'altro. Se ciò non è opportuno, è necessario configurare la pubblicazione / sottoscrizione instradata dell'host argomento. Se l'esistenza di un argomento cluster instradato direttamente potrebbe

compromettere la stabilità del cluster, a causa dei problemi di ridimensionamento di ciascun gestore code, è possibile disabilitare completamente la funzionalità di pubblicazione / sottoscrizione del cluster impostando **PSCLUS** su DISABLED su ogni gestore code del cluster.

Come descritto in"Instradamento diretto nei cluster di pubblicazione / sottoscrizione" a pagina 78, quando si introduce un argomento con cluster instradato direttamente in un cluster, tutti i repository parziali vengono automaticamente informati di tutti gli altri membri del cluster. L'argomento con cluster può anche creare sottoscrizioni su tutti gli altri nodi (ad esempio, dove **PROXYSUB(FORCE)** è specificato) e causare l'avvio di un numero elevato di canali da un gestore code, anche quando non sono presenti sottoscrizioni locali. Ciò inserisce un carico aggiuntivo immediato su ciascun gestore code nel cluster. Per un cluster che contiene molti gestori code, ciò può causare una riduzione significativa delle prestazioni. Pertanto, l'introduzione della pubblicazione / sottoscrizione diretta instradata a un cluster deve essere pianificata attentamente.

Quando si sa che un cluster non può contenere i costi generali della pubblicazione / sottoscrizione instradata diretta, è possibile utilizzare invece la pubblicazione / sottoscrizione instradata dell'host argomento. Per una panoramica delle differenze, consultare "Progettazione di cluster di pubblicazione / sottoscrizione" a pagina 76.

Se si preferisce disabilitare completamente la funzionalità di pubblicazione / sottoscrizione per il cluster, è possibile farlo impostando l'attributo del gestore code **PSCLUS** su DISABLED su ogni gestore code nel cluster. Questa impostazione disabilita la pubblicazione / sottoscrizione instradata diretta e l'host argomento instradato nel cluster, modificando tre aspetti della funzionalità del gestore code:

- Un amministratore di questo gestore code non è più in grado di definire un oggetto Topic come cluster.
- Le definizioni di argomenti in entrata o le sottoscrizioni proxy da altri gestori code vengono rifiutate e viene registrato un messaggio di avvertenza per informare l'amministratore della configurazione non corretta.
- I repository completi non condividono più automaticamente le informazioni su ogni gestore code con tutti gli altri repository parziali quando ricevono una definizione di argomento.

Anche se **PSCLUS** è un parametro di ogni singolo gestore code in un cluster, non è destinato a disabilitare in modo selettivo la pubblicazione / sottoscrizione in un sottoinsieme di gestori code nel cluster. Se si disabilita in modo selettivo in questo modo, verranno visualizzati messaggi di errore frequenti. Ciò è dovuto al fatto che le sottoscrizioni proxy e le definizioni degli argomenti vengono costantemente visualizzate e rifiutate se un argomento è raggruppato in un gestore code in cui **PSCLUS** è abilitato.

Si consiglia pertanto di impostare **PSCLUS** su DISABLED su ogni gestore code nel cluster. Tuttavia, in pratica questo stato può essere difficile da raggiungere e gestire, ad esempio i gestori code possono unirsi e lasciare il cluster in qualsiasi momento. È necessario almeno assicurarsi che **PSCLUS** sia impostato su DISABLED su tutti i gestori code del repository completo. Se si esegue questa operazione, e un argomento in cluster viene successivamente definito su un gestore code ABILITATO nel cluster, ciò non fa in modo che i repository completi informino ogni gestore code di ogni altro gestore code, pertanto il cluster è protetto da potenziali problemi di scalabilità in tutti i gestori code. In questo scenario, l'origine dell'argomento del cluster viene riportata nei log degli errori dei gestori code del repository completo.

Se un gestore code partecipa a uno o più cluster di pubblicazione / sottoscrizione e anche a uno o più cluster point - to - point, è necessario impostare **PSCLUS** su ABILITATO su tale gestore code. Per questo motivo, quando si sovrappone un cluster point-to-point con un cluster di pubblicazione - sottoscrizione, è necessario utilizzare una serie separata di repository completi in ogni cluster. Questo approccio consente alle definizioni degli argomenti e alle informazioni su ogni gestore code di fluire solo nel cluster di pubblicazione / sottoscrizione.

Per evitare configurazioni incongruenti quando si passa da **PSCLUS** da ENABLED a DISABLED, non possono esistere oggetti argomento in cluster in alcun cluster di cui questo gestore code è membro. Tali argomenti, anche quelli definiti in remoto, devono essere eliminati prima di modificare **PSCLUS** in DISABLED.

Per ulteriori informazioni su **PSCLUS**, consultare ALTER QMGR (PSCLUS).

Concetti correlati

Prestazioni cluster di pubblicazione / sottoscrizione instradate dirette

Pubblica / sottoscrivi e più cluster

Un singolo gestore code può essere membro di più di un cluster. Questa disposizione è talvolta nota come *cluster di sovrapposizione*. Tramite tale sovrapposizione, le code cluster possono essere rese accessibili da più cluster e il traffico di messaggi point-to-point può essere instradato dai gestori code di un cluster ai gestori code di un altro cluster. Gli argomenti raggruppati nei cluster di pubblicazione / sottoscrizione non forniscono la stessa funzionalità. Pertanto, il loro comportamento deve essere chiaramente compreso quando si utilizzano più cluster.

A differenza di una coda, non è possibile associare una definizione di argomento a più di un cluster. L'ambito di un argomento con cluster è limitato ai gestori code nello stesso cluster per cui è definito l'argomento. Ciò consente la propagazione delle pubblicazioni alle sottoscrizioni solo sui gestori code nello stesso cluster.

Una struttura ad albero degli argomenti del gestore code

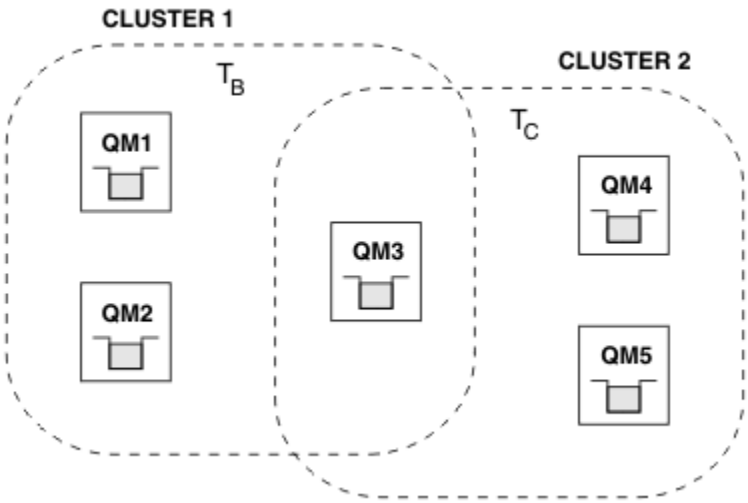


Figura 28. Sovrapposizione di cluster: due cluster ciascuno che effettua la sottoscrizione a argomenti differenti

Quando un gestore code è membro di più cluster, viene informato di tutti gli argomenti del cluster definiti in ognuno di questi cluster. Ad esempio, nella figura precedente, QM3 è consapevole sia degli oggetti argomento del cluster gestiti T_B che T_C, mentre QM1 è consapevole solo di T_B. QM3 applica entrambe le definizioni di argomento al relativo argomento locale e, pertanto, ha un comportamento diverso rispetto a QM1 per alcuni argomenti. Per questo motivo è importante che gli argomenti raggruppati da cluster differenti non interferiscano tra loro. L'interferenza può verificarsi quando un argomento cluster viene definito sopra o sotto un altro argomento cluster in un cluster differente (ad esempio, hanno stringhe di argomenti di `/Sport` e `/Sport/Football`) o anche per la stessa stringa di argomenti in entrambi. Un'altra forma di interferenza è quando gli oggetti argomento con cluster gestiti sono definiti con lo stesso nome oggetto in cluster differenti, ma per stringhe argomento differenti.

Se viene effettuata tale configurazione, la consegna delle pubblicazioni alle sottoscrizioni corrispondenti diventa molto dipendente dalle ubicazioni relative dei publisher e dei sottoscrittori rispetto al cluster. Per questo motivo, non è possibile fare affidamento su una configurazione di questo tipo ed è necessario modificarla per rimuovere gli argomenti che interferiscono.

Quando si pianifica una topologia cluster sovrapposta con la messaggistica di pubblicazione / sottoscrizione, è possibile evitare qualsiasi interferenza trattando la struttura ad albero degli argomenti e i nomi degli oggetti argomento raggruppati come se si estendessero a tutti i cluster sovrapposti nella topologia.

Integrazione di più cluster di pubblicazione / sottoscrizione

Se esiste un requisito per la messaggistica di pubblicazione / sottoscrizione per estendere i gestori code in cluster differenti, sono disponibili due opzioni:

- Collegare i cluster tramite l'utilizzo di una configurazione della gerarchia di pubblicazione / sottoscrizione. Consultare Combinazione di spazi argomento di più cluster.
- Creare un cluster aggiuntivo che si sovrapponga ai cluster esistenti e includa tutti i gestori code che devono pubblicare o sottoscrivere un particolare argomento.

Con quest' ultima opzione, è necessario considerare attentamente la dimensione del cluster e il meccanismo di instradamento del cluster più efficace. Consultare“Progettazione di cluster di pubblicazione / sottoscrizione” a pagina 76.

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

Ci sono alcune limitazioni da considerare quando si progetta un cluster di pubblicazione / sottoscrizione per lavorare con le pubblicazioni conservate.

Considerazioni

Considerazione 1 I gestori code del seguente cluster memorizzano sempre l'ultima versione di una pubblicazione conservata:

- Il gestore code del publisher
- In un cluster instradato di host argomento, l'host argomento (fornito con un solo host argomento per l'argomento, come spiegato nella prossima sezione di questo articolo)
- Tutti i gestori code con sottoscrizioni corrispondenti alla stringa di argomenti della pubblicazione conservata

Considerazione 2: i gestori code non ricevono le pubblicazioni conservate aggiornate quando non dispongono di sottoscrizioni. Di conseguenza, qualsiasi pubblicazione conservata memorizzata su un gestore code che non fa più sottoscrizione all'argomento diventerà obsoleta.

Considerazione 3: quando si crea una sottoscrizione, se è presente una copia locale di una pubblicazione conservata per la stringa di argomenti, la copia locale viene consegnata alla sottoscrizione. Se si è il primo sottoscrittore per una determinata stringa di argomenti, anche una pubblicazione conservata corrispondente viene consegnata da uno dei seguenti membri del cluster:

- In un cluster instradato direttamente, il gestore code del publisher
- In un cluster di host argomento instradati, gli host argomento per l'argomento fornito

La consegna di una pubblicazione conservata da un host argomento o da un gestore code di pubblicazione al gestore code di sottoscrizione è asincrona alle chiamate MQSUB . Pertanto, se si utilizza la chiamata MQSUBRQ , la pubblicazione conservata più recente potrebbe essere persa fino a una successiva chiamata a MQSUBRQ.

Implicazioni

Per qualsiasi cluster di pubblicazione / sottoscrizione, quando viene effettuata una prima sottoscrizione, il gestore code locale potrebbe archiviare una copia obsoleta di una pubblicazione conservata e questa è la copia consegnata alla nuova sottoscrizione. L'esistenza di una sottoscrizione sul gestore code locale significa che questa verrà risolta al successivo aggiornamento della pubblicazione conservata.

Per un cluster di pubblicazione / sottoscrizione instradato dell'host argomento, se si configura più di un host argomento per un determinato argomento, i nuovi sottoscrittori potrebbero ricevere la pubblicazione conservata più recente da un host argomento oppure potrebbero ricevere una pubblicazione conservata obsoleta da un altro host argomento (con l'ultima perdita). Per l'instradamento dell'host argomento, è consuetudine configurare più host argomento per un determinato argomento. Tuttavia, se si prevede che

le applicazioni utilizzino le pubblicazioni conservate, è necessario configurare solo un host argomento per ciascun argomento.

Per qualsiasi stringa di argomenti fornita, è necessario utilizzare solo un singolo publisher e assicurarsi che il publisher utilizzi sempre lo stesso gestore code. Se non si esegue questa operazione, diverse pubblicazioni conservate potrebbero essere attive su gestori code differenti per lo stesso argomento, causando un comportamento imprevisto. Poiché vengono distribuite più sottoscrizioni proxy, è possibile che vengano ricevute più pubblicazioni conservate.

Se si è ancora preoccupati per i sottoscrittori che utilizzano pubblicazioni obsolete, considerare l'impostazione di una scadenza del messaggio quando si crea ogni pubblicazione conservata.

È possibile utilizzare il comando **CLEAR TOPICSTR** per rimuovere una pubblicazione conservata da un cluster di pubblicazione / sottoscrizione. In determinate circostanze, potrebbe essere necessario immettere il comando su più membri del cluster di pubblicazione / sottoscrizione, come descritto in **CLEAR TOPICSTR**.

Sottoscrizioni jolly e pubblicazioni conservate

Se si utilizzano sottoscrizioni con caratteri jolly, le sottoscrizioni proxy corrispondenti consegnate ad altri membri del cluster di pubblicazione / sottoscrizione vengono fornite con caratteri jolly dal separatore di argomenti immediatamente prima del primo carattere jolly. Consultare Caratteri jolly e argomenti cluster.

Pertanto, il carattere jolly utilizzato potrebbe corrispondere a un numero maggiore di stringhe di argomenti e di pubblicazioni conservate, rispetto all'applicazione di sottoscrizione.

Ciò aumenta la capacità di memoria necessaria per le pubblicazioni conservate ed è pertanto necessario assicurarsi che i gestori code host dispongano di capacità di memoria sufficiente.

Concetti correlati

Pubblicazioni conservate

Inoltro e pubblicazione di singole sottoscrizioni proxy ovunque

REFRESH CLUSTER considerazioni per i cluster di pubblicazione / sottoscrizione

Immettendo il comando **REFRESH CLUSTER** il gestore code elimina temporaneamente le informazioni conservate localmente su un cluster, inclusi gli argomenti del cluster e le relative sottoscrizioni proxy associate.

Il tempo impiegato dall'immissione del comando **REFRESH CLUSTER** al punto che il gestore code riacquista una conoscenza completa delle informazioni necessarie per la pubblicazione / sottoscrizione in cluster dipende dalla dimensione del cluster, dalla disponibilità e dalla reattività dei gestori code del repository completo.

Durante il processo di aggiornamento, si verifica un'interruzione del traffico di pubblicazione / sottoscrizione in un cluster di pubblicazione / sottoscrizione. Per i cluster di grandi dimensioni, l'utilizzo del comando **REFRESH CLUSTER** può interrompere il cluster mentre è in corso e di nuovo a intervalli di 27 giorni quando gli oggetti cluster inviano automaticamente gli aggiornamenti di stato a tutti i gestori code interessati. Consultare Refreshing in a large cluster can affect performance and availability of the cluster. Per questi motivi, il comando **REFRESH CLUSTER** deve essere utilizzato in un cluster di pubblicazione / sottoscrizione solo se sotto la guida del centro di assistenza IBM .

L'interruzione del cluster può apparire esternamente come i sintomi seguenti:

- Le sottoscrizioni agli argomenti del cluster su questo gestore code non ricevono pubblicazioni dai publisher connessi ad altri gestori code nel cluster.
- I messaggi pubblicati negli argomenti del cluster su questo gestore code non vengono propagati alle sottoscrizioni su altri gestori code.
- Le sottoscrizioni agli argomenti del cluster su questo gestore code create durante questo periodo non inviano in modo congruente sottoscrizioni proxy ad altri membri del cluster.
- Le sottoscrizioni agli argomenti del cluster su questo gestore code eliminati durante questo periodo non rimuovono in modo congruente le sottoscrizioni proxy da altri membri del cluster.

- Pause di 10 secondi, o più lunghe, nella consegna dei messaggi.
- Errori di **MQPUT** , ad esempio MQRC_PUBLICATION_FAILURE.
- Pubblicazioni collocate nella coda di messaggi non instradabili con motivo MQRC_UNKNOWN_REMOTE_Q_MGR

Per questi motivi le applicazioni di pubblicazione / sottoscrizione devono essere disattivate prima di immettere il comando **REFRESH CLUSTER** .

Dopo che un comando **REFRESH CLUSTER** è stato immesso su un gestore code in un cluster di pubblicazione / sottoscrizione, attendere che tutti i gestori code del cluster e gli argomenti del cluster siano stati aggiornati correttamente, quindi risincronizzare le sottoscrizioni proxy come descritto in Risincronizzazione delle sottoscrizioni proxy. Quando tutte le sottoscrizioni proxy sono state correttamente risincronizzate, riavviare le applicazioni di pubblicazione / sottoscrizione.

Se un comando **REFRESH CLUSTER** sta impiegando molto tempo per essere completato, monitorarlo osservando CURDEPTH di SYSTEM.CLUSTER.COMMAND.QUEUE.

Concetti correlati

“Cluster: utilizzo delle procedure consigliate per REFRESH CLUSTER” a pagina 70
Utilizzare il comando **REFRESH CLUSTER** per eliminare tutte le informazioni contenute localmente su un cluster e ricreare tali informazioni dai repository completi nel cluster. Non è necessario utilizzare questo comando, tranne in circostanze eccezionali. Se hai bisogno di usarlo, ci sono considerazioni speciali su come usarlo. Queste informazioni sono una guida basata sul test e sul feedback dei clienti.

Riferimenti correlati

Problemi dell'applicazione durante l'esecuzione di REFRESH CLUSTER

Riferimento comandi MQSC: REFRESH CLUSTER

Instradamento nelle gerarchie di pubblicazione / sottoscrizione

Se la topologia del gestore code distribuito è una gerarchia di pubblicazione / sottoscrizione e viene effettuata una sottoscrizione su un gestore code, per impostazione predefinita viene creata una sottoscrizione proxy su ogni gestore code nella gerarchia. Le pubblicazioni ricevute su qualsiasi gestore code vengono quindi instradate attraverso la gerarchia a ciascun gestore code su cui è presente una sottoscrizione corrispondente.

Per un'introduzione al modo in cui i messaggi vengono instradati tra i gestori code nelle gerarchie di pubblicazione / sottoscrizione e nei cluster, consultare Reti di pubblicazione / sottoscrizione distribuite.

Quando viene effettuata una sottoscrizione a un argomento su un gestore code in una gerarchia di pubblicazione / sottoscrizione distribuita, il gestore code gestisce il processo mediante il quale la sottoscrizione viene propagata ai gestori code connessi. Le *sottoscrizioni proxy* passano a tutti i gestori code nella rete. Una sottoscrizione proxy fornisce a un gestore code le informazioni necessarie per inoltrare una pubblicazione ai gestori code che ospitano le sottoscrizioni per tale argomento. Ogni gestore code in una gerarchia di pubblicazione / sottoscrizione è consapevole solo delle sue relazioni dirette. Le pubblicazioni immesse in un gestore code vengono inviate, tramite le relazioni dirette, a tali gestori code con sottoscrizioni. Questo è illustrato nella seguente figura, in cui *Sottoscrittore 1* registra una sottoscrizione per un particolare argomento sul gestore code *Asia* (1). Le sottoscrizioni proxy per questa sottoscrizione sul gestore code *Asia* vengono inoltrati a tutti gli altri gestori code nella rete (2,3, 4).

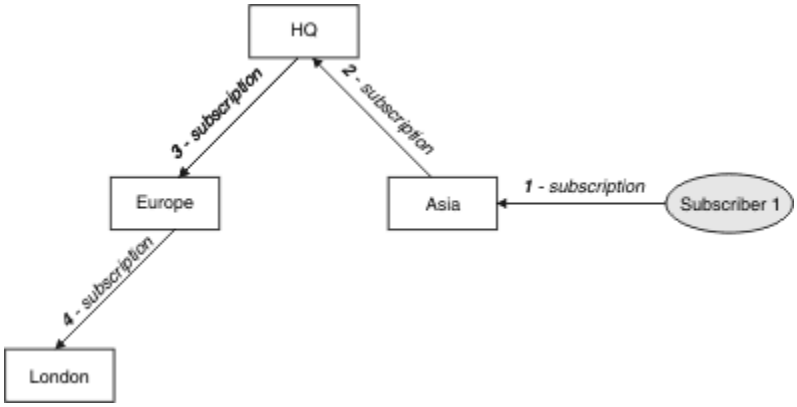


Figura 29. Propagazione delle sottoscrizioni attraverso una rete di gestori code

Un gestore code consolida tutte le sottoscrizioni create su di esso, sia da applicazioni locali che da gestori code remoti. Crea sottoscrizioni proxy per gli argomenti delle sottoscrizioni con i relativi vicini, a meno che non esista già una sottoscrizione proxy. Questo è illustrato nella seguente figura, in cui il *Sottoscrittore 2* registra una sottoscrizione, allo stesso argomento di Figura 29 a pagina 109, sul gestore code *HQ* (5). La sottoscrizione per questo argomento viene inoltrata al gestore code *Asia* , in modo che sia consapevole che le sottoscrizioni esistono altrove sulla rete (6). La sottoscrizione non viene inoltrata al gestore code *Europa* , perché è già stata registrata una sottoscrizione per questo argomento; consultare il passo 3 in Figura 29 a pagina 109.

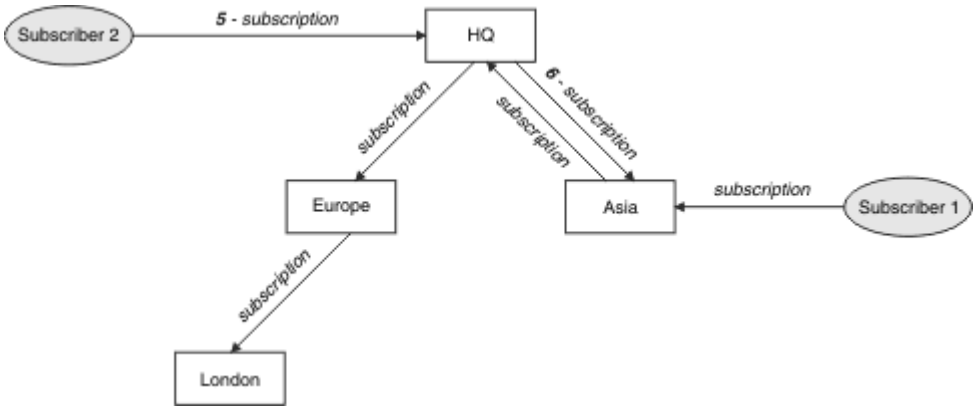


Figura 30. Sottoscrizioni multiple

Quando un'applicazione pubblica informazioni su un argomento, per impostazione predefinita il gestore code di ricezione le inoltra a tutti i gestori code che hanno sottoscrizioni valide per l'argomento. Potrebbe inoltrarlo tramite uno o più gestori code intermedi. Questo è illustrato nella seguente figura, in cui un publisher invia una pubblicazione, sullo stesso argomento di Figura 30 a pagina 109, al gestore code *Europa* (7). Una sottoscrizione per questo argomento esiste da *HQ* a *Europe*, quindi la pubblicazione viene inoltrata al gestore code *HQ* (8). Tuttavia, non esiste alcuna sottoscrizione da *Londra* a *Europa* (solo da *Europa* a *Londra*), quindi la pubblicazione non viene inoltrata al gestore code *Londra* . Il gestore code *HQ* invia la pubblicazione direttamente al *Sottoscrittore 2* e al gestore code *Asia* (9). La pubblicazione viene inoltrata al *Sottoscrittore 1* da *Asia* (10).

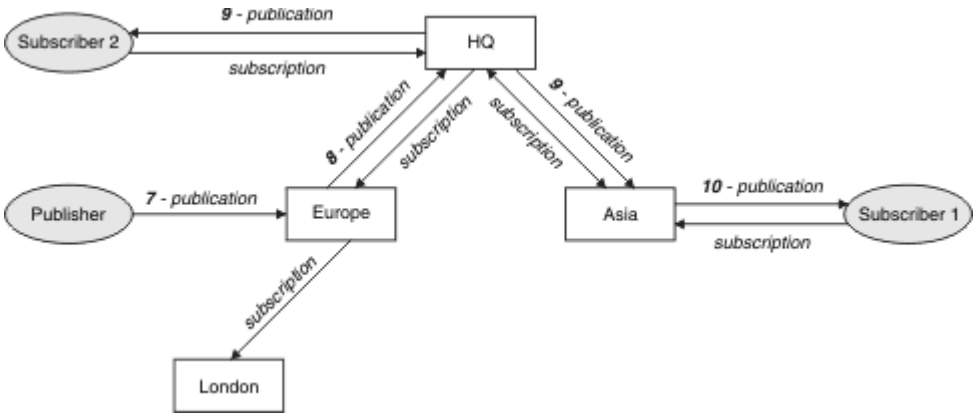


Figura 31. Propagazione delle pubblicazioni tramite una rete di gestori code

Quando un gestore code invia pubblicazioni o sottoscrizioni a un altro gestore code, imposta il proprio ID utente nel messaggio. Se si sta utilizzando una gerarchia di pubblicazione / sottoscrizione e se il canale in entrata è configurato per inserire i messaggi con l'autorizzazione dell'ID utente nel messaggio, è necessario autorizzare l'ID utente del gestore code di invio. Consultare Utilizzo degli ID utente predefiniti con una gerarchia di gestori code.

Nota: Se si utilizzano invece i cluster di pubblicazione / sottoscrizione, l'autorizzazione viene gestita dal cluster.

Sintesi e considerazioni aggiuntive

Una gerarchia di pubblicazione / sottoscrizione fornisce un controllo preciso sulla relazione tra i gestori code. Dopo che è stato creato, ha bisogno di un piccolo intervento manuale da amministrare. Tuttavia, impone anche alcuni vincoli al sistema:

- I nodi superiori nella gerarchia, in particolare il nodo root, devono essere ospitati su apparecchiature robuste, altamente disponibili e performanti. Ciò è dovuto al fatto che si prevede che un maggiore traffico di pubblicazione scorrerà attraverso questi nodi.
- La disponibilità di ogni gestore code non foglia nella gerarchia influenza la capacità della rete di trasmettere i messaggi dai publisher ai sottoscrittori su altri gestori code.
- Per impostazione predefinita, tutte le stringhe argomento sottoscritte vengono propagate in tutta la gerarchia e le pubblicazioni vengono propagate solo ai gestori code remoti che hanno una sottoscrizione all'argomento associato. Pertanto, le rapide modifiche all'insieme delle sottoscrizioni possono diventare un fattore limitante. È possibile modificare questo comportamento predefinito e fare in modo che tutte le pubblicazioni vengano propagate a tutti i gestori code, il che elimina la necessità di sottoscrizioni proxy. Vedere Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione.

  Nota: Una restrizione simile si applica anche ai cluster instradati direttamente.
- A causa della natura interconnessa dei gestori code di pubblicazione / sottoscrizione, la propagazione delle sottoscrizioni proxy su tutti i nodi nella rete richiede tempo. Le pubblicazioni remote non iniziano necessariamente ad essere sottoscritte immediatamente, quindi le pubblicazioni iniziali potrebbero non essere inviate in seguito ad una sottoscrizione ad una nuova stringa di argomenti. È possibile rimuovere i problemi causati dal ritardo della sottoscrizione facendo in modo che tutte le pubblicazioni vengano propagate a tutti i gestori code, eliminando la necessità di sottoscrizioni proxy. Consultare Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione.

  Nota: Questa limitazione si applica anche ai cluster instradati direttamente.
- Per una gerarchia di pubblicazione / sottoscrizione, l'aggiunta o la rimozione di gestori code richiede la configurazione manuale della gerarchia, con un'attenta considerazione dell'ubicazione di tali gestori code e della loro dipendenza da altri gestori code. A meno che non si stiano aggiungendo o rimuovendo

gestori code che si trovano nella parte inferiore della gerarchia e pertanto non vi siano ulteriori rami al di sotto di essi, sarà necessario configurare anche altri gestori code nella gerarchia.

Prima di utilizzare una gerarchia di pubblicazione / sottoscrizione come meccanismo di instradamento, esplorare gli approcci alternativi descritti in “Instradamento diretto nei cluster di pubblicazione / sottoscrizione” a pagina 78 e “Instradamento host argomento nei cluster di pubblicazione / sottoscrizione” a pagina 83.

Code di sistema di pubblicazione / sottoscrizione distribuite

Quattro code di sistema vengono utilizzate dai gestori code per la messaggistica di pubblicazione / sottoscrizione. È necessario essere consapevoli della loro esistenza solo per scopi di determinazione dei problemi e di pianificazione della capacità.

Consultare Bilanciamento di produttori e consumatori nelle reti di pubblicazione / sottoscrizione per istruzioni su come monitorare queste code.

Tabella 6. Code di sistema di pubblicazione / sottoscrizione su Multiplatforms

Coda di sistema	**Finalità**
SYSTEM.INTER.QMGR.CONTROL	Coda di controllo di pubblicazione / sottoscrizione distribuita IBM MQ
SYSTEM.INTER.QMGR.FANREQ	Coda di input del processo fan - out della sottoscrizione proxy interno di pubblicazione / sottoscrizione distribuita IBM MQ
SYSTEM.INTER.QMGR.PUBS	Pubblicazioni di pubblicazione / sottoscrizione distribuite IBM MQ
SYSTEM.HIERARCHY.STATE	IBM MQ stato della relazione della gerarchia di pubblicazione / sottoscrizione distribuita

z/OS Su z/OS, si impostano gli oggetti di sistema necessari quando si crea il gestore code, includendo gli esempi CSQ4INSX, CSQ4INSR e CSQ4INSG nel dataset di input di inizializzazione CSQINP2 . Per ulteriori informazioni, consultare Attività 13: personalizzazione dei dataset di input di inizializzazione.

Gli attributi delle code di sistema di pubblicazione / sottoscrizione sono mostrati in Tabella 7 a pagina 111.

Tabella 7. Attributi delle code di sistema di pubblicazione / sottoscrizione

Attributo	**Valore predefinito**
DEFPSIST	Sì
DEFSOPT	CONDIVISO
MAXMSGL	Multi Su Multipiattaforme: il valore del parametro MAXMSGL del comando ALTER QMGR z/OS Su z/OS: 4194304 (ovvero, 4 MB)
MAXDEPTH	999999999
SHARE	N/A
z/OS z/OS STGCLASS	Questo attributo viene utilizzato solo su piattaforme z/OS

Nota: L'unica coda che contiene i messaggi immessi dalle applicazioni è SYSTEM.INTER.QMGR.PUBS. **MAXDEPTH** è impostato sul relativo valore massimo per questa coda per consentire la creazione temporanea di messaggi pubblicati durante le interruzioni o i tempi di carico eccessivo. Se il gestore code è in esecuzione su un sistema in cui non è stato possibile contenere tale profondità di coda, è necessario modificarla.

Attività correlate

Risoluzione dei problemi di pubblicazione / sottoscrizione distribuiti

Errori della coda di sistema di pubblicazione / sottoscrizione distribuita

Gli errori possono verificarsi quando le code del gestore code di pubblicazione / sottoscrizione distribuite non sono disponibili. Ciò influisce sulla diffusione della conoscenza della sottoscrizione nella rete di pubblicazione / sottoscrizione e sulla pubblicazione delle sottoscrizioni sui gestori code remoti.

Se la coda di richieste di fan - out SYSTEM.INTER.QMGR.FANREQ non è disponibile, la creazione di una sottoscrizione potrebbe generare un errore e i messaggi di errore verranno scritti nel log di errori del gestore code quando le sottoscrizioni proxy devono essere consegnate ai gestori code connessi direttamente.

Se la coda di stato della relazione della gerarchia SYSTEM.HIERARCHY.STATE non è disponibile, viene scritto un messaggio di errore nel log degli errori del gestore code e il motore di pubblicazione / sottoscrizione viene messo in modalità COMPAT . Per visualizzare la modalità di pubblicazione / sottoscrizione, utilizzare il comando DISPLAY QMGR PSMODE.

Se altre code SYSTEM.INTER.QMGR non sono disponibili, viene scritto un messaggio di errore nel log degli errori del gestore code e, sebbene la funzione non sia disabilitata, è probabile che i messaggi di pubblicazione / sottoscrizione si accumulino sulle code su questo o su gestori code remoti.

Se la coda del sistema di pubblicazione / sottoscrizione o la coda di trasmissione richiesta a un gestore code del cluster principale, secondario o di pubblicazione / sottoscrizione non è disponibile, si verificano i seguenti risultati:

- Le pubblicazioni non vengono consegnate e un'applicazione di pubblicazione potrebbe ricevere un errore. Per informazioni dettagliate su quando l'applicazione di pubblicazione riceve un errore, consultare i seguenti parametri del comando **DEFINE TOPIC** : **PMSGDLV** , **NPMSGDLV** e **USEDLQ** .
- Le pubblicazioni tra gestori code ricevute vengono sottoposte a backout nella coda di input e successivamente ritentate. Se viene raggiunta la soglia di backout, le pubblicazioni non consegnate vengono inserite nella coda dei messaggi non recapitabili. Il log degli errori del gestore code conterrà i dettagli del problema.
- Viene eseguito il backout di una sottoscrizione proxy non consegnata nella coda di richiesta fanout e successivamente un nuovo tentativo. Se viene raggiunta la soglia di backout, la sottoscrizione proxy non consegnata non viene consegnata ad alcun gestore code connesso e viene collocata nella coda di messaggi non recapitabili. Il log degli errori del gestore code conterrà i dettagli del problema, inclusi i dettagli di qualsiasi azione di gestione correttiva necessaria richiesta.
- I messaggi del protocollo di relazione della gerarchia hanno esito negativo e lo stato della connessione è contrassegnato come ERROR. Per visualizzare lo stato della connessione, utilizzare il comando **DISPLAY PUBSUB**.

Attività correlate

Risoluzione dei problemi di pubblicazione / sottoscrizione distribuiti

Multi Pianificazione dei requisiti di storage e prestazioni su Multiplatforms

È necessario impostare un archivio realistico e raggiungibile e obiettivi di prestazioni per il proprio sistema IBM MQ . Utilizzare i link per informazioni sui fattori che influenzano l'archiviazione e le prestazioni sulla piattaforma.

I requisiti variano in base ai sistemi su cui si utilizza IBM MQ e ai componenti che si desidera utilizzare.

Per le informazioni più recenti sugli ambienti hardware e software supportati, consultare Requisiti di sistema per IBM MQ.

IBM MQ memorizza i dati del gestore code nel filesystem. Utilizzare i seguenti collegamenti per informazioni sulla pianificazione e la configurazione delle strutture di directory da utilizzare con IBM MQ:

- “Pianificazione del supporto del file system su Multiplatforms” a pagina 115
- “Requisiti per i file system condivisi su Multiplatforms” a pagina 116
- “Condivisione di file IBM MQ su Multiplatforms” a pagina 126
- Linux AIX “Struttura di directory su sistemi AIX and Linux .” a pagina 128
- Windows “Struttura di directory su sistemi Windows .” a pagina 137
- IBM i “Struttura di directory su IBM i” a pagina 141

Utilizzare i collegamenti seguenti per informazioni relative alle risorse di sistema, alla memoria condivisa e alla priorità del processo su AIX and Linux:

- Linux AIX “Risorse IPC IBM MQ e UNIX System V” a pagina 145
- AIX “Memoria condivisa su AIX” a pagina 144
- Linux AIX “Priorità processo IBM MQ e UNIX” a pagina 145

Utilizzare i seguenti collegamenti per informazioni sui file di log:

- “Scelta della registrazione circolare o lineare su Multiplatforms” a pagina 144
- Calcolo della dimensione del log

Concetti correlati

“Planning your IBM MQ environment on z/OS” a pagina 145
When planning your IBM MQ environment, you must consider the resource requirements for data sets, page sets, Db2, Coupling Facilities, and the need for logging, and backup facilities. Use this topic to plan the environment where IBM MQ runs.

Attività correlate

“Pianificazione di un'architettura IBM MQ” a pagina 5
Quando si pianifica l'ambiente IBM MQ , considerare il supporto fornito da IBM MQ per le architetture di gestori code singoli e multipli e per gli stili di messaggistica point-to-point e di pubblicazione / sottoscrizione. Inoltre, pianificare i requisiti delle risorse e l'utilizzo delle funzioni di registrazione e backup.

Riferimenti correlati

Requisiti hardware e software su AIX and Linux
Requisiti hardware e software su Windows

Multi Requisiti di spazio su disco su Multiplatforms

I requisiti di memoria per IBM MQ dipendono dai componenti che si installano e dallo spazio di lavoro necessario.

L'archiviazione su disco è richiesta per i componenti facoltativi che si sceglie di installare, inclusi i componenti prerequisiti richiesti. Il requisito di memoria totale dipende anche dal numero di code utilizzate, dal numero e dalla dimensione dei messaggi nelle code e se i messaggi sono persistenti. È inoltre necessaria la capacità di archiviazione su disco, nastro o altri supporti, nonché lo spazio per i propri programmi applicativi.

Le seguenti tabelle mostrano lo spazio su disco approssimativo richiesto quando si installano varie combinazioni del prodotto su piattaforme differenti. (I valori vengono arrotondati per eccesso ai 5 MB più vicini, dove un MB è 1.048.576 byte.)

- LTS “Requisiti di spazio su disco per Long Term Support” a pagina 114
- CD “Requisiti di spazio su disco per Continuous Delivery” a pagina 115

Requisiti di spazio su disco per Long Term Support

LTS V 9.4.0

Tabella 8. IBM MQ requisiti di spazio su disco per Multiplatforms for Long Term Support			
Piattaforma	**Installazione client** “1” a pagina 114	**Installazione server** “2” a pagina 114	**Installazione completa** “3” a pagina 114
AIX AIX	335 MB	375 MB	1810 MB
IBM i IBM i (consultare Note aggiuntive per IBM i)	485 MB	845 MB	1965 MB
Linux Linux for x86-64	270 MB	295 MB	2010 MB
Linux Linux on Power Systems - Little Endian	170 MB	190 MB	1400 MB
Linux Linux for IBM Z	255 MB	290 MB	1485 MB
Windows Windows (installazione a 64 bit) “4” a pagina 114	295 MB	425 MB	2310 MB

Note:

1. Un'installazione client include i seguenti componenti:
   - Runtime
   - Client
2. Un'installazione server include i componenti seguenti:
   - Runtime
   - Server
3. Un'installazione completa include tutti i componenti disponibili.
4. Windows Non tutti i componenti qui elencati sono funzioni installabili su sistemi Windows ; la loro funzionalità è talvolta inclusa in altre funzioni. Vedere IBM MQ funzioni per i sistemi Windows.

Ulteriori note per IBM i: IBM i

1. Su IBM i non è possibile separare il client nativo dal server. La figura del server nella tabella è per 5724H72*BASE senza Java, insieme al caricamento in lingua inglese (2924). Ci sono 22 possibili carichi di lingua univoci.
2. La figura nella tabella è per il client nativo 5725A49 *BASE senza Java.
3. Le classi Java e JMS possono essere aggiunte sia ai bind server che client. Se si desidera includere queste funzioni aggiungere 110 MB.
4. L'aggiunta dell'origine degli esempi al client o al server aggiunge altri 10 MB.

5. L'aggiunta di esempi alle classi Java e JMS aggiunge ulteriori 5 MB.

Requisiti di spazio su disco per Continuous Delivery

V 9.4.0 CD

Tabella 9. IBM MQ requisiti di spazio su disco per Multiplatforms for Continuous Delivery			
Release Platform/CD	**Installazione client** **"1" a pagina 115**	**Installazione server** **"2" a pagina 115**	**Installazione completa** **"3" a pagina 115**
AIX **AIX**			
V 9.4.0 IBM MQ 9.4.0	355 MB	390 MB	1440 MB
Linux **Linux per x86-64 (64 bit)**			
V 9.4.0 IBM MQ 9.4.0	280 MB	295 MB	1195 MB
Linux **Linux on Power Systems - Little Endian**			
V 9.4.0 IBM MQ 9.4.0	170 MB	195 MB	1075 MB
Linux **Linux per IBM Z**			
V 9.4.0 IBM MQ 9.4.0	260 MB	290 MB	1160 MB
Windows **Windows (installazione a 64 bit)** "4" a pagina 115			
V 9.4.0 IBM MQ 9.4.0	300 MB	425 MB	1785 MB

Note:

1. Un'installazione client include i seguenti componenti:
   - Runtime
   - Client
2. Un'installazione server include i componenti seguenti:
   - Runtime
   - Server
3. Un'installazione completa include tutti i componenti disponibili.
4. Windows Non tutti i componenti qui elencati sono funzioni installabili su sistemi Windows ; la loro funzionalità è talvolta inclusa in altre funzioni. Vedere IBM MQ funzioni per i sistemi Windows.

Concetti correlati

Componenti e funzioni di IBM MQ

Multi Pianificazione del supporto del file system su Multiplatforms

I dati del gestore code vengono memorizzati nel file system. Un gestore code utilizza il blocco del file system per impedire che più istanze di un gestore code a più istanze siano attive contemporaneamente.

File system condivisi

I file system condivisi consentono a più sistemi di accedere simultaneamente alla stessa periferica di archiviazione fisica. Si verificherebbe un danneggiamento se più sistemi accedessero direttamente allo stesso dispositivo di archiviazione fisico senza alcun mezzo per applicare il blocco e il controllo della simultaneità. I sistemi operativi forniscono i file system locali con il controllo del blocco e della simultaneità per i processi locali; i file system di rete forniscono il controllo del blocco e della simultaneità per i sistemi distribuiti.

Storicamente, i file system di rete non sono stati eseguiti abbastanza velocemente, o hanno fornito un sufficiente controllo di blocco e simultaneità, per soddisfare i requisiti per la registrazione dei messaggi. Oggi, i file system in rete possono fornire buone prestazioni e implementazioni di protocolli di file system di rete affidabili come *RFC 3530, Network File System (NFS) version 4 protocol*, soddisfano i requisiti per la registrazione dei messaggi in modo affidabile.

File system condivisi e IBM MQ

I dati del gestore code per un gestore code a più istanze sono memorizzati in un file system di rete condiviso. Sui sistemi AIX, Linux, and Windows , i file di dati e i file di log del gestore code devono essere collocati nel file system di rete condiviso. IBM i Su IBM i, i journal vengono utilizzati al posto dei file di log e i journal non possono essere condivisi. I gestori code a più istanze su IBM i utilizzano la replica del journal o journal commutabili, per rendere i journal disponibili tra diverse istanze del gestore code.

IBM MQ utilizza il blocco per impedire che più istanze dello stesso gestore code a più istanze siano attive contemporaneamente. Lo stesso blocco garantisce inoltre che due gestori code separati non possano inavvertitamente utilizzare la stessa serie di file di dati del gestore code. Solo un'istanza di un gestore code può avere il blocco alla volta. Di conseguenza, IBM MQ supporta i dati del gestore code memorizzati nella memoria di rete a cui si accede come un file system condiviso.

Poiché non tutti i protocolli di blocco dei file system di rete sono solidi e poiché un file system potrebbe essere configurato per le prestazioni piuttosto che per l'integrità dei dati, è necessario eseguire il comando **amqmfsck** per verificare se un file system di rete controllerà correttamente l'accesso ai dati e ai log del gestore code. Questo comando si applica solo ai sistemi UNIX, Linux e IBM i . Su Windows, esiste un solo file system di rete supportato e il comando **amqmfsck** non è obbligatorio.

Attività correlate

“Verifica del funzionamento del file system condiviso su Multiplatforms” a pagina 118
Eseguire **amqmfsck** per controllare se un file system condiviso su AIX, Linux o IBM i soddisfa i requisiti per la memorizzazione dei dati del gestore code di un gestore code a più istanze. (L'unico requisito per una configurazione Windows è che utilizzi SMB 3 per il provisioning di memoria condivisa.)

Multi Requisiti per i file system condivisi su Multiplatforms

I file system condivisi devono fornire l'integrità di scrittura dei dati, garantire l'accesso esclusivo ai file e rilasciare i blocchi in caso di errore per lavorare in modo affidabile con IBM MQ.

Requisiti che un file system condiviso deve soddisfare

Ci sono tre requisiti fondamentali che un file system condiviso deve soddisfare per funzionare in modo affidabile con IBM MQ:

1. Integrità di scrittura dei dati

   L'integrità di scrittura dei dati è a volte denominata *Write through to disk on flush*. Il gestore code deve essere in grado di sincronizzarsi con i dati di cui è stato eseguito correttamente il commit sul dispositivo fisico. In un sistema transazionale, è necessario essere certi che alcune scritture siano state sottoposte a commit in modo sicuro prima di continuare con altre elaborazioni.

   Più specificamente, le piattaforme IBM MQ for AIX or Linux utilizzano l'opzione di apertura *O_SYNC* e la chiamata di sistema `fsync()` per forzare esplicitamente le scritture sul supporto ripristinabile e l'operazione di scrittura dipende dal corretto funzionamento di queste opzioni.

Attenzione: Linux È necessario montare il filesystem con l'opzione `async` , che supporta ancora l'opzione di scritture sincrone e fornisce prestazioni migliori rispetto all'opzione `sync` .

Tenere presente, tuttavia, che se il file system è stato esportato da Linux, è comunque necessario esportare il file system utilizzando l'opzione `sync` .

2. Accesso esclusivo garantito ai file

   Per sincronizzare più gestori code, è necessario che un gestore code ottenga un blocco esclusivo su un file.

3. Rilascia i blocchi in caso di errore

   Se un gestore code ha esito negativo o se si verifica un errore di comunicazione con il file system, i file bloccati dal gestore code devono essere sbloccati e resi disponibili ad altri processi senza attendere che il gestore code venga riconnesso al file system.

Un file system condiviso deve soddisfare questi requisiti affinché IBM MQ possa funzionare in modo affidabile. In caso contrario, i log e i dati del gestore code vengono danneggiati quando si utilizza il file system condiviso in una configurazione del gestore code a più istanze.

Per i gestori code a più istanze su Microsoft Windows, la memoria di rete deve essere acceduta dal protocollo SMB (Server Message Block) utilizzato dalle reti Microsoft Windows . Il client SMB (Server Message Block) non soddisfa i requisiti IBM MQ per bloccare la semantica su piattaforme diverse da Microsoft Windows, quindi i gestori code a più istanze in esecuzione su piattaforme diverse da Microsoft Windows non devono utilizzare SMB (Server Message Block) come file system condiviso.

Per i gestori code a più istanze su altre piattaforme supportate, l'archiviazione deve essere acceduta da un protocollo del file system di rete compatibile con Posix e che supporta il blocco basato sul lease. Network File System 4 soddisfa questo requisito. I file system più vecchi, come Network File System Versione 3, che non dispongono di un meccanismo affidabile per rilasciare i blocchi dopo un malfunzionamento, non devono essere utilizzati con i gestori code a più istanze.

Verifica se il file system condiviso soddisfa i requisiti

È necessario verificare se il file system condiviso che si intende utilizzare soddisfa questi requisiti. È inoltre necessario verificare se il filesystem è configurato correttamente per l'affidabilità. I file system condivisi a volte forniscono opzioni di configurazione per migliorare le prestazioni a scapito dell'affidabilità.

Per ulteriori informazioni, consultare Verifica dell'istruzione per i file system del gestore code a più istanze IBM MQ.

In circostanze normali, IBM MQ funziona correttamente con la memorizzazione nella cache dell'attributo e non è necessario disabilitare la memorizzazione nella cache, ad esempio impostando NOAC su un montaggio NFS . La memorizzazione nella cache degli attributi può causare problemi quando più client del file system si contendono l'accesso in scrittura allo stesso file sul server del file system, in quanto gli attributi memorizzati nella cache utilizzati da ciascun client potrebbero non essere gli stessi degli attributi sul server. Un esempio di file a cui si accede in questo modo sono i log degli errori del gestore code per un gestore code a più istanze. I log degli errori del gestore code potrebbero essere scritti sia da un'istanza del gestore code attiva che da un'istanza del gestore code in standby e gli attributi dei file memorizzati nella cache potrebbero causare una crescita dei log degli errori superiore a quella prevista, prima che si verifichi il rollover dei file.

Per un ausilio nel controllo del file system, eseguire l'attività Verifica del funzionamento del file system condiviso. Questa attività controlla se il file system condiviso soddisfa i requisiti 2 e 3. È necessario verificare il requisito 1 nella documentazione del file system condiviso o sperimentando la registrazione dei dati sul disco.

Gli errori del disco possono causare errori durante la scrittura su disco, che IBM MQ riporta come errori FFDC (First Failure Data Capture). È possibile eseguire il programma di controllo del file system per

il proprio sistema operativo per controllare il file system condiviso per eventuali errori del disco. Ad esempio:

- Linux AIX Su AIX and Linux il programma di controllo del file system è denominato fsck.
- Windows Su piattaforme Windows , il programma di controllo del file system è denominato CHKDSK o SCANDISK.

Sicurezza server NFS

Note:

- Non è possibile utilizzare le opzioni **nosuid** o **noexec** per un punto di montaggio utilizzato per contenere la directory di installazione di IBM MQ . Ciò è dovuto al fatto che IBM MQ include programmi eseguibili setuid / setgid e a questi non deve essere impedita la corretta esecuzione.
- Quando si inseriscono i dati del gestore code solo su un server NFS (Network File System) (NFS), è possibile utilizzare le seguenti tre opzioni con il comando mount per rendere il sistema sicuro, senza alcun impatto dannoso sull'esecuzione del gestore code:

  noexec
  Utilizzando questa opzione, si impedisce l'esecuzione dei file binari su NFS, il che impedisce a un utente remoto di eseguire codice indesiderato sul sistema.

  nosuid
  Utilizzando questa opzione, si impedisce l'utilizzo dei bit set - user - identifier e set - group - identifier, che impediscono a un utente remoto di ottenere privilegi più elevati.

  nessun dev
  Utilizzando questa opzione, si arrestano i caratteri e si bloccano i dispositivi speciali da utilizzare o definire, il che impedisce a un utente remoto di uscire da una prigione chroot.

IBM i Linux AIX *Verifica del funzionamento del file system condiviso su Multiplatforms*

Eseguire **amqmfsck** per controllare se un file system condiviso su AIX, Linux o IBM i soddisfa i requisiti per la memorizzazione dei dati del gestore code di un gestore code a più istanze. (L'unico requisito per una configurazione Windows è che utilizzi SMB 3 per il provisioning di memoria condivisa.)

Prima di iniziare

È necessario un server con memoria di rete e altri due server connessi ad esso che hanno IBM MQ installato. È necessario disporre dell'autorizzazione di amministratore (root) per la configurazione del file system ed essere un IBM MQ amministratore per eseguire **amqmfsck**.

Informazioni su questa attività

“Requisiti per i file system condivisi su Multiplatforms” a pagina 116 descrive i requisiti del filesystem per l'utilizzo di un filesystem condiviso con gestori code a più istanze. La IBM MQ nota tecnica Istruzione di test per i file system del gestore code a più istanze IBM MQ elenca i file system condivisi con cui IBM ha già eseguito il test. La procedura in questa attività descrive come eseguire il test di un file system per valutare se un file system non elencato conserva l'integrità dei dati.

Il failover di un gestore code a più istanze può essere attivato da errori hardware o software, inclusi problemi di rete che impediscono al gestore code di scrivere nei propri dati o file di log. Principalmente, si è interessati a causare errori sul server di file. Ma è anche necessario far sì che i server IBM MQ abbiano esito negativo, per verificare che tutti i blocchi siano stati rilasciati correttamente. Per essere sicuri in un file system condiviso, verificare tutti i seguenti errori e tutti gli altri errori specifici del proprio ambiente:

1. Chiusura del sistema operativo sul file server inclusa la sincronizzazione dei dischi.
2. Arresto del sistema operativo sul file server senza sincronizzazione dei dischi.

3. Premendo il pulsante di reimpostazione su ciascuno dei server.
4. Estrarre il cavo di rete da ciascuno dei server.
5. Estrarre il cavo di alimentazione da ciascun server.
6. Disattivare ciascuno dei server.

Creare la directory sulla memoria di rete che si intende utilizzare per condividere i dati e i log del gestore code. Il proprietario della directory deve essere un Amministratore IBM MQ o, in altre parole, un membro del gruppo `mqm` su AIX and Linux. L'utente che esegue i test deve disporre dell'autorizzazione di amministratore IBM MQ .

Utilizzare l'esempio di esportazione e montaggio di un filesystem in Creazione di un gestore code a più istanze su Linux o Creazione di un gestore code a più istanze utilizzando il mirroring del journal e NetServer su IBM i per facilitare la configurazione del filesystem. File system diversi richiedono diversi passi di configurazione. Leggere la documentazione del file system.

Nota: Eseguire il IBM MQ MQI client programma di esempio **amqsfhac** in parallelo con **amqmfsck** per dimostrare che un gestore code conserva l'integrità del messaggio durante un malfunzionamento.

Procedura

In ogni controllo, causare tutti gli errori nell'elenco precedente mentre il programma di controllo del file system è in esecuzione. Se si intende eseguire **amqsfhac** contemporaneamente a **amqmfsck**, eseguire l'attività “Esecuzione di amqsfhac per verificare l'integrità del messaggio” a pagina 124 in parallelo con questa attività.

1. Montare la directory esportata sui due server IBM MQ .

   Sul server di file system creare una directory condivisa `share`de una sottodirectory per salvare i dati per i gestori code a più istanze, `qmdata`. Per un esempio di impostazione di una directory condivisa per i gestori code a più istanze su Linux, consultare Creazione di un gestore code a più istanze su Linux
2. Controllare il funzionamento del file system di base.

   Su un server IBM MQ , eseguire il programma di controllo del file system senza parametri.

   Sul server IBM MQ 1:

   ```
   amqmfsck /shared/qmdata
   ```
3. Controllare la scrittura simultanea nella stessa directory da entrambi i server IBM MQ .

   Su due server IBM MQ , eseguire il programma di controllo del file system contemporaneamente con l'opzione `-c` .

   Sul server IBM MQ 1:

   ```
   amqmfsck -c /shared/qmdata
   ```

   Sul server IBM MQ 2:

   ```
   amqmfsck -c /shared/qmdata
   ```
4. Verificare di attendere e rilasciare i blocchi su entrambi i server IBM MQ .

   Su entrambi i server IBM MQ eseguire il programma di controllo del filesystem contemporaneamente con l'opzione `-w` .

   Sul server IBM MQ 1:

   ```
   amqmfsck -w /shared/qmdata
   ```

Sul server IBM MQ 2:

```
amqmfsck -w /shared/qmdata
```

5. Verificare l'integrità dei dati.

   a) Formattare il file di test.

   Creare un file di grandi dimensioni nella directory che si sta verificando. Il file viene formattato in modo che le fasi successive possano essere completate correttamente. Il file deve essere abbastanza grande da avere tempo sufficiente per interrompere la seconda fase per simulare il failover. Provare il valore predefinito di 262144 pagine (1 GB). Il programma riduce automaticamente questo valore predefinito sui filesystem lenti in modo che la formattazione venga completata in circa 60 secondi

   Sul server IBM MQ 1:

   ```
   amqmfsck -f /shared/qmdata
   ```

   Il server risponde con i seguenti messaggi:

   ```
   Formatting test file for data integrity test.

   Test file formatted with 262144 pages of data.
   ```

   b) Scrivere i dati nel file di test utilizzando il programma di controllo del file system durante la causa di un errore.

   Eseguire il programma di test su due server contemporaneamente. Avviare il programma di test sul server su cui si verificherà l'errore, quindi avviare il programma di test sul server che sopravviverà all'errore. Causa dell'errore che si sta analizzando.

   Il primo programma di test viene arrestato con un messaggio di errore. Il secondo programma di test ottiene il blocco sul file di test e scrive i dati nel file di test a partire dal punto in cui il primo programma di test si è spento. Consentire il completamento del secondo programma di test.

Tabella 10. Esecuzione del controllo di integrità dei dati su due server contemporaneamente

IBM MQ Server 1	**IBM MQ Server 2**
`amqmfsck -a /shared/qmdata`	

Tabella 10. Esecuzione del controllo di integrità dei dati su due server contemporaneamente (Continua)

IBM MQ Server 1	IBM MQ Server 2

`Please start this program on a second machine`
`with the same parameters.`
 	
`File lock acquired.`
 	
`Start a second copy of this program with the same parameters on another server.`
 	
`Writing data into test file.`
 	
`To increase the effectiveness of the test, interrupt the writing by ending the process,`
`temporarily breaking the network connection`
`to the networked storage,`
`rebooting the server or turning off the power.`
	`amqmfsck -a /shared/qmdata`
	
	`Waiting for lock...`
	
	`Waiting for lock...`
	
	`Waiting for lock...`
	
	`Waiting for lock...`
	
	`Waiting for lock...`
	
	`Waiting for lock...`

Turn the power off here.	

	`File lock acquired.`
	
	`Reading test file`
	
	`Checking the integrity of the data read.`
	
	`Appending data into the test file`
	`after data already found.`
	
	`The test file is full of data.`
	`It is ready to be inspected for data integrity.`

La tempistica del test dipende dal comportamento del file system. Ad esempio, generalmente impiega 30-90 secondi perché un file system rilasci i blocchi di file ottenuti dal primo programma dopo un'interruzione di corrente. Se si ha poco tempo per introdurre l'errore prima che il primo programma di test abbia riempito il file, utilizzare l'opzione `-x` di **amqmfsck** per eliminare il file di test. Provare il test dall'inizio con un file di test più grande.

c) Verificare l'integrità dei dati nel file di test.

Sul server IBM MQ 2:

```
amqmfsck -i /shared/qmdata
```

Il server risponde con i seguenti messaggi:

```
File lock acquired
```

```
Reading test file checking the integrity of the data read.

The data read was consistent.

The tests on the directory completed successfully.
```

6. Eliminare i file di test.

   Sul server IBM MQ 2:

```
amqmfsck -x /shared/qmdata
Test files deleted.
```

   Il server risponde con il messaggio:

```
Test files deleted.
```

Risultati

Il programma restituisce un codice di uscita di zero se i test vengono completati correttamente, altrimenti un codice diverso da zero.

Esempi

La prima serie di tre esempi mostra il comando che produce un output minimo.

Test riuscito del blocco file di base su un server

```
> amqmfsck /shared/qmdata
The tests on the directory completed successfully.
```

Test non riuscito del blocco del file di base su un server

```
> amqmfsck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfsck.lck' error '2'.
```

Test di blocco riuscito su due server

Tabella 11. Blocco riuscito su due server

IBM MQ Server 1	IBM MQ Server 2

`> amqmfsck -w /shared/qmdata`
`Please start this program on a second`
`machine with the same parameters.`
`Lock acquired.`
`Press Return`
`or terminate the program to release the lock.`	

	`> amqmfsck -w /shared/qmdata` `Waiting for lock...`
`[ Return pressed ]` `Lock released.`	

	`Lock acquired.`
	`The tests on the directory completed`
	`successfully`

La seconda serie di tre esempi mostra gli stessi comandi utilizzando la modalità dettagliata.

Test riuscito del blocco file di base su un server

```
> amqmfsck -v /shared/qmdata
System call: stat("/shared/qmdata")'
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd1 = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd1, F_SETLK, F_RDLCK)
System call: fd2 = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd2, F_SETLK, F_RDLCK)
System call: close(fd2)
System call: write(fd1)
System call: close(fd1)
The tests on the directory completed successfully.
```

Test non riuscito del blocco del file di base su un server

```
> amqmfsck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_RDLCK)
System call: fdSameFile = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fdSameFile, F_SETLK, F_RDLCK)
System call: close(fdSameFile)
System call: write(fd)
AMQxxxx: Error calling 'write()[2]' on file '/shared/qmdata/amqmfsck.lck', errno 2
(Permission denied).
```

Test di blocco riuscito su due server

Tabella 12. Blocco riuscito su due server - modalità dettagliata

IBM MQ Server 1	IBM MQ Server 2

`> amqmfsck -wv /shared/qmdata`
`Calling 'stat("/shared/qmdata")'`
`Calling 'fd = open("/shared/qmdata/`
`amqmfsck.lkw",`
`O_EXCL | O_CREAT | O_RDWR, 0666)'`
`Calling 'fchmod(fd, 0666)'`
`Calling 'fstat(fd)'`
`Please start this program on a second`
`machine with the same parameters.`
`Calling 'fcntl(fd, F_SETLK, F_WRLCK)'`
`Lock acquired.`
`Press Return`
`or terminate the program to release the lock.`	

	`> amqmfsck -wv /shared/qmdata`
	`Calling 'stat("/shared/qmdata")'`
	`Calling 'fd = open("/shared/qmdata/`
	`amqmfsck.lkw",`
	`O_EXCL | O_CREAT | O_RDWR,0666)'`
	`Calling 'fd = open("/shared/qmdata/amqmfsck.lkw,`
	`O_RDWR, 0666)'`
	`Calling 'fcntl(fd, F_SETLK, F_WRLCK)`
	`'Waiting for lock...`

Tabella 12. Blocco riuscito su due server - modalità dettagliata (Continua)	
IBM MQ Server 1	**IBM MQ Server 2**

`[ Return pressed ]`
`Calling 'close(fd)'`
`Lock released.`	

	`Calling 'fcntl(fd, F_SETLK, F_WRLCK)'`
	`Lock acquired.`
	`The tests on the directory completed successfully`

Riferimenti correlati

Programmi di esempio HA (High Availability)

Multi

Esecuzione di amqsfhac per verificare l'integrità del messaggio

Eseguire il IBM MQ MQI client programma di esempio **amqsfhac** in parallelo con **amqmfsck** per dimostrare che un gestore code conserva l'integrità del messaggio durante un errore.

Prima di iniziare

Sono necessari quattro server per questo test. Due server per il gestore code a più istanze, uno per il file system e uno per l'esecuzione di **amqsfhac** come applicazione IBM MQ MQI client .

Seguire il passo “1” a pagina 119 in “Verifica del funzionamento del file system condiviso su Multiplatforms” a pagina 118 per configurare il file system per un gestore code a più istanze.

Informazioni su questa attività

Il IBM MQ MQI client programma di esempio **amqsfhac** controlla che un gestore code che utilizza la memoria di rete mantenga l'integrità dei dati in seguito a un errore. Eseguire **amqsfhac** in parallelo con **amqmfsck** per dimostrare che un gestore code conserva l'integrità del messaggio durante un errore.

Procedura

1. Creare un gestore code a più istanze su un altro server, `QM1`, utilizzando il file system creato al passo “1” a pagina 119 nella Procedura.

   Consultare Crea un gestore code a più istanze.
2. Avviare il gestore code su entrambi i server rendendolo altamente disponibile.

   Sul server 1:

   ```
   strmqm -x QM1
   ```

   Sul server 2:

   ```
   strmqm -x QM1
   ```
3. Impostare la connessione client per eseguire **amqsfhac**.
   a) Utilizzare la procedura contenuta in *Verifica di un'installazione di IBM MQ* per la piattaforma o le piattaforme che l'azienda utilizza per configurare una connessione client o gli script di esempio in Esempi di client riconnettibili.
   b) Modificare il canale del client in modo che abbia due indirizzi IP, corrispondenti ai due server su cui è in esecuzione `QM1`.

Nello script di esempio, modificare:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME('LOCALHOST(2345)') QMNAME(QM1) REPLACE
```

A:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME('server1(2345),server2(2345)') QMNAME(QM1) REPLACE
```

dove `server1` e `server2` sono i nomi host dei due server e 2345 è la porta su cui è in ascolto il listener del canale. Di solito, il valore predefinito è 1414. È possibile utilizzare 1414 con la configurazione listener predefinita.

4. Creare due code locali su `QM1` per la verifica.
   Eseguire lo script MQSC riportato di seguito:

```
DEFINE QLOCAL(TARGETQ) REPLACE
DEFINE QLOCAL(SIDEQ) REPLACE
```

5. Verifica la configurazione con **amqsfhac**

```
amqsfhac QM1 TARGETQ SIDEQ 2 2 2
```

6. Verificare l'integrit ... del messaggio durante la verifica dell'integrit ... del file system.

   Eseguire **amqsfhac** durante il passo “5” a pagina 120 di “Verifica del funzionamento del file system condiviso su Multiplatforms” a pagina 118.

```
amqsfhac QM1 TARGETQ SIDEQ 10 20 0
```

   Se si arresta l'istanza del gestore code attivo, **amqsfhac** si riconnette all'altra istanza del gestore code una volta che è diventata attiva. Riavviare di nuovo l'istanza del gestore code arrestata, in modo da poter invertire l'errore nella verifica successiva. Sarà probabilmente necessario aumentare il numero di iterazioni in base alla sperimentazione con il proprio ambiente in modo che il programma di test venga eseguito per un tempo sufficiente affinché si verifichi il failover.

Risultati

Un esempio di esecuzione di **amqsfhac** nel passo “6” a pagina 125 viene mostrato nel seguente esempio. In questo esempio, il test ha esito positivo.

```
Sample AMQSFHAC start
qmname = QM1
qname = TARGETQ
sidename = SIDEQ
transize = 10
iterations = 20
verbose = 0
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Resolving MQRC_CALL_INTERRUPTED
MQGET browse side tranid=14 pSideinfo->tranid=14
Resolving to committed
Iteration 7
Iteration 8
Iteration 9
Iteration 10
Iteration 11
Iteration 12
```

```
Iteration 13
Iteration 14
Iteration 15
Iteration 16
Iteration 17
Iteration 18
Iteration 19
Sample AMQSFHAC end
```

Se il test ha rilevato un problema, l'output riporta l'errore. In alcune esecuzioni di test, `MQRC_CALL_INTERRUPTED` potrebbe riportare "`Resolving to backed out`". Non fa alcuna differenza per il risultato. Il risultato dipende dal fatto che la scrittura su disco sia stata sottoposta a commit dall'archivio file di rete prima o dopo che si è verificato l'errore.

Riferimenti correlati

amqmfsck (controllo file system)

Programmi di esempio HA (High Availability)

Multi Condivisione di file IBM MQ su Multiplatforms

Ad alcuni file IBM MQ si accede esclusivamente da un gestore code attivo, mentre altri file sono condivisi.

I file IBM MQ sono suddivisi in file di programma e file di dati. I file di programma sono generalmente installati localmente su ciascun server che esegue IBM MQ. I gestori code condividono l'accesso ai file di dati e alle directory nella directory di dati predefinita. Richiedono l'accesso esclusivo alle proprie strutture di directory del gestore code contenute in ciascuna delle directory `qmgrs` e `log` mostrate in Figura 32 a pagina 126.

Figura 32 a pagina 126 è una vista di alto livello della struttura di directory IBM MQ . Mostra le directory che possono essere condivise tra gestori code e rese remote. I dettagli variano per piattaforma. Le linee tratteggiate indicano percorsi configurabili.

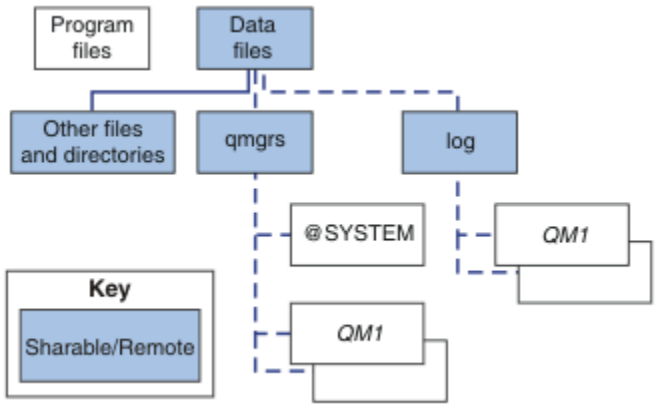


Figura 32. Vista generale della struttura di directory IBM MQ

File di programma

La directory dei file di programma viene generalmente lasciata nell'ubicazione predefinita, è locale e condivisa da tutti i gestori code sul server.

File di dati

La directory dei file di dati è in genere locale nell'ubicazione predefinita, `/var/mqm` sui sistemi AIX and Linux e configurabile sull'installazione su Windows. È condiviso tra gestori code. È possibile rendere remota l'ubicazione predefinita, ma non condividerla tra diverse installazioni di IBM MQ. L'attributo `DefaultPrefix` nella configurazione IBM MQ punta a questo percorso.

qmgrs

Esistono due modi alternativi per specificare l'ubicazione dei dati del gestore code.

Utilizzo dell'attributo Prefix

L'attributo **Prefix** specifica l'ubicazione della directory qmgrs . IBM MQ crea il nome della directory del gestore code dal nome del gestore code e lo crea come sottodirectory della directory qmgrs .

L'attributo **Prefix** si trova nella stanza QueueManager del file mqs.ini ed è ereditato dal valore nell'attributo **DefaultPrefix** della stanza Tutti i gestori code. Per impostazione predefinita, per semplicità amministrativa, i gestori code generalmente condividono la stessa directory qmgrs .

Se si modifica l'ubicazione della directory qmgrs per qualsiasi gestore code, è necessario modificare il valore del suo attributo **Prefix** .

L'attributo **Prefix** per la directory QM1 in Figura 32 a pagina 126 per una piattaforma AIX and Linux è:

```
Prefix=/var/mqm
```

Utilizzo dell'attributo DataPath

L'attributo **DataPath** specifica l'ubicazione della directory di dati del gestore code.

L'attributo **DataPath** specifica il percorso completo, incluso il nome della directory di dati del gestore code. L'attributo **DataPath** è diverso dall'attributo **Prefix** , che specifica un percorso incompleto alla directory dei dati del gestore code.

L'attributo **DataPath** , se specificato, si trova nella stanza stanzaQueueManager del file mqs.ini . Se è stato specificato, ha la precedenza su qualsiasi valore nell'attributo **Prefix** .

Se si modifica l'ubicazione della directory dei dati del gestore code per qualsiasi gestore code, è necessario modificare il valore dell'attributo DataPath .

L'attributo DataPath per la directory QM1 in Figura 32 a pagina 126, per una piattaforma Linux o AIX , è il seguente:

```
DataPath=/var/mqm/qmgrs/QM1
```

log

La directory di log viene specificata separatamente per ogni gestore code nella sezione Log nella configurazione del gestore code. La configurazione del gestore code è in qm.ini.

Sottodirectory *DataPath*/*QmgrName*/@IPCC

Le sottodirectory *DataPath*/*QmgrName*/@IPCC si trovano nel percorso della directory condivisa. Vengono utilizzati per creare il percorso indirizzario per gli oggetti file system IPC. È necessario distinguere lo spazio dei nomi di un gestore code quando un gestore code è condiviso tra sistemi.

Gli oggetti del file system IPC devono essere distinti dal sistema. Una sottodirectory, per ogni sistema su cui viene eseguito il gestore code, viene aggiunta al percorso della directory, consultare Figura 33 a pagina 127.

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

Figura 33. Sottodirectory IPC di esempio

myHostName contiene fino a 20 caratteri del nome host restituito dal sistema operativo. Su alcuni sistemi, il nome host potrebbe avere una lunghezza massima di 64 caratteri prima del troncamento. Il valore generato di *myHostName* potrebbe causare un problema per due motivi:

1. I primi 20 caratteri non sono univoci.
2. Il nome host viene generato da un algoritmo DHCP che non sempre assegna lo stesso nome host a un sistema.

In questi casi, impostare *myHostName* utilizzando la variabile di ambiente **MQS_IPC_HOST**; consultare Figura 34 a pagina 128.

```
export MQS_IPC_HOST= myHostName
```

*Figura 34. Esempio: impostazione **MQS_IPC_HOST***

Altri file e directory

Altri file e directory, come la directory che contiene i file di traccia e il log degli errori comune, vengono normalmente condivisi e conservati sul file system locale.

Con il sostegno dei file system condivisi, IBM MQ gestisce l'accesso esclusivo a questi file utilizzando i blocchi del file system. Un blocco del file system consente di attivare una sola istanza di un particolare gestore code alla volta.

Quando si avvia la prima istanza di uno specifico gestore code, questo assume la proprietà della relativa directory del gestore code. Se si avvia una seconda istanza, questa può assumere la proprietà solo se la prima istanza è stata arrestata. Se il primo gestore code è ancora in esecuzione, la seconda istanza non riesce ad avviarsi e riporta che il gestore code è in esecuzione altrove. Se il primo gestore code è stato arrestato, il secondo gestore code assume la proprietà dei file del gestore code e diventa il gestore code in esecuzione.

È possibile automatizzare la procedura del secondo gestore code che prende il posto del primo gestore code. Avviare il primo gestore code con l'opzione `strmqm -x` che consente a un altro gestore code di eseguire il comando. Il secondo gestore code attende che i file del gestore code vengano sbloccati prima di tentare di acquisire la proprietà dei file del gestore code e di avviarli.

Linux AIX Struttura di directory su sistemi AIX and Linux .

La struttura di directory IBM MQ sui sistemi AIX and Linux può essere associata a diversi file system per una gestione più semplice, migliori prestazioni e una maggiore affidabilità.

Utilizzare la struttura di directory flessibile di IBM MQ per sfruttare i filesystem condivisi per l'esecuzione di gestori code a più istanze.

Utilizzare il comando **crtmqm** *QM1* per creare la struttura di directory mostrata in Figura 35 a pagina 129 dove R è la release del prodotto. Si tratta di una tipica struttura di directory per un gestore code creato su un sistema IBM MQ . Alcune impostazioni di directory, file e attributi .ini vengono omesse per chiarezza e un altro nome gestore code potrebbe essere modificato da un gestore code. I nomi dei file system variano a seconda dei sistemi.

In un'installazione tipica, ogni gestore code creato fa riferimento a directory comuni `log` e `qmgrs` sul file system locale. In una configurazione a più istanze, le indirizzari `log` e `qmgrs` si trovano su un filesystem di rete condiviso con un'altra installazione di IBM MQ.

Figura 35 a pagina 129 mostra la configurazione predefinita per IBM MQ v7.R su AIX , dove R è la release del prodotto. Per esempi di configurazioni alternative a più istanze, vedere“Configurazioni di directory di esempio su sistemi AIX and Linux” a pagina 134.

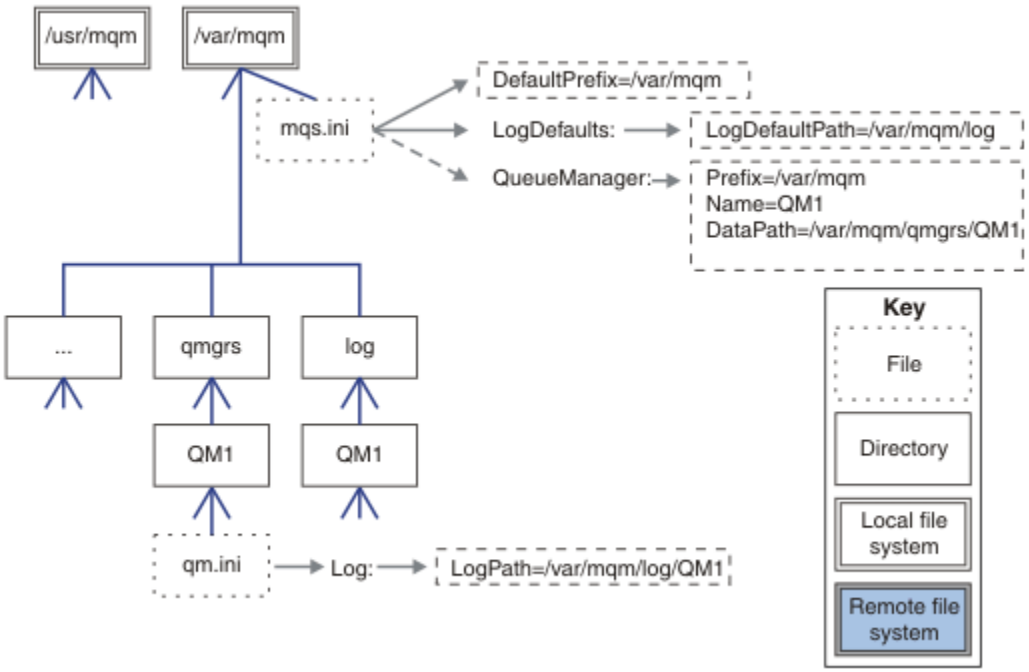


Figura 35. Struttura di directory IBM MQ predefinita di esempio per sistemi AIX and Linux

Per impostazione predefinita, il prodotto è installato in `/usr/mqm` su AIX e `/opt/mqm` sugli altri sistemi. Le directory di lavoro vengono installate nella directory `/var/mqm` .

Nota: Se è stato creato il file system `/var/mqm` prima di installare IBM MQ , assicurarsi che l'utente mqm disponga delle autorizzazioni di directory complete, ad esempio, la modalità file 755.

Nota: La directory `/var/mqm/errors` deve essere un filesystem separato per impedire che FFDC prodotti dal gestore code riempiscano il filesystem che contiene `/var/mqm`.

Per ulteriori informazioni, consultare Creazione di file system su sistemi AIX and Linux .

Le directory `log` e `qmgrs` vengono visualizzate nelle relative ubicazioni predefinite come definite dai valori predefiniti degli attributi `LogDefaultPath` e `DefaultPrefix` nel file `mqs.ini` . Quando viene creato un gestore code, per default la directory dei dati del gestore code viene creata in *`DefaultPrefix`*`/qmgrs`e la directory del file di log in *`LogDefaultPath`*`/log`. `LogDefaultPath` e `DefaultPrefix` hanno effetto solo quando i gestori code e i file di log vengono creati per impostazione predefinita. L'ubicazione effettiva di una directory del gestore code viene salvata nel file `mqs.ini` e l'ubicazione della directory del file di log viene salvata nel file `qm.ini` .

La directory del file di log per un gestore code è definita nel file `qm.ini` nell'attributo `LogPath` . Utilizzare l'opzione `-ld` nel comando **`crtmqm`** per impostare l'attributo `LogPath` per un gestore code; ad esempio **`crtmqm`** `-ld` *`LogPath`* `QM1` . Se si omette il parametro `ld` , viene utilizzato il valore di `LogDefaultPath` .

La directory dei dati del gestore code è definita nell'attributo `DataPath` della stanza `QueueManager` del file `mqs.ini` . Utilizzare l'opzione `-md` sul comando **`crtmqm`** per impostare `DataPath` per un gestore code; ad esempio, **`crtmqm`** `-md` *`DataPath`* `QM1` . Se si omette il parametro `md` , viene utilizzato il valore dell'attributo `DefaultPrefix` o `Prefix` . `Prefisso` ha la precedenza su `DefaultPrefix`.

In genere, creare `QM1` specificando le directory di log e di dati in un singolo comando.

```
 crtmqm
-md DataPath -ld
LogPath QM1
```

È possibile modificare l'ubicazione di un log del gestore code e le directory dei dati di un gestore code esistente modificando gli attributi `DataPath` e `LogPath` nel file `qm.ini` quando il gestore code viene arrestato.

Il percorso della directory `errors` , come i percorsi di tutte le directory in `/var/mqm`, non è modificabile. Tuttavia, le directory possono essere montate su file system differenti o collegate simbolicamente a directory differenti.

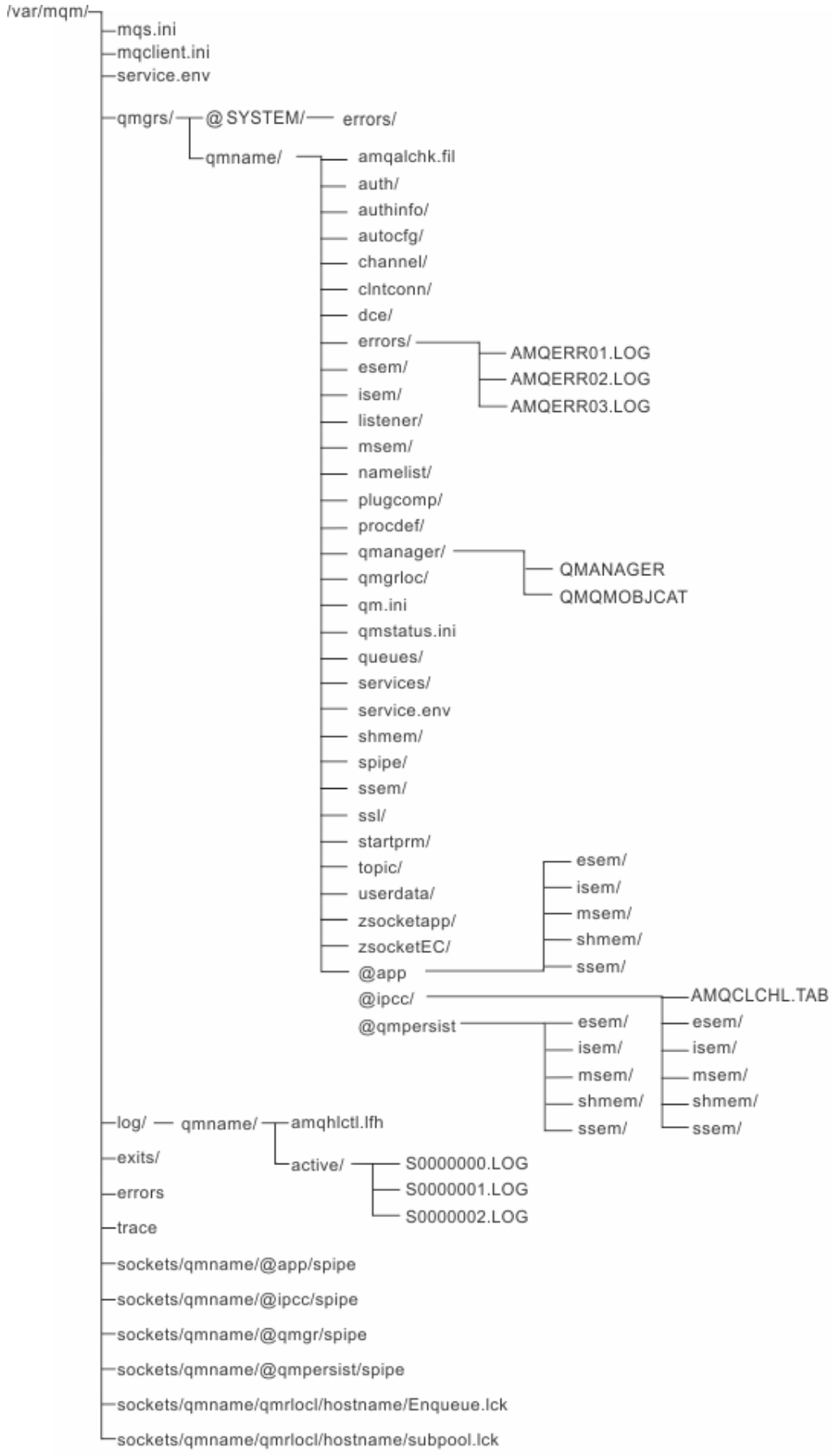
Linux AIX *Contenuto della directory su sistemi AIX and Linux*

Contenuto delle directory associate a un gestore code.

Per informazioni sul percorso dei file del prodotto, consultare Scelta di un percorso di installazione

Per informazioni sulle configurazioni di directory alternative, consultare “Pianificazione del supporto del file system su Multiplatforms” a pagina 115.

La seguente struttura di directory è rappresentativa di IBM MQ dopo che un gestore code è stato in uso per qualche tempo. La struttura effettiva dipende dalle operazioni che si sono verificate sul gestore code.


/var/mqm/
mqs.ini
mqclient.ini
service.env
qmgrs/
@SYSTEM/
errors/
qmname/
amqalchk.fil
auth/
authinfo/
autocfg/
channel/
clntconn/
dce/
errors/
AMQERR01.LOG
AMQERR02.LOG
AMQERR03.LOG
esem/
isem/
listener/
msem/
namelist/
plugcomp/
procdef/
qmanager/
QMANAGER
QMQMOBJCAT
qmgrloc/
qm.ini
qmstatus.ini
queues/
services/
service.env
shmem/
spipe/
ssem/
ssl/
startprm/
topic/
userdata/
zsocketapp/
zsocketEC/
@app
esem/
isem/
msem/
shmem/
ssem/
@ipcc/
AMQCLCHL.TAB
esem/
isem/
msem/
shmem/
ssem/
@qmpersist
esem/
isem/
msem/
shmem/
ssem/
log/
qmname/
amqhlctl.lfh
active/
S0000000.LOG
S0000001.LOG
S0000002.LOG
exits/
errors
trace
sockets/qmname/@app/spipe
sockets/qmname/@ipcc/spipe
sockets/qmname/@qmgr/spipe
sockets/qmname/@qmpersist/spipe
sockets/qmname/qmrlocl/hostname/Enqueue.lck
sockets/qmname/qmrlocl/hostname/subpool.lck

/var/mqm/

La directory /var/mqm contiene i file di configurazione e le directory di output che si applicano a un'installazione di IBM MQ nel suo complesso e non a un singolo gestore code.

Tabella 13. Contenuto documentato della directory /var/mqm su AIX and Linux	
Nome file o directory	**Indice**
mqs.ini	File di configurazione dell'installazione di IBM MQ , letto all'avvio di un gestore code. Percorso file modificabile utilizzando la variabile di ambiente **AMQ_MQS_INI_LOCATION** . Verificare che sia impostato ed esportato nella shell in cui viene eseguito il comando **strmqm** .
mqclient.ini	File di configurazione client predefinito letto dai programmi IBM MQ MQI client . Percorso file modificabile utilizzando la variabile di ambiente **MQCLNTCF** .
service.env	Contiene le variabili di ambiente dell'ambito macchina per un processo di servizio. Percorso file fisso.
errori /	Log degli errori dell'ambito macchina e file FFST . Percorso directory fisso. Consultare anche FFST: IBM MQ for UNIX e Linux systems.
socket /	Contiene informazioni per ogni gestore code solo per il sistema.
traccia /	File di traccia. Percorso directory fisso.
web /	Directory server mqweb.
uscite /	Directory predefinita contenente i programmi di uscita del canale utente. Ubicazione modificabile nelle stanze ApiExit nel file mqs.ini .
exits64/	

/var/mqm/qmgrs/*qmname*/

/var/mqm/qmgrs/*qmname*/ contiene directory e file per un gestore code. La directory è bloccata per l'accesso esclusivo dall'istanza del gestore code attivo. Il percorso della directory è direttamente modificabile nel file mqs.ini oppure utilizzando l'opzione **md** del comando **crtmqm** .

Tabella 14. Contenuto documentato della directory /var/mqm/qmgrs/qmname su AIX and Linux	
Nome file o directory	**Indice**
qm.ini	File di configurazione del gestore code, letto all'avvio del gestore code.
errori /	Log degli errori dell'ambito gestore code. *qmname* = @system contiene i messaggi relativi al canale per un gestore code sconosciuto o non disponibile.

Tabella 14. Contenuto documentato della directory /var/mqm/qmgrs/qmname su AIX and Linux (Continua)	
Nome file o directory	**Indice**
@ipcc/ AMQCLCHL.TAB	Tabella di controllo del canale client predefinita, creata dal server IBM MQ e letta dai programmi IBM MQ MQI client . Percorso file modificabile utilizzando le variabili di ambiente **MQCHLLIB** e **MQCHLTAB** .
QMANAGER	File oggetto gestore code: QMANAGER Catalogo oggetti gestore code: QMQMOBJCAT
authinfo /	Ogni oggetto definito nel gestore code è associato a un file in queste directory. Il nome del file corrisponde approssimativamente al nome della definizione; consultare Descrizione dei nomi file IBM MQ.
canale /	
clntconn /	
listener /	
elenco nomi /	
procdef /	
code /	
servizi /	
argomenti /	
...	Altre directory utilizzate da IBM MQ, come @ipcc, che devono essere modificate solo da IBM MQ.
dati utente /	Può essere utilizzato per memorizzare lo stato persistente delle applicazioni (può essere utilizzato da RDQM quando si spostano i gestori code su nodi diversi - consultare Memorizzazione dello stato dell'applicazione persistente.)
DataPath**autocfg	Utilizzato per la configurazione automatica

/var/mqm/log/*qmname*/

/var/mqm/log/*qmname*/ contiene i file di log del gestore code. La directory è bloccata per l'accesso esclusivo dall'istanza del gestore code attivo. Il percorso della directory è modificabile nel file qm.ini o utilizzando l'opzione **ld** del comando **crtmqm** .

Tabella 15. Contenuto documentato della directory /var/mqm/log/qmname su AIX and Linux	
Nome file o directory	**Indice**
amqhlctl.lfh	File di controllo log.
attivo /	Questa directory contiene i file di log con numero S0000000.LOG, S0000001.LOG, S0000002.LOGe così via.

/opt/mqm

/opt/mqm è, per impostazione predefinita, la directory di installazione sulla maggior parte delle piattaforme. Consultare “Requisiti di spazio su disco su Multiplatforms” a pagina 113 per ulteriori

informazioni sulla quantità di spazio necessario per la directory di installazione sulla piattaforma o sulle piattaforme utilizzate dall'azienda.

Linux AIX *Configurazioni di directory di esempio su sistemi AIX and Linux*

Esempi di configurazioni di file system alternativi su sistemi AIX and Linux .

È possibile personalizzare la struttura di directory IBM MQ in vari modi per raggiungere diversi obiettivi.

- Inserire le directory `qmgrs` e `log` sui file system condivisi remoti per configurare un gestore code a più istanze.
- Utilizzare file system separati per le directory di dati e di log e assegnare le directory a dischi differenti, per migliorare le prestazioni riducendo il conflitto I/O.
- Utilizzare le unità di memoria più veloci per le directory che hanno un effetto maggiore sulle prestazioni. La latenza del dispositivo fisico è spesso un fattore più importante nelle prestazioni della messaggistica persistente rispetto al fatto che un dispositivo sia montato localmente o in remoto. Il seguente elenco mostra quali directory sono più e meno sensibili alle prestazioni.
  1. `log`
  2. `qmgrs`
  3. Altre directory, incluso `/usr/mqm`
- Creare le indirizzari `qmgrs` e `log` sui file system assegnati all'archiviazione con una buona resilienza, ad esempio un disk array ridondante.
- È meglio memorizzare i log degli errori comuni in `var/mqm/errors`, localmente, piuttosto che su un file system di rete, in modo che sia possibile registrare gli errori relativi al file system di rete.

Figura 36 a pagina 134 è un modello da cui derivano strutture di directory IBM MQ alternative. Nel modello, le linee tratteggiate rappresentano percorsi configurabili. Negli esempi, le linee tratteggiate vengono sostituite da linee continue che corrispondono alle informazioni di configurazione memorizzate nella variabile di ambiente AMQ_MQS_INI_LOCATION e nei file `mqs.ini` e `qm.ini` .

Nota: Le informazioni sul percorso vengono mostrate come appaiono nei file `mqs.ini` o `qm.ini` . Se si forniscono parametri di percorso nel comando **`crtmqm`** , omettere il nome della directory del gestore code: il nome del gestore code viene aggiunto al percorso da IBM MQ.

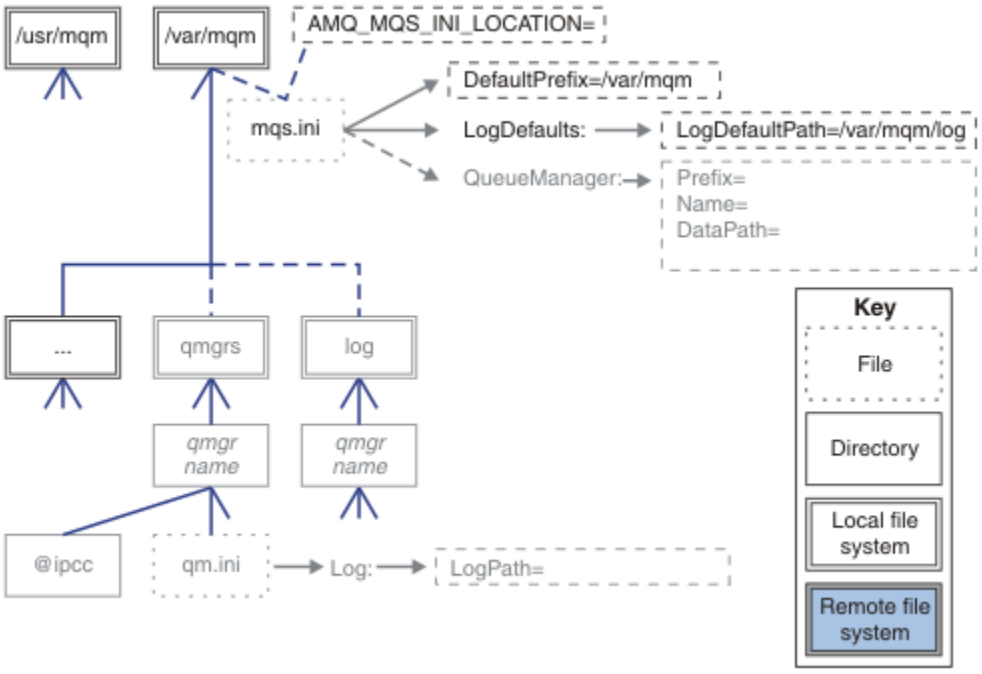


Figura 36. Modello di modello struttura di directory

Struttura di directory tipica per IBM MQ

Figura 37 a pagina 135 è la struttura di directory predefinita creata in IBM MQ con il comando **crtmqm** *QM1*.

Il file `mqs.ini` ha una stanza per il gestore code QM1 , creato facendo riferimento al valore di `DefaultPrefix`. La stanza `Log` nel file `qm.ini` ha un valore per `LogPath`, impostato mediante riferimento a `LogDefaultPath` in `mqs.ini`.

Utilizzare i parametri facoltativi di **crtmqm** per sovrascrivere i valori predefiniti di `DataPath` e `LogPath`.

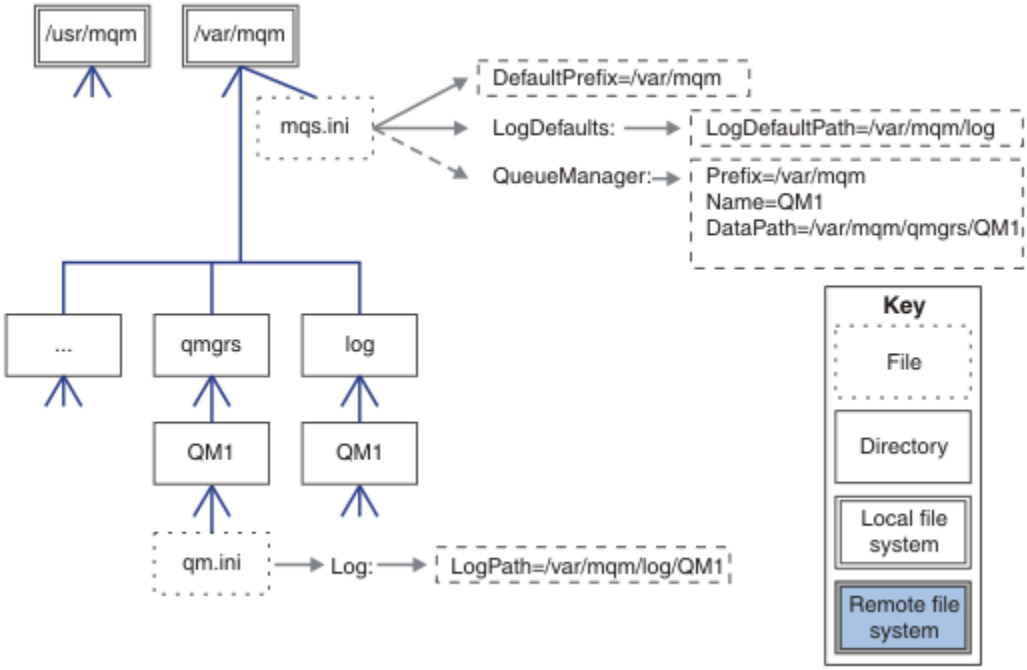


Figura 37. Struttura di directory IBM MQ predefinita di esempio per sistemi AIX and Linux

Condividi directory `qmgrs` e `log` predefinite

Un'alternativa a “Condividi tutto” a pagina 137è quella di condividere separatamente le directory `qmgrs` e `log` (Figura 38 a pagina 136). In questa configurazione, non è necessario impostare `AMQ_MQS_INI_LOCATION` poiché il file mqs.ini predefinito è memorizzato nel file system `/var/mqm` locale. I file e le directory, come `mqclient.ini` e `mqserver.ini` , non vengono condivisi.

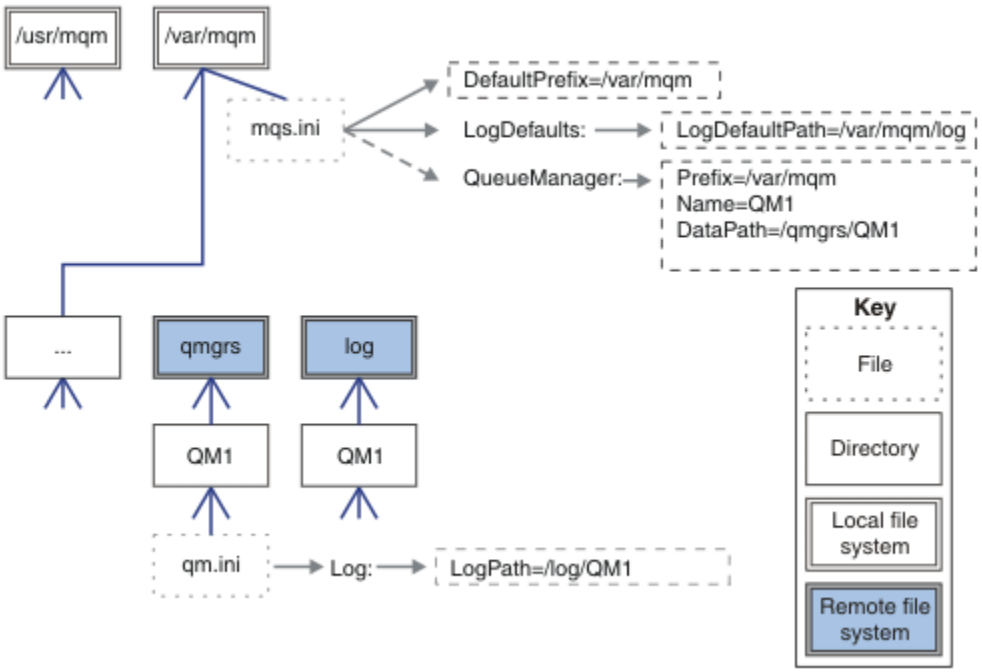


Figura 38. Condividi directory `qmgrs` e `log`

Condividere le directory denominate `qmgrs` e `log`

La configurazione in Figura 39 a pagina 136 colloca `log` e `qmgrs` in un file system condiviso remoto denominato `/ha`. La stessa configurazione fisica può essere creata in due modi diversi.

1. Impostare `LogDefaultPath=/ha` ed eseguire il comando **`crtmqm`** - `md` */ha/qmgrs* `QM1`. Il risultato è esattamente come illustrato in Figura 39 a pagina 136.
2. Lasciare invariati i percorsi predefiniti ed eseguire il comando, **`crtmqm`** - `ld` */ha/log* - `md` */ha/qmgrs* `QM1`.

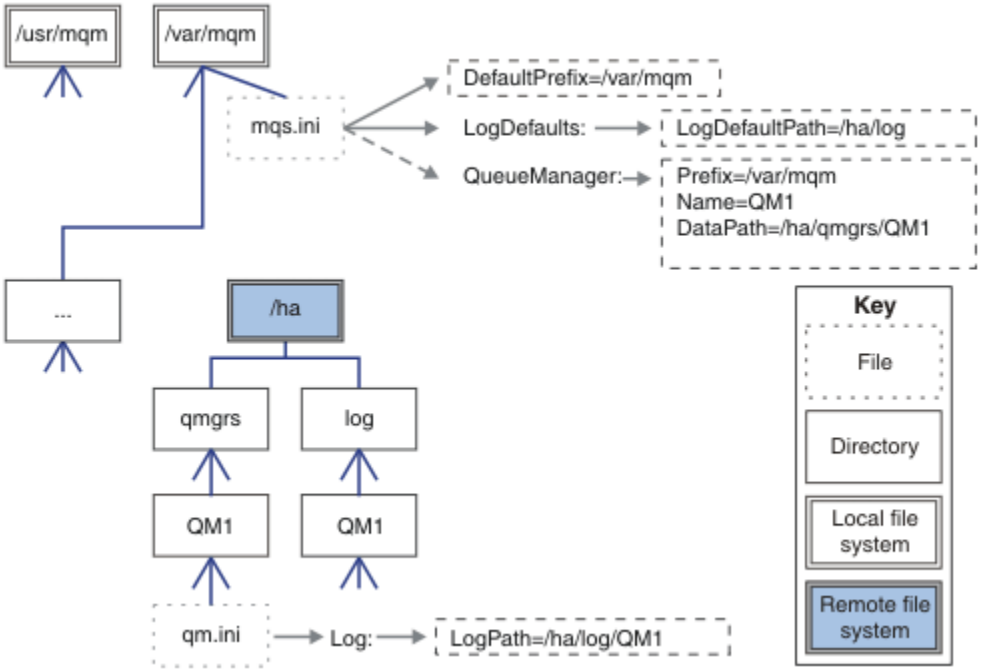


Figura 39. Condividere le directory denominate `qmgrs` e `log`

Condividi tutto

Figura 40 a pagina 137 è una configurazione semplice per il sistema con archiviazione file di rete veloce.

Montare `/var/mqm` come file system condiviso remoto. Per impostazione predefinita, quando si avvia QM1, cerca `/var/mqm`, lo trova sul file system condiviso e legge il file `mqs.ini` in `/var/mqm`. Invece di utilizzare il file `/var/mqm/mqs.ini` singolo per i gestori code su tutti i server, è possibile impostare la variabile di ambiente `AMQ_MQS_INI_LOCATION` su ciascun server in modo che punti a file `mqs.ini` diversi.

Nota: Il contenuto del file di errori generico in `/var/mqm/errors/` viene condiviso tra gestori code su server differenti.

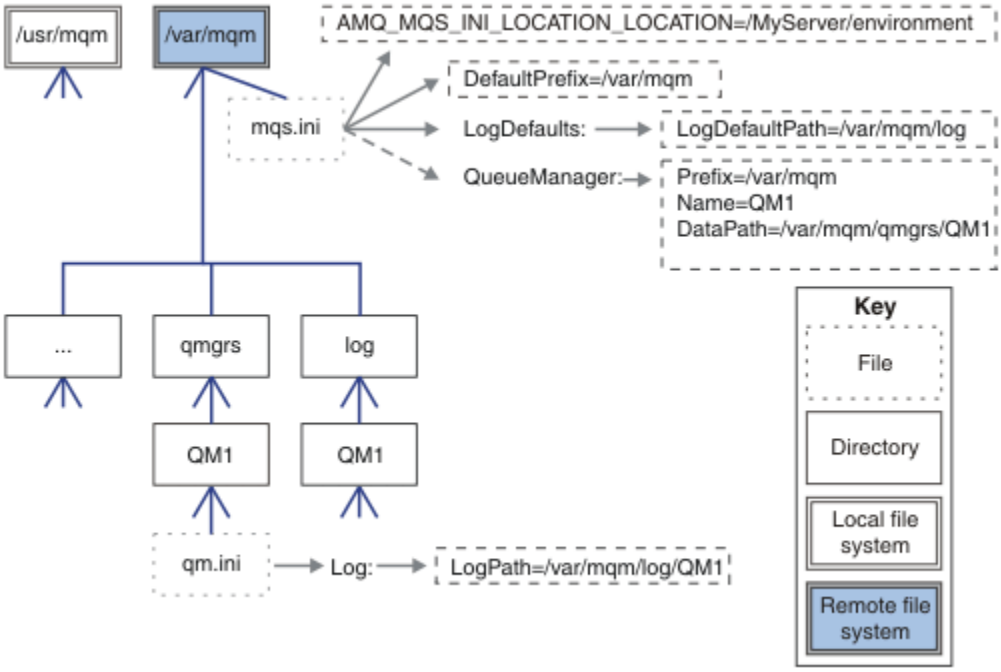


Figura 40. Condividi tutto

Notare che non è possibile utilizzarlo per i gestori code a più istanze. Il motivo è che è necessario che ogni host in un gestore code a più istanze disponga di una propria copia locale di `/var/mqm` per tenere traccia dei dati locali, come i semafori e la memoria condivisa. Queste entità non possono essere condivise tra host.

Windows Struttura di directory su sistemi Windows .

Come trovare le informazioni di configurazione del gestore code e le directory su Windows.

Le directory predefinite per l'installazione di IBM MQ for Windows sono:

Directory programma
```
C:\Program Files\IBM\MQ
```
Directory di dati
```
C:\ProgramData\IBM\MQ
```

Importante: Windows Per le installazioni Windows, le directory sono come indicate, a meno che non vi sia un'installazione precedente del prodotto che contiene ancora delle voci di registro e/o gestori code. In questa situazione, la nuova installazione utilizza il vecchio percorso di directory dei dati. Per ulteriori informazioni, consultare Program and data directory locations.

Se si desidera conoscere la directory di installazione e la directory di dati utilizzata, eseguire il comando dspmqver .

la directory di installazione è elencata nel campo **InstPath** e la directory di dati è elencata nel campo **DataPath** .

L'esecuzione del comando **dspmqver** visualizza, ad esempio, le seguenti informazioni:

```
>dspmqver
Name:        IBM MQ
Version:     9.0.0.0
Level:       p900-L160512.4
BuildType:   IKAP - (Production)
Platform:    IBM MQ for Windows (x64 platform)
Mode:        64-bit
O/S:         Windows 7 Professional x64 Edition, Build 7601: SP1
InstName:    Installation1
InstDesc:
Primary:     Yes
InstPath:    C:\Program Files\IBM\MQ
DataPath:    C:\ProgramData\IBM\MQ
MaxCmdLevel: 900
LicenseType: Production
```

Gestori code a più istanze

Per configurare un gestore code a più istanze, le directory di log e di dati devono essere collocate nella memoria di rete, preferibilmente su un server diverso rispetto a uno qualsiasi dei server che eseguono le istanze del gestore code.

Due parametri vengono forniti nel comando **crtmqm** , **-md** e **-ld**, per rendere più semplice specificare l'ubicazione dei dati del gestore code e le directory di log. L'effetto della specifica del parametro **-md** è quadruplicato:

1. La stanza `mqs.ini` `QueueManager\`*`QmgrName`* contiene una nuova variabile, *DataPath*, che punta alla directory dei dati del gestore code. A differenza della variabile *Prefisso* , il percorso include il nome della directory del gestore code.
2. Le informazioni di configurazione del gestore code memorizzate nel file mqs.ini sono ridotte a *Nome*, *Prefisso*, *Directory* e *DataPath*.

Windows *Contenuto della directory*

Elenca l'ubicazione e il contenuto delle directory IBM MQ .

Una configurazione IBM MQ ha tre serie principali di file e directory:

1. Eseguibile e altri file di sola lettura che vengono aggiornati solo quando viene applicata la manutenzione. Ad esempio:
   - Il file readme
   - Il plug-in di IBM MQ Explorer e i file della guida
   - File di licenza

   Questi file sono descritti in Tabella 16 a pagina 139.
2. File e directory potenzialmente modificabili che non sono specifici di un particolare gestore code. Questi file e directory sono descritti in Tabella 17 a pagina 139.
3. File e directory specifici di ciascun gestore code su un server. Questi file e directory sono descritti in Tabella 18 a pagina 140.

File e directory di risorse

Le directory di risorse e i file contengono tutto il codice eseguibile e le risorse per eseguire un gestore code. La variabile *FilePath*, nella chiave di registro di configurazione IBM MQ specifica dell'installazione, contiene il percorso delle directory delle risorse.

Tabella 16. Directory e file nella directory `FilePath`	
Percorso file	**Indice**
FilePath\bin	Comandi e DLL
FilePath\bin64	Comandi e DLL (64 bit)
FilePath\conv	Tabelle conversione dati
FilePath\doc	File della guida della procedura guidata
FilePath\MQExplorer	Plug-in Eclipse della guida di Explorer ed Explorer
FilePath\gskit8	GSK (Global Security Kit)
FilePath\java	Risorse Java , incluso JRE
FilePath\licenses	Informazioni sulla licenza
FilePath\Non_IBM_License	Informazioni sulla licenza
FilePath\properties	Utilizzato internamente
FilePath\Tivoli	
FilePath\tools	Esempi e risorse di sviluppo
FilePath\web	Descritto nella struttura del file del componente di installazione IBM MQ Console e REST API per i file non modificabili.
FilePath\Uninst	Utilizzato internamente
FilePath\README.TXT	File readme

Directory non specifiche per un gestore code

Alcune directory contengono file, ad esempio file di traccia e log degli errori, che non sono specifici di un determinato gestore code. La variabile *DefaultPrefix* contiene il percorso di queste directory. *DefaultPrefix* fa parte della stanza `AllQueueManagers` .

Tabella 17. Directory e file nella directory `DefaultPrefix`	
Percorso file	**Indice**
DefaultPrefix\config	Utilizzato internamente
DefaultPrefix\conv	File di controllo conversione `ccsid_part2.tbl` e `ccsid.tbl data` , descritto in Conversione dati
DefaultPrefix\errors	Log degli errori non relativi al gestore code, `AMQERR` *nn*`.LOG`
DefaultPrefix\exits	Programmi di uscita canale
DefaultPrefix\exits64	Programmi di uscita canale (64 bit)
DefaultPrefix\ipc	Non utilizzato
DefaultPrefix\qmgrs	Descritto in Tabella 18 a pagina 140
DefaultPrefix\trace	File di traccia
DefaultPrefix\web	Descritto nella struttura del file del componente di installazione IBM MQ Console e REST API per i file modificabili dall'utente
DefaultPrefix\amqmjpse.txt	Utilizzato internamente

Directory del gestore code

Quando si crea un gestore code, viene creata una nuova serie di directory, specifiche del gestore code.

Se si crea un gestore code con il parametro **-md** *filepath* , il percorso viene memorizzato nella variabile *DataPath* nella stanza del gestore code del file mqs.ini . Se si crea un gestore code senza impostare il parametro **-md** *filepath* , le directory del gestore code vengono create nel percorso memorizzato in *DefaultPrefix*, e il percorso viene copiato nella variabile *Prefisso* nella stanza del gestore code del file mqs.ini .

Tabella 18. Directory e file nelle directory DataPath e Prefix\qmgrs\QmgrName

Percorso file	**Indice**
DataPath\@ipcc	Ubicazione predefinita per AMQCLCHL.TAB, la tabella di connessione client.
DataPath\authinfo	Utilizzato internamente.
DataPath\channel	
DataPath\clntconn	
DataPath\errors	Log degli errori, AMQERR *nn*.LOG
DataPath\listener	Utilizzato internamente.
DataPath\namelist	
DataPath\plugcomp	
DataPath\procdef	
DataPath\qmanager	
DataPath\queues	
DataPath\services	
DataPath\ssl	
DataPath\startprm	
DataPath\topic	
DataPath\active	
DataPath\active.dat	
DataPath\amqalchk.fil	
DataPath\master	
DataPath\master.dat	
DataPath\qm.ini	Configurazione del gestore code
DataPath\qmstatus.ini	Stato gestore code
DataPath\userdata	Può essere utilizzato per memorizzare lo stato persistente delle applicazioni.
Prefix\qmgrs*QmgrName*	Utilizzato internamente
Prefix\qmgrs\@SYSTEM	Non utilizzato
Prefix\qmgrs\@SYSTEM\errors	
*DataPath**autocfg*	Utilizzato per la configurazione automatica

IBM i Struttura di directory su IBM i

Viene fornita una descrizione di IFS e la struttura di directory di IBM MQ IFS viene descritta per server, client e Java.

IFS (integrated file system) è una parte di IBM i che supporta la gestione dell'input/output e della memoria del flusso simile al personal computer, ai sistemi operativi AIX and Linux , fornendo una struttura di integrazione su tutte le informazioni memorizzate nel server.

In IBM i i nomi di directory iniziano con il carattere `& (ampersand)` anziché con il carattere `@ (at)`. Ad esempio, `@system` su IBM i è `&system`.

File system root IFS per server IBM MQ

Quando si installa il server IBM MQ per IBM i, le seguenti directory vengono create nel filesystem root IFS.

ProdData:

Panoramica

```
QIBM
   '-- ProdData
      '-- mqm
      '-- doc
      '-- inc
      '-- lib
      '-- samp
      '-- licenses
      '-- LicenseDoc
      '-- 5724H72_V8R0M0
```

/QIBM/ProdData/mqm
Le sottodirectory sottostanti contengono tutti i dati del prodotto, ad esempio le classi C + +, i file di formato traccia e i file di licenza. I dati in questa directory vengono eliminati e sostituiti ogni volta che il prodotto viene installato.

/QIBM/ProdData/mqm/doc
Un riferimento al comando per i comandi CL viene fornito in formato HTML e installato qui.

/QIBM/ProdData/mqm/inc
I file di intestazione per la compilazione dei programmi C o C + +.

/QIBM/ProdData/mqm/lib
File ausiliari utilizzati da MQ.

/QIBM/ProdData/mqm/samp
Ulteriori campioni.

/QIBM/ProdData/mqm/licenses
File di licenza. I due file per ciascuna lingua sono denominati come `LA_` *xx* e `LI_` *xx* , dove *xx* è l'identificativo della lingua di 2 caratteri per ciascuna lingua fornita.

Inoltre, la seguente directory memorizza i file degli accordi di licenza:

/QIBM/ProdData/LicenseDoc/5724H72_V8R0M0
File di licenza. I file sono denominati come `5724H72_V8R0M0_` *xx* dove *xx* è l'identificativo della lingua di 2 o 5 caratteri per ogni lingua fornita.

UserData:

Panoramica

```
QIBM
   '-- UserData
      '-- mqm
```

```
      '-- errors
      '-- trace
      '-- qmgrs
      '-- &system
      '-- qmgrname1
      '-- qmgrname2
      '-- and so on
```

/QIBM/UserData/mqm
Le sottodirectory sottostanti contengono tutti i dati utente relativi ai gestori code.

Quando si installa il prodotto, viene creato un file mqs.ini nella directory /QIBM/UserData/mqm/ (a meno che non sia già presente in un'installazione precedente).

Quando si crea un gestore code, nella directory /QIBM/UserData/mqm/qmgrs/ *QMGRNAME* / viene creato un file qm.ini (dove *QMGRNAME* è il nome del gestore code).

I dati nelle directory vengono conservati quando il prodotto viene eliminato.

File system root IFS per IBM MQ MQI client

Quando si installa IBM MQ MQI client for IBM i, le seguenti directory vengono create nel file system root IFS:

ProdData:

Panoramica

```
QIBM
   '-- ProdData
      '-- mqm
      '-- lib
```

/QIBM/ProdData/mqm
Le sottodirectory al di sotto di questa directory contengono tutti i dati del prodotto. I dati in questa directory vengono eliminati e sostituiti ogni volta che il prodotto viene sostituito.

UserData:

Panoramica

```
QIBM
   '-- UserData
      '-- mqm
      '-- errors
      '-- trace
```

/QIBM/UserData/mqm
Le sottodirectory al di sotto di questa directory contengono tutti i dati utente.

File system root IFS per IBM MQ Java

Quando si installa IBM MQ Java su IBM i, le seguenti directory vengono create nel file system root IFS:

ProdData:

Panoramica

```
QIBM
   '-- ProdData
      '-- mqm
      '-- java
```

```
'--samples
'-- bin
'-- lib
```

/QIBM/ProdData/mqm/java
Le sottodirectory sottostanti contengono tutti i dati del prodotto, incluse le classi Java . I dati in questa directory vengono eliminati e sostituiti ogni volta che il prodotto viene sostituito.

/QIBM/ProdData/mqm/java/samples
Le sottodirectory sottostanti contengono tutti i dati e le classi Java di esempio.

Librerie create dalle installazioni server e client

L'installazione del server o del client IBM MQ crea le librerie riportate di seguito:

- QMQM

  La libreria del prodotto.

- QMQMSAMP

  La libreria degli esempi (se si sceglie di installare gli esempi).

- QMxxxx

  Solo server.

  Ogni volta che si crea un gestore code, IBM MQ crea automaticamente una libreria associata, con un nome come QMxxxx dove xxxx deriva dal nome del gestore code. Questa libreria contiene oggetti specifici del gestore code, inclusi i giornali e i ricevitori associati. Per impostazione predefinita, il nome di questa libreria deriva dal nome del gestore code con prefisso QM. Ad esempio, per un gestore code denominato TEST, la libreria viene denominata QMTEST.

  Nota: Quando si crea un gestore code, è possibile specificare il nome della relativa libreria. Ad esempio:

  ```
  CRTMQM MQMNAME(TEST) MQMLIB(TESTLIB)
  ```

  È possibile utilizzare il comando WRKLIB per elencare tutte le librerie create da IBM MQ per IBM i . Rispetto alle librerie del gestore code, verrà visualizzato QMGR: QMGRNAME. Il formato del comando è:

  ```
  WRKLIB LIB(QM*)
  ```

  Queste librerie associate al gestore code vengono conservate quando il prodotto viene eliminato.

Multi Pianificazione del supporto file system per MFT su Multiplatforms

Gli agent IBM MQ Managed File Transfer MFT possono essere utilizzati per trasferire i dati da e verso i file su un filesystem. Inoltre, i monitoraggi delle risorse in esecuzione all'interno di un agent possono essere configurati per monitorare i file su un file system.

MFT richiede che questi file siano memorizzati su un filesystem che supporta il blocco. Ci sono due ragioni per questo:

- Un agent blocca un file per assicurarsi che non venga modificato una volta che ha iniziato la lettura dei dati da esso o la scrittura dei dati su di esso.
- I monitoraggi delle risorse bloccano i file per controllare che nessun altro processo li stia attualmente utilizzando.

Gli agent e i monitoraggi delle risorse utilizzano il Java metodo **`FileChannel.tryLock()`** per eseguire il blocco e il file system deve essere in grado di bloccare i file quando richiesto utilizzando questa chiamata.

Importante: I seguenti file system non sono supportati poiché non soddisfano i requisiti tecnici di MFT:

- GlusterFS

- NFS versione 3

Multi Scelta della registrazione circolare o lineare su Multiplatforms

In IBM MQ, è possibile scegliere la registrazione circolare o lineare. Le seguenti informazioni forniscono una panoramica di entrambi i tipi.

Vantaggi della registrazione circolare

I principali vantaggi della registrazione circolare sono:

- Più facile da amministrare.

  Una volta configurata correttamente la registrazione circolare per il carico di lavoro, non è necessaria alcuna ulteriore gestione. Mentre, per la registrazione lineare, le immagini dei supporti devono essere registrate e le estensioni di log che non sono più richieste devono essere archiviate o eliminate.
- Prestazioni migliori

  La registrazione circolare funziona meglio della registrazione lineare, perché la registrazione circolare è in grado di riutilizzare le estensioni di log che sono già state formattate. Mentre la registrazione lineare deve allocare nuove estensioni log e formattarle.

Per ulteriori informazioni, vedi Gestione dei log .

Vantaggi della registrazione lineare

Il vantaggio principale della registrazione lineare è che la registrazione lineare fornisce protezione contro ulteriori errori.

Né la registrazione circolare né la registrazione lineare proteggono da un log danneggiato o eliminato, o da messaggi o code che sono stati eliminati dalle applicazioni o dall'amministratore.

La registrazione lineare (ma non circolare) consente il recupero degli oggetti danneggiati. Quindi, la registrazione lineare fornisce protezione contro i file delle code danneggiati o eliminati, poiché queste code danneggiate possono essere recuperate da un log lineare.

Protezione circolare e lineare contro la perdita di alimentazione e gli errori di comunicazione come descritto in Ripristino da perdita di alimentazione o errori di comunicazione.

Altre considerazioni

La scelta tra lineare o circolare dipende dalla ridondanza richiesta.

Vi è un costo per la scelta di una maggiore ridondanza, ossia la registrazione lineare, causato dal costo delle prestazioni e dal costo di gestione.

Per ulteriori informazioni, vedi Tipi di registrazione .

AIX Memoria condivisa su AIX

Se alcuni tipi di applicazione non riescono a connettersi a causa di una limitazione della memoria AIX , nella maggior parte dei casi questo problema può essere risolto impostando la variabile di ambiente EXTSHM=ON.

Alcuni processi a 32 bit su AIX potrebbero incontrare una limitazione del sistema operativo che influisce sulla loro capacità di connettersi ai gestori code IBM MQ . Ogni connessione standard a IBM MQ utilizza la memoria condivisa, ma a differenza di altre piattaforme UNIX , AIX consente ai processi a 32 bit di collegare solo 11 serie di memoria condivisa.

La maggior parte dei processi a 32 bit non incontrerà questo limite, ma le applicazioni con requisiti di memoria elevati potrebbero non riuscire a collegarsi a IBM MQ con codice di errore 2102: MQRC_RESOURCE_PROBLEM. I seguenti tipi di applicazione potrebbero visualizzare questo errore:

- Programmi in esecuzione in macchine virtuali Java a 32 bit
- Programmi che utilizzano modelli di memoria grandi o molto grandi
- Programmi che si collegano a molti gestori code o database
- Programmi che si collegano a serie di memoria condivisa da soli

AIX offre una funzione di memoria condivisa estesa per i processi a 32 bit che consente loro di collegare più memoria condivisa. Per eseguire un'applicazione con questa funzione, esportare la variabile di ambiente EXTSHM=ON prima di avviare i gestori code e il programma. La funzione EXTSHM=ON previene questo errore nella maggior parte dei casi, ma è incompatibile con i programmi che utilizzano l'opzione SHM_SIZE della funzione shmctl.

Le applicazioni IBM MQ MQI client e tutti i processi a 64 - bit non sono interessati da questa limitazione. Possono connettersi ai gestori code IBM MQ indipendentemente dal fatto che EXTSHM sia stato impostato o meno.

Linux AIX Risorse IPC IBM MQ e UNIX System V

Un gestore code utilizza alcune risorse IPC. Utilizzare **ipcs -a** per individuare le risorse utilizzate.

Queste informazioni si applicano solo a sistemi IBM MQ in esecuzione su AIX and Linux

IBM MQ utilizza risorse IPC (interprocess communication) System V (*semafori* e *segmenti di memoria condivisi*) per memorizzare e trasmettere i dati tra i componenti del sistema. Queste risorse sono utilizzate dai processi del gestore code e dalle applicazioni che si connettono al gestore code. IBM MQ MQI clients non utilizzano le risorse IPC, ad eccezione del controllo di traccia IBM MQ . Utilizzare il UNIX comando **ipcs -a** per ottenere informazioni complete sul numero e la dimensione delle risorse IPC attualmente in uso sulla macchina.

Linux AIX Priorità processo IBM MQ e UNIX

Buone pratiche durante l'impostazione dei valori *nice* della priorità del processo.

Queste informazioni si applicano solo a sistemi IBM MQ in esecuzione su AIX and Linux

Se si esegue un processo in background, a tale processo può essere assegnato un valore *nice* superiore (e quindi una priorità inferiore) dalla shell di richiamo. Ciò potrebbe avere implicazioni generali sulle prestazioni di IBM MQ . In situazioni di stress elevato, se ci sono molti thread pronti per l'esecuzione con una priorità più alta e alcuni con una priorità più bassa, le caratteristiche di pianificazione del sistema operativo possono privare i thread con priorità più bassa del tempo del processore.

È buona norma che i processi avviati in modo indipendente associati ai gestori code, come ad esempio **runmqlsr**, abbiano gli stessi valori *nice* del gestore code a cui sono associati. Assicurarsi che la shell non assegni un valore *nice* superiore a questi processi in background. Ad esempio, in ksh, utilizzare l'impostazione "`set +o bgnice`" per impedire a ksh di aumentare il valore *nice* dei processi in background. È possibile verificare i valori *nice* dei processi in esecuzione esaminando la colonna *NI* di un elenco "ps -efl" .

Inoltre, avviare i processi dell'applicazione IBM MQ con lo stesso valore *nice* del gestore code. Se vengono eseguiti con valori *nice* differenti, un thread dell'applicazione potrebbe bloccare un thread del gestore code o viceversa, causando un peggioramento delle prestazioni.

z/OS Planning your IBM MQ environment on z/OS

When planning your IBM MQ environment, you must consider the resource requirements for data sets, page sets, Db2, Coupling Facilities, and the need for logging, and backup facilities. Use this topic to plan the environment where IBM MQ runs.

Before you plan your IBM MQ architecture, familiarize yourself with the basic IBM MQ for z/OS concepts, see the topics in IBM MQ for z/OS concepts.

When planning your queue manager, you might need to work with different people in your organization. It is usually a good idea to involve those people early, as change control procedures can take a long time. They might also be able to tell you what parameters you need to configure IBM MQ for z/OS.

For example you might need to work with the:

- Storage administrator, to determine the high level qualifier of queue manager data sets, and to allocate enough space for queue manager data sets.
- z/OS system programmer to define the IBM MQ subsystem to z/OS and APF authorize the IBM MQ for z/OS libraries.
- Network administrator to determine which TCP/IP stack and ports should be used for IBM MQ for z/OS.
- Security administrator to set up access to queue manager data sets, security profiles for IBM MQ for z/OS resources, and TLS certificates.
- Db2 administrator to set up Db2 tables when configuring a queue sharing group.

Related concepts

IBM MQ Technical overview

Related tasks

“Pianificazione di un'architettura IBM MQ” on page 5

Quando si pianifica l'ambiente IBM MQ , considerare il supporto fornito da IBM MQ per le architetture di gestori code singoli e multipli e per gli stili di messaggistica point-to-point e di pubblicazione / sottoscrizione. Inoltre, pianificare i requisiti delle risorse e l'utilizzo delle funzioni di registrazione e backup.

Configuring z/OS

Administering IBM MQ for z/OS

z/OS Planning for your queue manager

When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

The best way to configure a queue manager is in steps:

1. Configure the base queue manager
2. Configure the channel initiator which does queue manager to queue manager communications, and remote client application communication
3. If you want to encrypt and protect messages, configure Advanced Message Security
4. If you want to use File Transfer over IBM MQ, configure Managed File Transfer for z/OS.
5. If you want to use the administrative or messaging REST API, or the IBM MQ Console to manage IBM MQ from a web browser, configure the mqweb server.

Some enterprises have hundreds of thousands of queue managers in their environment. You need to consider your IBM MQ network now, and in five years time.

On z/OS, some queue managers process thousands of messages a second, and log over 100 MB a second. If you expect very high volumes you may need to consider having more than one queue manager.

On z/OS, IBM MQ can run as part of a queue sharing group (QSG) where messages are stored in the Coupling Facility, and any queue manager in the queue sharing group can access the messages. If you want to run in a queue sharing group you need to consider how many queue managers you need. Typically, there is one queue manager for each LPAR. You might also have one queue manager to backup CF structures regularly.

Some changes to configuration are easy to do, such as defining a new queue. Some are harder, such as making logs and page sets bigger; and some configuration cannot be changed, such as the name of a queue manager or the queue sharing group name.

There is performance and tuning information available in the MP16 performance SupportPac .

Naming conventions

You need to have a naming convention for the queue manager data sets.

Many enterprises use the release number in the name of the load libraries, and so on. You might want to consider having an alias of `MQM.SCSQAUTH` pointing to the version currently in use, such as `MQM.V930.SCSQAUTH`, so you do not have to change CICS®, Batch, and IMS JCL when you migrate to a new version of IBM MQ.

You can use a symbolic link in z/OS UNIX System Services to reference the installation directory for the version of IBM MQ currently in use.

The data sets used by the queue manager (logs, page sets, JCL libraries) need a naming convention to simplify the creation of security profiles, and the mapping of data sets to SMS storage classes that control where the data sets are placed on disk, and the attributes they have.

Note, that putting the version of IBM MQ into the name of the page sets or logs, is not a good idea. One day you might migrate to a new version, and the data set will have the "wrong" names.

Applications

You need to understand the business applications and the best way to configure IBM MQ. For example if applications have logic to provide recovery and repeat capability, then non persistent messages might be good enough. If you want IBM MQ to handle the recovery, then you need to use persistent messages and put and get messages in syncpoint.

You need to isolate queues from different business transactions. If a queue for one business application fills up, you do not want this impacting other business applications. Isolate the queues in different page sets and buffer pools, or structures, if possible.

You need to understand the profile of messages. For many applications the queues have only a few messages. Other applications can have queues build up during the day, and be processed overnight. A queue which normally has only a few messages on it, might need to hold many hours worth of messages if there is a problem and messages are not processed. You need to size the CF structures and page sets to allow for your expected peak capacity.

Post configuration

Once you have configured your queue manager (and components) you need to plan for:

- Backing up page sets.
- Backing up definitions of objects.
- Automating the backup of any CF structures.
- Monitoring IBM MQ messages, and taking action when a problem is detected.
- Collecting the IBM MQ statistics data.
- Monitoring resource usage, such as virtual storage, and amount of data logged per hour. With this you can see if your resource usage is increasing and if you need to take actions, such as setting up a new queue manager

z/OS Planning your storage and performance requirements on z/OS

You must set realistic and achievable storage, and performance goals for your IBM MQ system. Use this topic help you understand the factors which affect storage, and performance.

This topic contains information about the storage and performance requirements for IBM MQ for z/OS. It contains the following sections:

- z/OS performance options for IBM MQ
- Determining z/OS workload management importance and velocity goals
- “Library storage” on page 148

- “System LX usage” on page 148
- “Storage configuration” on page 149
- “Disk storage” on page 154

See, “Where to find more information about storage and performance requirements” on page 154 for more information.

z/OS performance options for IBM MQ

With workload management, you define performance goals and assign a business importance to each goal. You define the goals for work in business terms, and the system decides how much resource, such as processor and storage, should be given to the work to meet its goal. Workload management controls the dispatching priority based on the goals you supply. Workload management raises or lowers the priority as needed to meet the specified goal. Thus, you need not fine-tune the exact priorities of every piece of work in the system and can focus instead on business objectives.

The three kinds of goals are:

Response time
How quickly you want the work to be processed

Execution velocity
How fast the work should be run when ready, without being delayed for processor, storage, I/O access, and queue delay

Discretionary
A category for low priority work for which there are no performance goals

Response time goals are appropriate for end-user applications. For example, CICS users might set workload goals as response time goals. For IBM MQ address spaces, velocity goals are more appropriate. A small amount of the work done in the queue manager is counted toward this velocity goal but this work is critical for performance. Most of the work done by the queue manager counts toward the performance goal of the end-user application. Most of the work done by the channel initiator address space counts toward its own velocity goal. The receiving and sending of IBM MQ messages, which the channel initiator accomplishes, is typically important for the performance of business applications using them.

Determining z/OS workload management importance and velocity goals

See “Determining z/OS workload management importance” on page 149 for more information.

Library storage

You must allocate disk storage for the product libraries. The exact figures depend on your configuration, and should include both the target and distribution libraries, as well as the SMP/E libraries.

The target libraries used by IBM MQ for z/OS use PDSE formats. Ensure that any PDSE target libraries are not shared outside a sysplex. For more information about the required libraries and their sizes and the required format, see the Program Directory. Per i collegamenti di download per le directory del programma, consultare IBM MQ for z/OS Program Directory PDF files.

System LX usage

Each defined IBM MQ subsystem reserves one system linkage index (LX) at IPL time, and a number of non-system linkage indexes when the queue manager is started. The system linkage index is reused when the queue manager is stopped and restarted. Similarly, distributed queuing reserves one non-system linkage index. In the unlikely event of your z/OS system having inadequate system LXs defined, you might need to take these reserved system LXs into account.

If required, the number of system LXs can be increased by setting the *NSYSLX* parameter in SYS1.PARMLIB member IEASYSxx.

z/OS *Determining z/OS workload management importance*

For full information about workload management and defining goals through the service definition, see the .z/OS product documentation.

This topic suggests how to set the z/OS workload management importance and velocity goals relative to other important work in your system. See *z/OS MVS Planning: Workload Management* for more information.

The queue manager address space needs to be defined with high priority as it provides subsystem services. The channel initiator is an application address space, but is usually given a high priority to ensure that messages being sent to a remote queue manager are not delayed. Advanced Message Security (AMS) also provides subsystem services and needs to be defined with high priority.

Use the following service classes:

The default SYSSTC service class

- VTAM and TCP/IP address spaces
- IRLM address space (IRLMPROC)

Note: The VTAM, TCP/IP, and IRLM address spaces must have a higher dispatching priority than all the DBMS address spaces, their attached address spaces, and their subordinate address spaces. Do not allow workload management to reduce the priority of VTAM, TCP/IP, or IRLM to (or below) that of the other DBMS address spaces

A high velocity goal and importance of 1 for a service class with a name that you define, such as PRODREGN, for the following:

- IBM MQ queue manager, channel initiator and AMS address spaces
- Db2 (all address spaces, except for the Db2-established stored procedures address space)
- CICS (all region types)
- IMS (all region types except BMPs)

A high velocity goal is good for ensuring that startups and restarts are performed as quickly as possible for all these address spaces.

The velocity goals for CICS and IMS regions are only important during startup or restart. After transactions begin running, workload management ignores the CICS or IMS velocity goals and assigns priorities based on the response time goals of the transactions that are running in the regions. These transaction goals should reflect the relative priority of the business applications they implement. They might typically have an importance value of 2. Any batch applications using IBM MQ should similarly have velocity goals and importance reflecting the relative priority of the business applications they implement. Typically the importance and velocity goals will be less than those for PRODREGN.

z/OS *Storage configuration*

V 9.4.0 In a 64 bit address space, there is a virtual line called "the bar" that marks the 2GB address. The bar separates storage below the 2GB address, called "below the bar", from storage above the 2GB address, called "above the bar". Storage below the bar uses 31 bit addressability, storage above the bar uses 64 bit addressability.

V 9.4.0

You can specify the limit of 31-bit storage by using the JCL REGION parameter, and the limit of 64-bit storage by using the MEMLIMIT parameter. These specified values can be overridden by z/OS exits.

Suggested storage configuration

The following table shows suggested **REGION** and **MEMLIMIT** values for the queue manager, channel initiator, and AMS address spaces. These suggestions should be used as a starting point and adjusted using the information in:

- “Queue manager storage configuration” on page 150
- “Channel initiator storage configuration from IBM MQ 9.4.0” on page 152

Table 19. Suggested definitions for REGION and MEMLIMIT

Address space	**Storage configuration**
Queue manager	REGION=0M, MEMLIMIT=3G
V 9.4.0 Channel initiator from IBM MQ 9.4.0	REGION=0M, MEMLIMIT=2G
AMS address space	REGION=0M

Managing the MEMLIMIT and REGION size

Other mechanisms, for example the **MEMLIMIT** parameter in the SMFPRMxx member of SYS1.PARMLIB or the IEFUSI exit might be used at your installation to provide a default amount of virtual storage above the bar for z/OS address spaces. See Memory management above the bar for full details about limiting storage above the bar.

z/OS *Queue manager storage configuration*

The queue manager address space is likely to be the major user of 64-bit storage in an IBM MQ installation. Each connection to the queue manager requires common storage to be allocated as described in the following text. In addition to 64-bit storage, you should allow the queue manager to use all available 31-bit storage by specifying REGION=0M on the queue manager JCL.

Common storage

Each IBM MQ for z/OS subsystem has the following approximate storage requirements:

- CSA 4KB
- ECSA 800KB, plus the size of the trace table that is specified in the **TRACTBL** parameter of the CSQ6SYSP system parameter macro. For more information, see Using CSQ6SYSP.

In addition, each concurrent logical connection to the queue manager requires about 5 KB of ECSA. When a task ends, other IBM MQ tasks can reuse this storage.

IBM MQ does not release the storage until the queue manager is shut down, so you can calculate the maximum amount of ECSA required by multiplying the maximum number of concurrent connections by 5KB. The number of concurrent logical connections is the sum of the number of:

- Tasks (TCBs) in Batch, TSO, z/OS UNIX System Services, IMS, and Db2 stored procedure address space (SPAS) regions that are connected to IBM MQ, but not disconnected.
- CICS transactions that have issued an IBM MQ request, but have not terminated
- JMS Connections, Sessions, TopicSessions or QueueSessions that have been created (for bindings connection), but not yet destroyed or garbage collected.
- Active IBM MQ channels

You can set a limit to the common storage, used by logical connections to the queue manager, with the **ACELIM** configuration parameter. The **ACELIM** control is primarily of interest to sites where Db2 stored procedures cause operations on IBM MQ queues.

When driven from a stored procedure, each IBM MQ operation can result in a new logical connection to the queue manager. Large Db2 units of work, for example due to table load, can result in an excessive demand for common storage.

ACELIM is intended to limit common storage use and to protect the z/OS system, by limiting the number of connections in the system. You should only set **ACELIM** on queue managers that have been identified

as using excessive quantities of ECSA storage. See the **ACELIM** section in *Using CSQ6SYSP* for more information.

To set a value for **ACELIM**, firstly determine the amount of storage currently in the subpool controlled by the **ACELIM** value. This information is in the SMF 115 subtype 5 records produced by statistics CLASS(3) trace.

IBM MQ SMF data can be formatted using SupportPac MP1B. The number of bytes in use in the subpool controlled by **ACELIM** is displayed in the STGPOOL DD, on the line titled *ACE/PEB*.

For more information about SMF 115 statistics records, see Interpreting IBM MQ for z/OS performance statistics.

Increase the normal value by a sufficient margin to provide space for growth and workload spikes. Divide the new value by 1024 to yield a maximum storage size in KB for use in the **ACELIM** configuration.

Private storage

The queue manager address space uses 64-bit storage for many internal control blocks. The **MEMLIMIT** parameter of the queue manager JCL defines the maximum amount of 64-bit storage available. 3GB of storage, `MEMLIMIT=3G`, is the minimum you should use, however, depending on your configuration significantly more might be required.

You should specify a specific **MEMLIMIT** value rather than `MEMLIMIT=NOLIMIT` to prevent potential problems. If you specify `NOLIMIT` or a very large value, then there is the potential to use up all of the available z/OS virtual storage, which leads to paging in your system. When increasing the value of **MEMLIMIT** you should discuss the new setting with your z/OS system programmer in case there is a system-wide limit on the amount of on storage that can be used.

If you have a large value for **MEMLIMIT** you might need to increase the size of your dump data sets as more data is captured in a dump.

You can monitor the address space storage usage from the CSQY220I message that indicates the amount of 31 and 64-bit private storage in use, and the remaining free amount.

Buffer pools

Buffer pools are a significant user of private storage in the queue manager address space. Each buffer pool size is determined at queue manager initialization time, and storage is allocated for the buffer pool when a page set that is using that buffer pool is connected. The parameter **LOCATION (ABOVE|BELOW)** is used to specify where the buffers are allocated. You can use the ALTER BUFFPOOL command to dynamically change the size of buffer pools.

When calculating a value for **MEMLIMIT** it is critical that you take into account the buffer pool sizes if they are configured with **LOCATION(ABOVE)**. You should perform the calculation as follows.

Calculate the value of **MEMLIMIT** as 2GB plus the size of the buffer pools configured with **LOCATION(ABOVE)**, rounded up to the nearest GB. Set MEMLIMIT to a minimum of 3GB and increase this as necessary when you need to increase the size of your buffer pools.

For example, for three buffer pools configured with **LOCATION(ABOVE)**, buffer pool one has 10,000 buffers, and buffer pools two and three have 50,000 buffers each. Memory usage above the bar equals 110,000 (total number of buffers) * 4096 = 450,560,000 bytes = 430MB.

All buffer pools regardless of **LOCATION** make use of 64-bit storage for control structures. As the number of buffer pools and number of buffers in those pools increase this can become significant. Each buffer requires around an additional 200 bytes of 64-bit storage. For the preceding configuration that would require: 200 * 110,000 = 22,000,000 bytes = 21MB.

Therefore, in this scenario 3GB can be used for the **MEMLIMIT**, which allows scope for growth: 21MB + 430MB + 2GB which rounds up to 3GB.

For some configurations there can be significant performance benefits to using buffer pools that have their buffers permanently backed by real storage. You can achieve this by specifying the FIXED4KB value

for the **PAGECLAS** attribute of the buffer pool. However, you should only do this if there is sufficient real storage available on the LPAR, otherwise other address spaces might be affected. For information about when you should use the FIXED4KB value for **PAGECLAS**, see IBM MQ Support Pac MP16: IBM MQ for z/OS - Capacity planning & tuning.

Making the buffer pools so large that there is MVS™ paging might adversely affect performance. You might consider using a smaller buffer pool that does not page, with IBM MQ moving the message to and from the page set.

Indexed queues

On z/OS, local queues are indexed if the queue has an **INDXTYPE** attribute that has not been set to NONE. The indexes for shared queues are held in a coupling facility, but for private queues the index is held in 64 bit storage. For each message on an indexed queue 136 bytes of data are used to index the message. For very deep queues this can result in a significant amount of 64 bit storage being allocated. For example, 10 million messages on an indexed queue will use 1.27 GB of 64 bit storage in order to maintain the index.

If you expect to have a large number of messages on indexed queues you should allow for this when setting **MEMLIMIT**. To calculate an upper limit for the amount of storage required for indexes, multiply the **MAXDEPTH** attribute for each indexed queue by 136 and sum the value. This value should be added to your existing **MEMLIMIT**.

V 9.4.0 RECOVER CFSTRUCT

From IBM MQ 9.4.0 the **RECOVER CFSTRUCT** command makes greater use of 64-bit storage. In many cases there should be spare 64-bit storage available and so use of the command does not require an increase in the value of **MEMLIMIT**. However, if you are likely to have large structure backups, containing more than a few million messages, you should increase the **MEMLIMIT** for all queue managers which might process the **RECOVER CFSTRUCT** command by 500MB.

For example if you had MEMLIMIT=3G already, you should consider using MEMLIMIT=4G as the **MEMLIMIT** parameter does not allow for decimal points.

Shared Message Data Set (SMDS) buffers and MEMLIMIT

When running messaging workloads using shared message data sets, there are two levels of optimizations that can be achieved by adjusting the DSBUFS and DSBLOCK attributes.

The amount of above bar queue manager storage used by the SMDS buffer is DSBUFS x DSBLOCK. This means that by default, 100 x 256KB (25MB) is used for each CFLEVEL(5) structure in the queue manager.

Although this value is not too high, if your enterprise, or enterprises have many CFSTRUCTs, some of them might allocate a high value of MEMLIMIT for buffer pools, and sometimes they have deep indexed queues, so in total, they might run out of storage above the bar.

V 9.4.0 z/OS *Channel initiator storage configuration from IBM MQ 9.4.0*

The channel initiator typically uses much less 64-bit storage than the queue manager. However, from IBM MQ 9.4.0 the usage has increased. In addition to 64-bit storage, you should allow the channel initiator to use all available 31-bit storage by specifying REGION=0M on the queue manager JCL.

Common storage

The channel initiator typically requires ECSA usage of up to 160KB.

31-bit private storage

The 31-bit storage available to the channel initiator limits the number of concurrent connections the CHINIT can have.

Every channel uses approximately 170KB of extended private region in the channel initiator address space. For message channels, for example, sender or receiver channels, storage is increased by message size if messages larger than 32KB are transmitted. This increased storage is freed when:

- A sending or client channel requires less than half the current buffer size for 10 consecutive messages.
- A heartbeat is sent or received.

The storage is freed for reuse within the Language Environment, however, the storage is not seen as free by the z/OS virtual storage manager. This means that the upper limit for the number of channels is dependent on message size and arrival patterns, and on limitations of individual user systems on extended private region size.

The upper limit on the number of channels is likely to be approximately 9000 on many systems because the extended region size is unlikely to exceed 1.6GB.

The channel initiator trace is written to a data space. The size of the data space storage, is controlled by the **TRAXTBL** parameter. See ALTER QMGR.

64-bit private storage

The MEMLIMIT parameter of the channel initiator JCL defines the maximum amount of 64-bit storage available. 2 GB of storage, MEMLIMIT=2 GB, is the minimum value you should use. Depending on your configuration significantly more might be required.

You should specify a sensible MEMLIMIT value rather than MEMLIMIT=NOLIMIT to prevent potential problems. If you specify NOLIMIT or a very large value, then there is the potential to use up all of the available z/OS virtual storage, leading to paging in your system. When increasing the value of MEMLIMIT you should discuss the new setting with your z/OS system programmer in case there is a system-wide limit on the amount of on storage that can be used.

If you have a large value for MEMLIMIT you might need to increase the size of your dump data sets as more data is captured in a dump.

There are two users of 64-bit storage in the channel initiator: SMF and server-connection channels.

SMF

If enabled, SMF class 4 accounting, or statistics, require 64-bit storage. A minimum of 256MB storage is required. If sufficient storage is not available, the channel initiator issues the CSQX124E message and class 4 accounting and statistics are not available.

Server-connection channels

From IBM MQ 9.4.0 server-connection channels allocate message buffers in 64-bit storage, if they are transferring messages larger than 32 KB in size.

These buffers are freed if the channels require less than half the current buffer size for 10 consecutive messages, or a heartbeat is sent or received.

The value of MEMLIMIT sets an upper limit on how many concurrent server-connection channels can run. You should use a minimum value of MEMLIMIT=2G to ensure that the same number of channels can run as in earlier versions of IBM MQ, as well as providing some capacity for growth.

You can calculate an approximate value for MEMLIMIT by working out the peak maximum number of concurrently active server-connection channels, and for those channels the maximum message size you expect them to transfer. You should use MEMLIMIT=2GB as a starting point and round up.

For example, if you set the maximum number of concurrent server-connection channels to be 2,000 and each channel to have a maximum message size of 1MB, then server-connection channels are using a maximum of just under 2GB of 64-bit storage. As this is very close to 2GB then you should round up to MEMLIMIT=3G.

z/OS *Disk storage*

Use this topic when planning your disk storage requirements for log data sets, Db2 storage, coupling facility storage, and page data sets.

Work with your storage administrator to determine where to put the queue manager data sets. For example, your storage administrator may give you specific DASD volumes, or SMS storage classes, data classes, and management classes for the different data set types.

- Log data sets must be on DASD. These logs can have high I/O activity with a small response time and do not need to be backed up.
- Archive logs can be on DASD or tape. After they have been created, they might never be read again except in an abnormal situation, such as recovering a page set from a backup. They should have a long retention date.
- Page sets might have low to medium activity and should be backed up regularly. On a high use system, they should be backed up twice a day.
- BSDS data sets should be backed up daily; they do not have high I/O activity.

All data sets are similar to those used by Db2, and similar maintenance procedures can be used for IBM MQ.

See the following sections for details of how to plan your data storage:

- **Logs and archive storage**

  "How long do I need to keep archive logs" on page 173 describes how to determine how much storage your active log and archive data sets require, depending on the volume of messages that your IBM MQ system handles and how often the active logs are offloaded to your archive data sets.
- **Db2 storage**

  "Db2 storage" on page 190 describes how to determine how much storage Db2 requires for the IBM MQ data.
- **coupling facility storage**

  "Defining coupling facility resources" on page 180 describes how to determine how large to make your coupling facility structures.
- **Page set and message storage**

  "Planning your page sets and buffer pools" on page 155 describes how to determine how much storage your page data sets require, depending on the sizes of the messages that your applications exchange, on the numbers of these messages, and on the rate at which they are created or exchanged.

z/OS *Where to find more information about storage and performance requirements*

Use this topic as a reference to find more information about storage and performance requirements.

You can find more information from the following sources:

Table 20. Where to find more information about storage requirements

Topic	Where to look
System parameters	Using CSQ6SYSP and Customizing your queue managers
Storage required to install IBM MQ	Program Directory. Per i collegamenti di download per le directory del programma, consultare IBM MQ for z/OS Program Directory PDF files.
IEALIMIT and IEFUSI exits	See IEALIMIT and IEFUSI in the *z/OS:MVS Installation Exits* documentation.

Table 20. Where to find more information about storage requirements (continued)	
Topic	**Where to look**
Latest information	IBM MQ SupportPac website SupportPacs per IBM MQ e altre aree progetto.
Workload management and defining goals through the service definition	*z/OS MVS Planning: Workload Management*

z/OS Planning your page sets and buffer pools

Information to help you with planning the initial number, and sizes of your page data sets, and buffer pools.

This topic contains the following sections:

Plan your page sets

Page set usage

For short-lived messages, few pages are normally used on the page set and there is little or no I/O to the data sets except at startup, during a checkpoint, or at shutdown.

For long-lived messages, those pages containing messages are normally written out to disk. This operation is performed by the queue manager in order to reduce restart time.

Separate short-lived messages from long-lived messages by placing them on different page sets and in different buffer pools.

Number of page sets

Using several large page sets can make the role of the IBM MQ administrator easier because it means that you need fewer page sets, making the mapping of queues to page sets simpler.

Using multiple, smaller page sets has a number of advantages. For example, they take less time to back up, and I/O can be carried out in parallel during backup and restart. However, consider that this adds a significant performance cost to the role of the IBM MQ administrator, who is required to map each queue to one of a much greater number of page sets.

Define at least five page sets, as follows:

- A page set reserved for object definitions (page set zero)
- A page set for system-related messages
- A page set for performance-critical long-lived messages

- A page set for performance-critical short-lived messages
- A page set for all other messages

"Defining your buffer pools" on page 160 explains the performance advantages of distributing your messages on page sets in this way.

Size of page sets

Define sufficient space in your page sets for the expected peak message capacity. Consider for any unexpected peak capacity, such as when a build-up of messages develops because a queue server program is not running. You can do this by allocating the page set with secondary extents or, alternatively, by enabling dynamic page set expansion. For more information, see "Enabling dynamic page set expansion" on page 158. It is difficult to make a page set smaller, so it is often better to allocate a smaller page set, and allow it to expand when needed.

When planning page set sizes, consider all messages that might be generated, including non-application message data. For example, trigger messages, event messages and any report messages that your application has requested.

The size of the page set determines the time taken to recover a page set when restoring from a backup, because a large page set takes longer to restore.

Note: Recovery of a page set also depends on the time the queue manager takes to process the log records written since the backup was taken; this time period is determined by the backup frequency. For more information, see "Planning for backup and recovery" on page 192.

Note: Page sets larger than 4 GB require the use of SMS extended addressability.

Planning for z/OS data set encryption

You can apply the z/OS data set encryption feature to page sets for queue managers running at IBM MQ for z/OS 9.1.4 or later.

You must allocate these page sets with EXTENDED attributes, and a data set key label that ensures the data is AES encrypted.

See the section, confidentiality for data at rest on IBM MQ for z/OS with data set encryption. for more information.

Calculate the size of your page sets

For queue manager object definitions (for example, queues and processes), it is simple to calculate the storage requirement because these objects are of fixed size and are permanent. For messages however, the calculation is more complex for the following reasons:

- Messages vary in size.
- Messages are transitory.
- Space occupied by messages that have been retrieved is reclaimed periodically by an asynchronous process.

Large page sets of greater than 4 GB that provide extra capacity for messages if the network stops, can be created if required. It is not possible to modify the existing page sets. Instead, new page sets with extended addressability and extended format attributes, must be created. The new page sets must be the same physical size as the old ones, and the old page sets must then be copied to the new ones. If backward migration is required, page set zero must not be changed. If page sets less than 4 GB are adequate, no action is needed.

Page set zero

Page set zero is reserved for object definitions.

For page set zero, the storage required is:

```
   (maximum number of local queue definitions x 1010)
      (excluding shared queues)
+  (maximum number of model queue definitions x 746)
+  (maximum number of alias queue definitions x 338)
+  (maximum number of remote queue definitions x 434)
+  (maximum number of permanent dynamic queue definitions x 1010)
+  (maximum number of process definitions x 674)
+  (maximum number of namelist definitions x 12320)
+  (maximum number of message channel definitions x 2026)
+  (maximum number of client-connection channel definitions x 5170)
+  (maximum number of server-connection channel definitions x 2026)
+  (maximum number of storage class definitions x 266)
+  (maximum number of authentication information definitions x 1010)
+  (maximum number of administrative topic definitions x 15000)
+  (total length of topic strings defined in administrative topic definitions)
```

Divide this value by 4096 to determine the number of records to specify in the cluster for the page set data set.

You do not need to allow for objects that are stored in the shared repository, but you must allow for objects that are stored or copied to page set zero (objects with a disposition of GROUP or QMGR).

The total number of objects that you can create is limited by the capacity of page set zero. The number of local queues that you can define is limited to 524 287.

Page sets 01 - 99

For page sets 01 - 99, the storage required for each page set is determined by the number and size of the messages stored on that page set. (Messages on shared queues are not stored on page sets.)

Divide this value by 4096 to determine the number of records to specify in the cluster for the page set data set.

Calculating the storage requirement for messages

This section describes how messages are stored on pages. Understanding this can help you calculate how much page set storage you must define for your messages. To calculate the approximate space required for all messages on a page set you must consider maximum queue depth of all the queues that map to the page set and the average size of messages on those queues.

Note: The sizes of the structures and control information given in this section are liable to change between major releases. For details specific to your release of IBM MQ, refer to SupportPac MP16 - IBM MQ per z/OS Capacity planning & tuning and Famiglia IBM MQ - Report sulle prestazioni

You must allow for the possibility that message "gets" might be delayed for reasons outside the control of IBM MQ (for example, because of a problem with your communications protocol). In this case, the "put" rate of messages might far exceed the "get" rate. This can lead to a large increase in the number of messages stored in the page sets and a consequent increase in the storage size demanded.

Each page in the page set is 4096 bytes long. Allowing for fixed header information, each page has 4057 bytes of space available for storing messages.

When calculating the space required for each message, the first thing you must consider is whether the message fits on one page (a short message) or whether it needs to be split over two or more pages (a long message). When messages are split in this way, you must allow for additional control information in your space calculations.

For the purposes of space calculation, a message can be represented as the following:

The message header section contains the message descriptor and other control information, the size of which varies depending on the size of the message. The message data section contains all the actual message data, and any other headers (for example, the transmission header or the IMS bridge header).

A minimum of two pages are required for page set control information which, is typically less than 1% of the total space required for messages.

Short messages

A short message is defined as a message that fits on one page.

Small messages are stored one on each page.

Long messages

If the size of the message data is greater than 3596 bytes, but not greater than 4 MB, the message is classed as a long message. When presented with a long message, IBM MQ stores the message on a series of pages, and stores control information that points to these pages in the same way that it would store a short message. This is shown in Figure 41 on page 158:

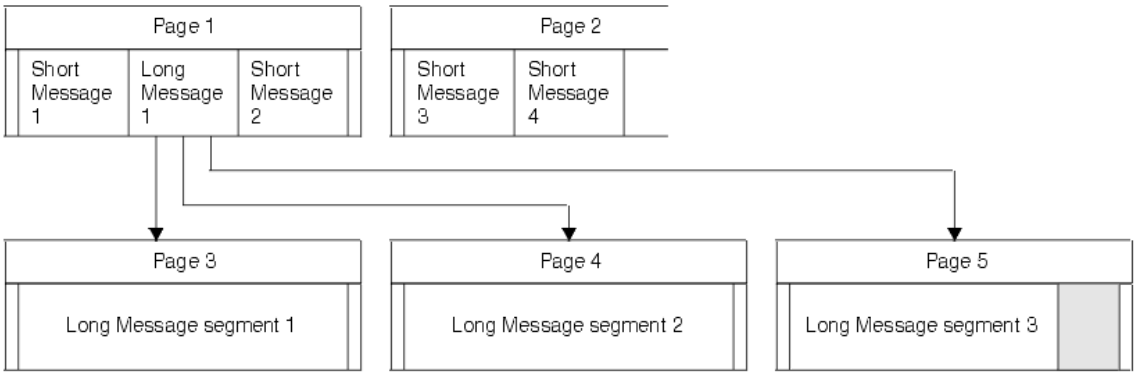


Figure 41. How IBM MQ stores long messages on page sets

Very long messages

Very long messages are messages with a size greater than 4 MB. These are stored so that each 4 MB uses 1037 pages. Any remainder is stored in the same way as a long message, as described above.

z/OS *Enabling dynamic page set expansion*

Page sets can be extended dynamically while the queue manager is running. A page set can have 123 extents, and can be spread over multiple disk volumes.

Each time a page set expands, a new data set extent is used. The queue manager continues to expand a page set when required, until the maximum number of extents has been reached, or until no more storage is available for allocation on eligible volumes.

Once page set expansion fails for one of the reasons above, the queue manager marks the page set for no further expansion attempts. This marking can be reset by altering the page set to EXPAND(SYSTEM).

Page set expansion takes place asynchronously to all other page set activity, when 90% of the existing space in the page set is allocated.

The page set expansion process formats the newly allocated extent and makes it available for use by the queue manager. However, none of the space is available for use, until the entire extent has been formatted. This means that expansion by a large extent is likely to take some time, and putting applications might 'block' if they fill the remaining 10% of the page set before the expansion has completed.

Sample thlqual.SCSQPROC(CSQ4PAGE) shows how to define the secondary extents.

To control the size of new extents, you use one of the following options of the EXPAND keyword of the DEFINE PSID and ALTER PSID commands:

- USER
- SYSTEM
- NONE

USER

Uses the secondary extent size specified when the page set was allocated. If a value was not specified, or if a value of zero was specified, dynamic page set expansion cannot occur.

Page set expansion occurs when the space in the page is 90% used, and is performed asynchronously with other page set activity.

This may lead to expansion by more than a single extent at a time.

Consider the following example: you allocate a page set with a primary extent of 100,000 pages and a secondary extent of 5000 pages. A message is put that requires 9999 pages. If the page set is already using 85,000 pages, writing the message crosses the 90% full boundary (90,000 pages). At this point, a further secondary extent is allocated to the primary extent of 100,000 pages, taking the page set size to 105,000 pages. The remaining 4999 pages of the message continue to be written. When the used page space reaches 94,500 pages, which is 90% of the updated page set size of 105,000 pages, another 5000 page extent is allocated, taking the page set size to 110,000 pages. At the end of the MQPUT, the page set has expanded twice, and 94,500 pages are used. None of the pages in the second page set expansion have been used, although they were allocated.

At restart, if a previously used page set has been replaced with a data set that is smaller, it is expanded until it reaches the size of the previously used data set. Only one extent is required to reach this size.

SYSTEM

Ignores the secondary extent size that was specified when the page set was defined. Instead, the queue manager sets a value that is approximately 10% of the current page set size. The value is rounded up to the nearest cylinder of DASD.

If a value was not specified, or if a value of zero was specified, dynamic page set expansion can still occur. The queue manager sets a value that is approximately 10% of the current page set size. The new value is rounded up depending on the characteristics of the DASD.

Page set expansion occurs when the space in the page set is approximately 90% used, and is performed asynchronously with other page set activity.

At restart, if a previously used page set has been replaced with a data set that is smaller, it is expanded until it reaches the size of the previously used data set.

NONE

No further page set expansion is to take place.

Related reference

ALTER PSID

DEFINE PSID

DISPLAYUSAGE

Defining your buffer pools

Use this topic to help plan the number of buffer pools you should define, and their settings.

This topic is divided into the following sections:

1. “Decide on the number of buffer pools to define” on page 160
2. “Decide on the settings for each buffer pool” on page 161
3. “Monitor the performance of buffer pools under expected load” on page 161
4. “Adjust buffer pool characteristics” on page 161

Decide on the number of buffer pools to define

You should define four buffer pools initially:

Buffer pool 0
Use for object definitions (in page set zero) and performance critical, system related message queues, such as the SYSTEM.CHANNEL.SYNCQ queue and the SYSTEM.CLUSTER.COMMAND.QUEUE and SYSTEM.CLUSTER.REPOSITORY.QUEUE queues.

However it is important to consider point “7” on page 162 in *Adjust buffer pool characteristics* if a large number of channels, or clustering, is to be used.

Use the remaining three buffer pools for user messages.

Buffer pool 1
Use for important long-lived messages.

Long-lived messages are those that remain in the system for longer than two checkpoints, at which time they are written out to the page set. If you have many long-lived messages, this buffer pool should be relatively small, so that page set I/O is evenly distributed (older messages are written out to DASD each time the buffer pool becomes 85% full).

If the buffer pool is too large, and the buffer pool never gets to 85% full, page set I/O is delayed until checkpoint processing. This might affect response times throughout the system.

If you expect few long-lived messages only, define this buffer pool so that it is sufficiently large to hold all these messages.

Buffer pool 2
Use for performance-critical, short-lived messages.

There is normally a high degree of buffer reuse, using few buffers. However, you should make this buffer pool large to allow for unexpected message accumulation, for example, when a server application fails.

Buffer pool 3
Use for all other (typically, performance noncritical) messages.

Queues such as the dead-letter queue, SYSTEM.COMMAND.* queues and SYSTEM.ADMIN.* queues can also be mapped to buffer pool 3.

Where virtual storage constraints exist, and buffer pools need to be smaller, buffer pool 3 is the first candidate for size reduction.

You might need to define additional buffer pools in the following circumstances:

- If a particular queue is known to require isolation, perhaps because it exhibits different behavior at various times.
  - Such a queue might either require the best performance possible under the varying circumstances, or need to be isolated so that it does not adversely affect the other queues in a buffer pool.
  - Each such queue can be isolated into its own buffer pool and page set.
- You want to isolate different sets of queues from each other for class-of-service reasons.

– Each set of queues might then require one, or both, of the two types of buffer pools 1 or 2, as described in Suggested definitions for buffer pool settings, necessitating creation of several buffer pools of a specific type.

Decide on the settings for each buffer pool

If you are using the four buffer pools described in “Decide on the number of buffer pools to define” on page 160, then Suggested definitions for buffer pool settings gives two sets of values for the size of the buffer pools.

The first set is suitable for a test system, the other for a production system or a system that will become a production system eventually. In all cases define your buffer pools with the **LOCATION**(*ABOVE*) attribute

Table 21. Suggested definitions for buffer pool settings

Definition setting	Test system	Production system
BUFFPOOL 0	1 050 buffers	50 000 buffers
BUFFPOOL 1	1 050 buffers	20 000 buffers
BUFFPOOL 2	1 050 buffers	50 000 buffers
BUFFPOOL 3	1 050 buffers	20 000 buffers

If you need more than the four suggested buffer pools, select the buffer pool (1 or 2) that most accurately describes the expected behavior of the queues in the buffer pool, and size it using the information in Suggested definitions for buffer pool settings.

Ensure that your MEMLIMIT is set high enough, so that all the buffer pools can be located above the bar.

Monitor the performance of buffer pools under expected load

You can monitor the usage of buffer pools by analyzing buffer pool performance statistics. In particular, you should ensure that the buffer pools are large enough so that the values of QPSTSOS, QPSTSTLA, and QPSTDMC remain at zero.

For further information, see Buffer manager data records.

Adjust buffer pool characteristics

Use the following points to adjust the buffer pool settings from “Decide on the settings for each buffer pool” on page 161, if required.

Use the performance statistics from “Monitor the performance of buffer pools under expected load” on page 161 as guidance.

1. If you are migrating from an earlier version of IBM MQ, only change your existing settings if you have more real storage available.
2. In general, bigger buffer pools are better for performance, and buffer pools can be much bigger if they are above the bar.

   However, at all times you should have sufficient real storage available so that the buffer pools are resident in real storage. It is better to have smaller buffer pools that do not result in paging, than big ones that do.

   Additionally, there is no point having a buffer pool that is bigger than the total size of the page sets that use it, although you should take into account page set expansion if it is likely to occur.
3. Aim for one page set per buffer pool, as this provides better application isolation.
4. If you have sufficient real storage, such that your buffer pools will never be paged out by the operating system, consider using page-fixed buffers in your buffer pool.

This is particularly important if the buffer pool is likely to undergo much I/O, as it saves the CPU cost associated with page-fixing the buffers before the I/O, and page-unfixing them afterwards.

5. There are several benefits to locating buffer pools above the bar even if they are small enough to fit below the bar. These are:
   - 31 bit virtual storage constraint relief - for example more space for common storage.
   - If the size of a buffer pool needs to be increased unexpectedly while it is being heavily used, there is less impact and risk to the queue manager, and its workload, by adding more buffers to a buffer pool that is already above the bar, than moving the buffer pool to above the bar and then adding more buffers.
6. Tune buffer pool zero and the buffer pool for short-lived messages (buffer pool 2) so that the 15% free threshold is never exceeded (that is, QPSTCBSL divided by QPSTNBUF is always greater than 15%). If more than 15% of buffers remain free, I/O to the page sets using these buffer pools can be largely avoided during normal operation, although messages older than two checkpoints are written to page sets.

   Attention: The optimum value for these parameters is dependent on the characteristics of the individual system. The values given are intended only as a guideline and might not be appropriate for your system.
7. SYSTEM.* queues which get very deep, for example SYSTEM.CHANNEL.SYNCQ, might benefit from being placed in their own buffer pool, if sufficient storage is available.

IBM MQ SupportPac MP16 - IBM MQ per z/OS Capacity planning & tuning provides further information about tuning buffer pools.

z/OS Planning your logging environment

Use this topic to plan the number, size and placement of the logs, and log archives used by IBM MQ.

Logs are used to:

- Write recovery information about persistent messages
- Record information about units of work using persistent messages
- Record information about changes to objects, such as define queue
- Backup CF structures

and for other internal information.

The IBM MQ logging environment is established using the system parameter macros to specify options, such as: whether to have single or dual active logs, what media to use for the archive log volumes, and how many log buffers to have.

These macros are described in Create the bootstrap and log data sets and Tailor your system parameter module.

Note: If you are using queue sharing groups, ensure that you define the bootstrap and log data sets with SHAREOPTIONS(2 3).

This section contains information about the following topics:

z/OS *Log data set definitions*

Use this topic to decide on the most appropriate configuration for your log data sets.

This topic contains information to help you answer the following questions:

- Should your installation use single or dual logging?
- How many active log data sets do you need?
- “How large should the active logs be?” on page 164
- Active log placement

- “Active log encryption with z/OS data set encryption” on page 165

Should your installation use single or dual logging?

In general you should use dual logging for production, to minimize the risk of losing data. If you want your test system to reflect production, both should use dual logging, otherwise your test systems can use single logging.

With single logging data is written to one set of log data sets. With dual logging data is written to two sets of log data sets, so in the event of a problem with one log data set, such as the data set being accidentally deleted, the equivalent data set in the other set of logs can be used to recover the data.

With dual logging you require twice as much DASD as with single logging.

If you are using dual logging, then also use dual BSDSs and dual archiving to ensure adequate provision for data recovery.

Dual active logging adds a small performance cost.

Attention: Use of disk mirroring technologies, such as Metro Mirror, are not necessarily a replacement for dual logging and dual BSDS. If a mirrored data set is accidentally deleted, both copies are lost.

If you use persistent messages, single logging can increase maximum capacity by 10-30% and can also improve response times.

Single logging uses 2 - 310 active log data sets, whereas dual logging uses 4 - 620 active log data sets to provide the same number of active logs. Thus single logging reduces the amount of data logged, which might be important if your installation is I/O constrained.

How many active log data sets do you need?

The number of logs depends on the activities of your queue manager. For a test system with low throughput, three active log data sets might be suitable. For a high throughput production system you might want the maximum number of logs available, so, if there is a problem with offloading logs you have more time to resolve the problems.

You must have at least three active log data sets, but it is preferable to define more. For example, if the time taken to fill a log is likely to approach the time taken to archive a log during peak load, define more logs.

Note: Page sets and active log data sets are eligible to reside in the extended addressing space (EAS) part of an extended address volumes (EAV) and an archive log dataset can also reside in the EAS.

You should also define more logs to offset possible delays in log archiving. If you use archive logs on tape, allow for the time required to mount the tape.

Consider having enough active log space to keep a day's worth of data, in case the system is unable to archive because of lack of DASD or because it cannot write to tape. If all the active logs fill up, then IBM MQ is unable to process persistent messages or transactions. It is very important to have enough active log space.

It is possible to dynamically define new active log data sets as a way of minimizing the effect of archive delays or problems. New data sets can be brought online rapidly, using the **DEFINE LOG** command to avoid queue manager 'stall' due to lack of space in the active log.

If you want to define more than 31 active log data sets, you must configure your logging environment to use a version 2 format BSDS. Once a version 2 format BSDS is in use, up to 310 active log data sets can be defined for each log copy ring. See “Planning to increase the maximum addressable log range” on page 174 for information on how you convert to a version 2 format BSDS.

You can tell whether your queue manager is using a version 2 or higher BSDS, either by running the print log map utility (CSQJU004), or from the CSQJ034I message issued during queue manager initialization.

An end of log RBA range of FFFFFFFFFFFFFFFF, in the CSQJ034I message, indicates that a version 2, or higher, format BSDS is in use. An end of log RBA range of 0000FFFFFFFFFFFF, in the CSQJ034I message, indicates that a version 1 format BSDS is in use.

When a queue manager is using a version 2, or higher, format BSDS, it is possible to use the **DEFINE LOG** command to dynamically add more than 31 active log data sets to a log copy ring.

How large should the active logs be?

The maximum supported active log size, when archiving to disk or to tape, is 4 GB.

You should create active logs of at least 1 GB in size for production and test systems.

Important: You need to be careful when allocating data sets, because IDCAMS rounds up the size you allocate.

To allocate a 3 GB log specify one of the following options:

- Cylinders(4369)
- Megabytes(3071)
- TRACKS(65535)
- RECORD(786420)

Any one of these allocates 2.99995 GB.

To allocate a 4GB log specify one of the following options:

- Cylinders(5825)
- Megabytes(4095)
- TRACKS(87375)
- RECORD(1048500)

Any one of these allocates 3.9997 GB.

When using striped data sets, where the data set is spread across multiple volumes, the specified size value is allocated on each DASD volume used for striping. So, if you want to use 4 GB logs and four volumes for striping, you should specify:

- CYLinders(1456)
- Megabytes(1023)

Setting these attributes allocates 4*1456 = 5824 Cylinders or 4 * 1023 = 4092 Megabytes.

Note: Striping is supported when using extended format data sets. This is usually set by the storage manager.

See Increasing the size of the active log for information on carrying out the procedure.

Active log placement

You should work with your storage management team to set up storage pools for the queue managers. You need to consider:

- A naming convention, so the queue managers use the correct SMS definitions.
- Space required for active and archive logs. Your storage pool should have enough space for the active logs from a whole day.
- Performance and resilience to failures.

For performance reasons you should consider striping your active log data sets. The I/O is spread across multiple volumes and reduces the I/O response times, leading to higher throughput. See the preceding text for information about allocating the size of the active logs when using striping.

You should review the I/O statistics using reports from RMF or a similar product. Perform the review of these statistics monthly (or more frequently) for the IBM MQ data sets, to ensure there are no delays due to the location of the data sets.

In some situations, there can be much IBM MQ page set I/O, and this can impact the IBM MQ log performance if they are located on the same DASD.

If you use dual logging, ensure that each set of active and archive logs is kept apart. For example, allocate them on separate DASD subsystems, or on different devices.

This reduces the risk of them both being lost if one of the volumes is corrupted or destroyed. If both copies of the log are lost, the probability of data loss is high.

When you create a new active log data, set you should preformat it using CSQJUFMT. If the log is not preformatted, the queue manager formats the log the first time it is used, which impacts the performance.

With older DASD with large spinning disks, you had to be careful which volumes were used to get the best performance.

With modern DASD, where data is spread over many PC sized disks, you do not need to worry so much about which volumes are used.

Your storage manager should be checking the enterprise DASD to review and resolve any performance problems. For availability, you might want to use one set of logs on one DASD subsystem, and the dual logs on a different DASD subsystem.

Active log encryption with z/OS data set encryption

You can apply the z/OS data set encryption feature to active log data sets for queue managers running at IBM MQ for z/OS 9.1.4 or later.

You must allocate these active log data sets with EXTENDED attributes, and a data set key label that ensures the data is AES encrypted.

See the section, confidentiality for data at rest on IBM MQ for z/OS with data set encryption. for more information.

z/OS *Using MetroMirror with IBM MQ*

IBM Metro Mirror, previously known as Synchronous Peer to Peer Remote Copy (PPRC), is a synchronous replication solution between two storage subsystems, where write operations are completed on both the primary and secondary volumes before the write operation is considered to be complete. Metro Mirror can be used in environments that require no data loss in the event of a storage subsystem failure.

Supported data set types

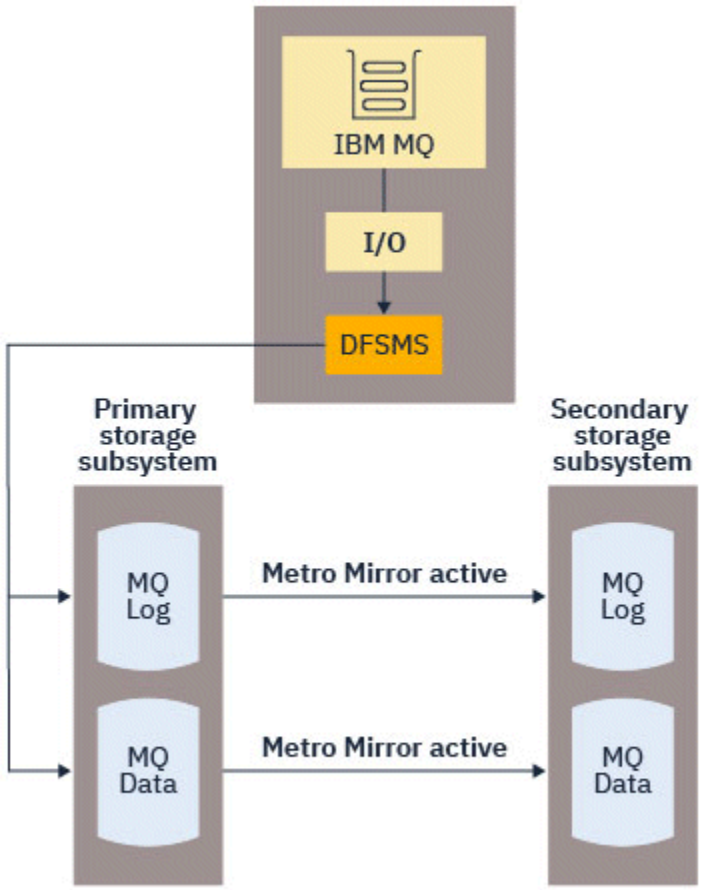
All of the following IBM MQ data set types can be replicated using Metro Mirror. However, exactly which ones are replicated depends on the availability requirements of your enterprise:

- Active logs
- Archive logs
- Bootstrap data set (BSDS)
- Page sets
- Shared message data set (SMDS)
- Data sets used for configuration, for example, in the CSQINP* DD cards on the MSTR JCL

Using zHyperWrite with IBM MQ active logs

When a write is made to a data set that is replicated using Metro Mirror, the write is first made to the primary volume, and then replicated to the secondary volume. This replication is done by the storage subsystem and is transparent to the application that issued the write, for example IBM MQ.

This process is illustrated in the following diagram.

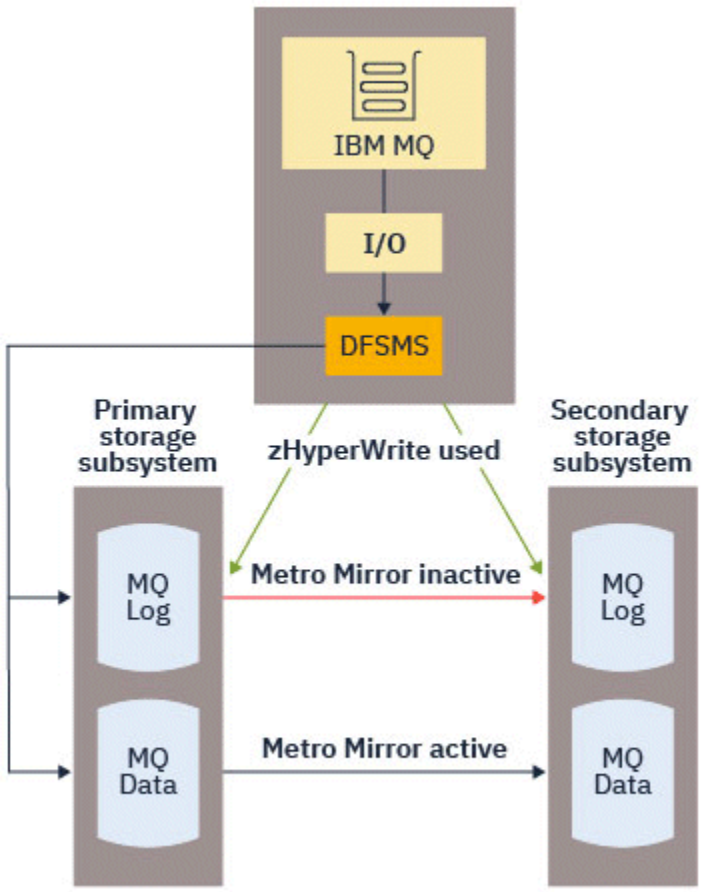


Because both writes to the primary and secondary storage subsystems need to complete before the write returns to IBM MQ, use of Metro Mirror can have a performance impact. You need to balance this performance impact against the availability benefits of using Metro Mirror.

The IBM MQ active logs are most sensitive to the performance impact of using Metro Mirror. IBM MQ allows use of zHyperWrite with the active logs to help reduce this performance impact.

zHyperWrite is a storage subsystem technology that works with z/OS to reduce the performance impact of writes made to data sets that are replicated using Metro Mirror. When zHyperWrite is used, the write to the primary and secondary volumes are issued in parallel at the Data Facility Storage Management Subsystem (DFSMS) level, instead of sequentially at the storage subsystem level, thereby reducing the performance impact.

The following diagram illustrates zHyperWrite being used for the active logs, and Metro Mirror being used for the other IBM MQ data set types. Note that if a zHyperWrite write fails, DFSMS will transparently reissue the write using Metro Mirror.



zHyperWrite on IBM MQ, is supported only on the active log data sets.

In order to use zHyperWrite with the active logs, you need to:

- Configure IBM MQ to use zHyperWrite, and
- The active logs need to be on zHyperWrite capable volumes

You can configure IBM MQ to use zHyperWrite by using one of the following methods:

- Specify ZHYWRITE(YES) in the system parameter module.
- Issue the command SET LOG ZHYWRITE(YES).

Set the following conditions for active log data sets to be on zHyperWrite capable volumes:

- Enable the volumes for Metro Mirror, and the volumes support zHyperWrite
- Ensure that the volumes are HyperSwap enabled
- Specify HYPERWRITE=YES in the IECIOSxx parameter

V 9.4.0 Prior to IBM MQ 9.4.0, if all the preceding conditions are met, then writes to the active logs are enabled for zHyperWrite. If one, or more, of these conditions are not met, IBM MQ writes to the active logs as normal, and Metro Mirror replicates the writes if it is configured.

V 9.4.0 From IBM MQ 9.4.0, if ZHYWRITE(YES) is specified, then IBM MQ always attempts to use zHyperWrite when writing to the active logs, regardless of whether the logs are on zHyperWrite capable volumes. If the logs are not on zHyperWrite capable volumes then Metro Mirror replicates the writes if it is configured. There are no negative effects of attempting to use zHyperWrite if the logs are not on zHyperWrite capable volumes

Notes:

- IBM MQ does not require that all active log data sets are on zHyperWrite capable volumes.

  If IBM MQ detects that some active log data sets are on zHyperWrite capable volumes, and others are not, it issues message CSQJ166E and carries on processing.
- IBM MQ checks whether active log data sets are zHyperWrite capable when the data sets are first opened.

  Log data sets are opened either at queue manager start up, or when dynamically adding using the DEFINE LOG command. If the log data sets are made zHyperWrite capable while a queue manager has them open, the queue manager will not detect this until it has been restarted.

You can use the output of the DISPLAY LOG command to indicate whether the current active log data sets are zHyperWrite capable. The following example shows that both of the data sets are zHyperWrite capable. If the queue manager has been configured with ZHYWRITE(YES), writes to these logs would be enabled for zHyperWrite:

```
Copy %Full zHyperWrite DSName
 1      4  CAPABLE     MQTST.SUBSYS.MQDL.LOGCOPY1.DS001
 2      4  CAPABLE     MQTST.SUBSYS.MQDL.LOGCOPY2.DS001
```

V 9.4.0 z/OS ***Velocità di trasmissione dei log più veloce con zHyperLink***

zHyper La tecnologia di collegamento è progettata per ridurre la latenza di input/output (I/O) fornendo un percorso di comunicazione veloce, affidabile e diretto tra la CPU e il dispositivo I/O.

Panoramica dizHyper Collegamento

zHyperLink può migliorare la velocità di trasmissione dei log attivi e ridurre il tempo di transazione IBM MQ fino a 3.5 volte. Questo obiettivo viene raggiunto installandozHyper Collegare gli adattatori suz/OS ospitare, selezionareIBM hardware di archiviazione e collegarli utilizzandozHyper Collegare i cavi. Ciò crea una connessione punto a punto tra la CPU e il dispositivo I/O, che riduce il tempo di risposta I/O fino a 10 volte rispetto aIBM z FICON® ad alte prestazioni (zHPF). Un tempo di risposta così basso si ottiene utilizzando richieste I/O sincrone.

I vantaggi dell'I/O sincrono rispetto all'I/O asincrono

ILIBM MQ L'attività del logger consiste in un ciclo in attesa del successivo pezzo di dati che deve essere scritto nel log. Quando i dati sono disponibili, il logger pianifica la scrittura, attende il completamento e quindi passa al dato successivo.

L'I/O tradizionale è più lento della CPU, quindi è più efficiente eseguire l'I/O in modo asincrono per liberare la CPU per altre attività. Pertanto, l'I/O asincrono tradizionale richiede la sospensione dell'attività del logger fino al completamento della scrittura. Una volta completata la scrittura, l'attività del logger deve attendere che una CPU diventi disponibile, aggiungendo un breve ritardo di reinvio, nonché ritardi causati dal ripopolamento della cache della CPU.

zHyper Link fornisce tempi di I/O molto più rapidi, più vicini alla velocità della CPU, quindi conzHyper Link, l'I/O può essere eseguito in modo sincrono, il che significa che l'attività del logger non viene sospesa durante l'operazione di scrittura, eliminando i ritardi legati al reinvio e alla cache.

Mentre è in corso la scrittura, l'attività del logger utilizza ancora attivamente la CPU, il che aumenta l'utilizzo della CPU rispetto all'I/O tradizionale.

Se il gestore code tenta di utilizzarezHyper Collegamento e ilzHyper La scrittura del collegamento fallisce, ad esempio a causa di problemi di configurazione, quindi il gestore code ritorna in modo trasparente all'I/O tradizionale.

Requisiti hardware minimi

- IBM z14 o versioni successive
- DS8880 o successive

Requisiti software

- zHyperLink Express è supportato su z/OS 2.3 o versioni successive.
- L'immagine z/OS deve essere eseguita in una LPAR, non come guest in IBM z/VM®.
- zHyperLink richiede l'abilitazione di IBM z High-Performance FICON (zHPF).

Utilizzo di zHyperLink con i log attivi IBM MQ

Per poter utilizzarezHyper Collegamento con i log attivi di un gestore code, è necessario:

- ConfiguraIBM MQ usarezHyper Collegamento e
- Assicurarsi che i registri attivi siano attivizHyper Collegare volumi compatibili.

  Vedere Iniziare conIBMzHyper Collegamento perz/OS per maggiori informazioni.

Puoi configurareIBM MQ usarezHyper Collegare utilizzando uno dei seguenti metodi:

- Specificare ZHYLINK(YES) nei parametri di log.
- Immettere il comando SET LOG ZHYLINK(YES).

Note:

- zHyperLink richiede che la scrittura zHypersia attivata. Ciò significa che per utilizzare ZHYLINK, ZHYWRITE deve essere attivato anche nei parametri di log. Quando si specifica solo ZHYLINK (YES) quando ZHYWRITE (NO) è impostato sul gestore code, il parametro ZHYWRITE sovrascrive automaticamente su YES.
- Il tentativo di impostare esplicitamente ZHYLINK (YES) con ZHYWRITE (NO) provoca un completamento anomalo del comando SET LOG.
- L'impostazione di ZHYLINK=YES nelle ZPRM sovrascrive ZHYWRITE su YES.

Se riscontri dei problemi, vedi Risoluzione dei problemi zHyperLink per ulteriori informazioni.

IBM MQ non richiede che tutti i dataset di log attivi si trovino su volumi compatibili con zHyperLink, ma si consiglia di farlo. Se IBM MQ rileva che alcuni dataset di log attivi si trovano su volumi compatibili con zHyperLink e altri no, emette il messaggio CSQJ601E e continua l'elaborazione.

IBM MQ controlla se i dataset di log attivi sono compatibili con zHyperLink quando i dataset vengono aperti per la prima volta. I dataset di log vengono aperti all'avvio del gestore code o quando vengono aggiunti dinamicamente utilizzando il comando DEFINE LOG . Se i dataset di log sono resi zHyperLink compatibili mentre un gestore code li ha aperti, il gestore code non li rileva fino a quando non viene riavviato.

Se viene specificato ZHYLINK (YES), IBM MQ tenta sempre di utilizzare zHyperLink durante la scrittura nei log attivi, indipendentemente dal fatto che i log si trovino su volumi compatibili con zHyperLink. Non ci sono effetti negativi nel tentativo di utilizzare il link zHyperse i log non si trovano su volumi compatibili con il link zHyper.

È possibile utilizzare l'output del comando DISPLAY LOG per indicare lo stato di zHyperLink per i dataset di log attivi correnti:

```
Copy %Full zHyperWrite    Encrypted          DSName
  1     81 YES          NO                MQTST.SUBSYS.MQDL.LOGCOPY1.DS001
  2     81 YES            NO                MQTST.SUBSYS.MQDL.LOGCOPY2.DS001
Copy zHyperLink
  1  YES
  2  YES
```

Lo stato del collegamento zHyperè uno dei seguenti:

SÌ
zHyperLink è abilitato sul gestore code e verrà tentato su tutte le scritture.

No
zHyperIl link non è abilitato sul gestore code e sul dataset **non è** su zHyperVolumi con capacità di link.

supportato
zHyperIl link non è abilitato sul gestore code e il set di dati **è** su un volume compatibile con link zHyper.

Sono disponibili numerose statistiche SMF aggiuntive per il monitoraggio e la comprensionezHyper Prestazioni del collegamento; VederezHyper Statistiche sui collegamenti per dettagli.

Scrivi sessioni

Quando si usazHyper Collegamento, uno o piùzHyper Le sessioni di scrittura dei collegamenti vengono stabilite con DASD. L'attuale DASD supporta un massimo di 64 sessioni di scrittura simultanee, quindi è necessario considerare attentamente quali gestori code abilitarezHyper Collegamento e se altri sottosistemi, come ad esempioDb2 stanno anche utilizzandozHyper Link per scrivere sullo stesso DASD. Se esaurisci le sessioni di scrittura disponibili, il gestore code torna automaticamente all'utilizzo dell'I/O asincrono tradizionale.

Puoi calcolare il numero dizHyper Collegare le sessioni di scrittura come segue:

```
Number of log copies (either 1 or 2) * number of stripes per log copy * 2
if Metro Mirror (PPRC) is used.
```

Pertanto, un gestore code in modalità di registrazione singola con uno stripe e noMetro Mirror utilizza una singola sessione di scrittura. Un gestore code in modalità dual logging, con due stripe e PPRC utilizza 8 sessioni di scrittura.

Nota: MentreMetro Mirror comporta l'utilizzo del doppio delle sessioni di scrittura, tali sessioni di scrittura vengono suddivise equamente tra i due DASD con mirroring.

z/OS *Planning your log archive storage*

Use this topic to understand the different ways of maintaining your archive log data sets.

You can place archive log data sets on standard-label tapes, or DASD, and you can manage them by data facility hierarchical storage manager (DFHSM). Each z/OS logical record in an archive log data set is a VSAM control interval from the active log data set. The block size is a multiple of 4 KB.

Archive log data sets are dynamically allocated, with names chosen by IBM MQ. The data set name prefix, block size, unit name, and DASD sizes needed for such allocations are specified in the system parameter module. You can also choose, at installation time, to have IBM MQ add a date and time to the archive log data set name.

It is not possible to specify with IBM MQ, specific volumes for new archive logs, but you can use Storage Management routines to manage this. If allocation errors occur, offloading is postponed until the next time offloading is triggered.

If you specify dual archive logs at installation time, each log control interval retrieved from the active log is written to two archive log data sets. The log records that are contained in the pair of archive log data sets are identical, but the end-of-volume points are not synchronized for multivolume data sets.

Should your archive logs reside on tape or DASD?

When deciding whether to use tape or DASD for your archive logs, there are a number of factors that you should consider:

- Review your operating procedures before deciding about tape or disk. For example, if you choose to archive to tape, there must be enough tape drive when they are required. After a disaster, all subsystems might want tape drives and you might not have as many free tape drives as you expect.
- During recovery, archive logs on tape are available as soon as the tape is mounted. If DASD archives have been used, and the data sets migrated to tape using hierarchical storage manager (HSM), there is a

delay while HSM recalls each data set to disk. You can recall the data sets before the archive log is used. However, it is not always possible to predict the correct order in which they are required.

- When using archive logs on DASD, if many logs are required (which might be the case when recovering a page set after restoring from a backup) you might require a significant quantity of DASD to hold all the archive logs.
- In a low-usage system or test system, it might be more convenient to have archive logs on DASD to eliminate the need for tape mounts.
- Both issuing a RECOVER CFSTRUCT command, and backing out a persistent unit of work, result in the log being read backwards. Tape drives with hardware compression perform badly on operations that read backwards. Plan sufficient log data on DASD to avoid reading backwards from tape.

Archiving to DASD offers faster recoverability but is more expensive than archiving to tape. If you use dual logging, you can specify that the primary copy of the archive log go to DASD and the secondary copy go to tape. This increases recovery speed without using as much DASD, and you can use the tape as a backup.

See “Changing the storage medium for archive logs” on page 172 for details of how you archive your logs from tape to DASD, and how you carry out the reverse process.

Archiving to tape

If you choose to archive to a tape device, IBM MQ can extend to a maximum of 20 volumes.

If you are considering changing the size of the active log data set so that the set fits on one tape volume, note that a copy of the BSDS is placed on the same tape volume as the copy of the active log data set. Adjust the size of the active log data set downward to offset the space required for the BSDS on the tape volume.

If you use dual archive logs on tape, it is typical for one copy to be held locally, and the other copy to be held off-site for use in disaster recovery.

Archiving to DASD volumes

IBM MQ requires that you catalog all archive log data sets allocated on non-tape devices (DASD). If you choose to archive to DASD, the CATALOG parameter of the CSQ6ARVP macro must be YES. If this parameter is NO, and you decide to place archive log data sets on DASD, you receive message CSQJ072E each time an archive log data set is allocated, although IBM MQ still catalogs the data set.

If the archive log data set is held on DASD, the archive log data sets can extend to another volume; multivolume is supported.

If you choose to use DASD, make sure that the primary space allocation (both quantity and block size) is large enough to contain either the data coming from the active log data set, or that from the corresponding BSDS, whichever is the larger of the two.

This minimizes the possibility of unwanted z/OS X'B37' or X'E37' abend codes during the offload process. The primary space allocation is set with the PRIQTY (primary quantity) parameter of the CSQ6ARVP macro.

Archive log data sets can exist on large or extended-format sequential data sets. SMS ACS routines now use DSNTYPE(LARGE) or DSNTYPE(EXT).

IBM MQ supports allocation of archive logs as extended format data sets. When extended format is used, the maximum archive log size is increased from 65535 tracks to the maximum active log size of 4GB. Archive logs are eligible for allocation in the extended addressing space (EAS) of extended address volumes (EAV).

Where the required hardware and software levels are available, allocating archive logs to a data class defined with COMPACTION using zEDC might reduce the disk storage required to hold archive logs. For more information, see IBM MQ for z/OS: Reducing storage occupancy with IBM zEnterprise Data Compression (zEDC) and zEnterprise Data Compression (zEDC) for more information.

The z/OS data set encryption feature can be applied to archive logs for queue managers running on IBM MQ. These archive logs must be allocated through Automatic Class Selection (ACS) routines to a data class defined with EXTENDED attributes, and a data set key label that ensures the data is AES encrypted.

Using SMS with archive log data sets

If you have MVS/DFP storage management subsystem (DFSMS) installed, you can write an Automatic Class Selection (ACS) user-exit filter for your archive log data sets, which helps you convert them for the SMS environment.

Such a filter, for example, can route your output to a DASD data set, which DFSMS can manage. You must exercise caution if you use an ACS filter in this manner. Because SMS requires DASD data sets to be cataloged, you must make sure the CATALOG DATA field of the CSQ6ARVP macro contains YES. If it does not, message CSQJ072E is returned; however, the data set is still cataloged by IBM MQ.

For more information about ACS filters, see Data sets that DFSMShsm dynamically allocates during aggregate backup processing.

z/OS *Changing the storage medium for archive logs*

The procedure for changing the storage medium used by archive logs.

About this task

This task describes how to change the storage medium used for archive logs, for example moving from archiving to tape to archiving to DASD.

You have a choice of how to make the changes:

1. Make the changes only using the CSQ6ARVP macro so that they are applied from the next time the queue manager restarts.
2. Make the changes using the CSQ6ARVP macro, and dynamically using the SET ARCHIVE command. This means that the changes apply from the next time the queue manager archives a log file, and persist after the queue manager restarts.

Procedure

1. Changing so archive logs are stored on DASD instead of tape:
   a) Read the section “Archiving to DASD volumes” on page 171 and review the CSQ6ARVP parameters.
   b) Make changes to the following parameters in CSQ6ARVP
      - Update the UNIT and, if necessary, the UNIT2 parameters.
      - Update the BLKSIZE parameter, as the optimal setting for DASD differs from tape.
      - Set the PRIQTY and SECQTY parameters to be large enough to hold the largest of the active log or BSDS.
      - Set the CATALOG parameter to be YES.
      - Confirm the ALCUNIT setting is what you want. You should use BLK, because it is independent of the device type.
      - Set the ARCWTOR parameter to NO if it is not already.
2. Changing so archive logs are stored on tape instead of DASD:
   a) Read the section “Archiving to tape” on page 171, and review the CSQ6ARVP parameters.
   b) Make changes to the following parameters in CSQ6ARVP:
      - Update the UNIT and, if necessary, the UNIT2 parameters.
      - Update the BLKSIZE parameter, as the optimal setting for tape differs from DASD.

- Confirm the ALCUNIT setting is what you want. You should use BLK, because it is independent of the device type.
- Review the setting of the ARCWTOR parameter.

z/OS *How long do I need to keep archive logs*

Use the information in this section to help you plan your backup strategy.

You specify how long archive logs are kept in days , using the ARCRETN parameter in USING CSQ6ARVP or the SET SYSTEM command. After this period the data sets can be deleted by z/OS.

You can manually delete archive log data sets when they are no longer needed.

- The queue manager might need the archive logs for recovery.

  The queue manager can only keep the most recent 1000 archives in the BSDS, When the archive logs are not in the BSDS they cannot be used for recovery, and are only of use for audit, analysis, or replay type purposes.
- You might want to keep the archive logs so that you can extract information from the logs. For example, extracting messages from the log, and reviewing which user ID put or got the message.

The BSDS contains information on logs and other recovery information. This data set is a fixed size. When the number of archive logs reaches the value of MAXARCH in CSQ6LOGP, or when the BSDS fills up, the oldest archive log information is overwritten.

There are utilities to remove archive log entries from the BSDS, but in general, the BSDS wraps and overlays the oldest archive log record.

When is an archive log needed

You need to back up your page sets regularly. The frequency of backups determines which archive logs are needed in the event of losing a page set.

You need to back up your CF structures regularly. The frequency of backups determines which archive logs are needed in the event of losing data in the CF structure.

The archive log might be needed for recovery. The following information explains when the archive log might be needed, where there are problems with different IBM MQ resources.

Loss of a page set

You must recover your system from your backup and restart the queue manager.

You need the logs from when the backup was taken, as well as up to three log data sets prior to the backup being taken.

All LPARs lose connectivity to a CF structure, or the structure is unavailable

Use the RECOVER CFSTRUCT command to recover the structure.

Structure recovery requires the logs from all queue managers that have accessed the structure since the last backup (back to the time when the backup was taken) plus the structure backup itself in the log of the queue manager that took the backup.

If you have been doing frequent backups of the CF structures, the data should be in active logs, and you should not need archive logs.

If there is no recent backup of the CF structure, you might need archive logs.

Note: All non persistent messages will be lost; all persistent messages will be re-created by performing the following tasks:

1. Reading the last CF structure backup from the log
2. Reading the logs from all queue managers that have used the structure
3. Merging updates since the backup

Administration structure rebuild
If you need to rebuild the administration structure, the information is read from the last checkpoint of the log for each queue manager in the QSG.

If a queue manager is not active, another queue manager in the QSG reads the log.

You should not need archive logs.

Loss of an SMDS data set
If you lose an SMDS data set, or the data set gets corrupted, the data set becomes unusable and the status for it is set to FAILED. The CF structure is unchanged.

In order to restore the SMDS data set, you need to:

1. Redefine the SMDS data set, and
2. Recover the CF structure by issuing the RECOVER CFSTRUCT command.

   Note: All non persistent messages on the CF structure will be lost; all persistent messages will be restored.

The requirement for queue manager logs is the same as for recovering from a structure that is unavailable.

z/OS *Planning to increase the maximum addressable log range*

You can increase the maximum addressable log range by configuring your queue manager to use a larger log relative byte address (RBA).

The log RBA size was increased from IBM MQ for z/OS 8.0. For an overview of this change, see Larger log Relative Byte Address.

Queue managers created at IBM MQ 9.3.0 or later, have 8 byte log RBA enabled by default and, therefore, do not require conversion.

You can convert your queue managers to use 8 byte log RBA values at any time. A queue sharing group can contain some queue managers with 8 byte log RBA enabled, and some queue managers with 6 byte log RBA enabled.

Undoing the change

The change cannot be backed out.

How long does it take?

The change requires a queue manager restart. Stop the queue manager, run the CSQJUCNV utility against the bootstrap data set (BSDS), or data sets, to create new data sets, rename these bootstrap data sets, and restart the queue manager. The CSQJUCNV utility usually takes a few seconds to run.

What impact does this have?

- With 8 byte log RBA in use, every write of data to the log data sets has additional bytes. Therefore, for a workload consisting of persistent messages there is a small increase in the amount of data written to the logs.
- Data written to a page set, or coupling facility (CF) structure, is not affected.

Related tasks
Implementing the larger log Relative Byte Address

z/OS Planning your channel initiator

The channel initiator provides communications between queue managers, and runs in its own address space.

There are two types of connections:

1. Application connections to a queue manager over a network. These are known as client channels.
2. Queue manager to queue manager connections. These are known as MCA channels.

Listeners

A channel listener program listens for incoming network requests and starts the appropriate channel when that channel is needed. To process inbound connections the channel initiator needs at least one IBM MQ listener task configured. A listener can either be a TCP listener, or a LU 6.2 listener.

Each listener requires a TCP port or LU name.

Note that you can have more than one listener for each channel initiator.

TCP/IP

A channel initiator can operate with more than one TCP stack on the same z/OS image. For example, one TCP stack could be for internal connections, and another TCP stack for external connections.

When you define an output channel:

1. You set the destination host and port of the connection. This can be either:
   - an IP address, for example `10.20.4.6`
   - a host name, for example `mvs-prod.myorg.com`

   If you use a host name to specify the destination, IBM MQ uses the Domain Name System (DNS) to resolve the IP address of the destination.
2. If you are using multiple TCP stacks you can specify the **LOCLADDR** parameter on the channel definition, which specifies the IP Stack address to be used.

You should plan to have a highly available DNS server, or DNS servers. If the DNS is not available, outbound channels might not be able to start, and channel authentication rules that map an incoming connection using a host name cannot be processed.

APPC and LU 6.2

If you are using APPC, the channel initiator needs an LU name, and configuration in APPC.

Queue sharing groups

To provide a single system image, and allow an incoming IBM MQ connection request to go to any queue manager in the queue sharing group, you need to do some configuration. For example:

1. A hardware network router. This router has one IP address seen by the enterprise, and can route the initial request to any queue manager connected to this hardware.
2. A Virtual IP address (VIPA). An enterprise wide IP address is specified, and that address can be routed to any one of the TCP stacks in a sysplex. The TCP stack can then route it to any listening queue manager in the sysplex.

Protecting IBM MQ traffic

You can configure IBM MQ to use TLS connections to protect data on the wire. To use TLS you need to use digital certificates and key rings.

You also need to work with the personnel at the remote end of the channel, to ensure that you have compatible IBM MQ definitions and compatible certificates.

You can control which connections can connect to IBM MQ and the user ID, based on

- IP address

- Client user ID
- Remote queue manager, or
- Digital certificate (see Channel Authentication Records)

It is also possible to restrict client applications by ensuring that they supply a valid user ID and password (see Connection Authentication).

You can get the channel initiator working, and then configure each channel to use TLS, one at a time.

Monitoring the channel initiator

There are MQSC commands that give information about the channel initiator and channels:

- The DISPLAY CHINIT command gives information about the channel initiator, and active listeners.
- The DISPLAY CHSTATUS command displays the activity and status of a channel.

The channel initiator can also produce SMF records with information about the channel initiator tasks and channel activity. See “Planning for channel initiator SMF data” on page 177 for more information.

The channel initiator emits messages to the job log when channels start and stop. Automation in your enterprise can use these messages to capture status. As some channels are active for only a few seconds, many messages can be produced. You can suppress these messages either by using the z/OS message processing facility, or by setting **EXCLMSG** with the SET SYSTEM command.

Configuring your IBM MQ channel definitions

When you have many queue managers connected together it can be hard to manage all the object definitions. Using IBM MQ clustering can simplify this.

You specify two queue managers as full repositories. Other queue managers need one connection to, and one connection from, one of the repositories. When connections to other queue managers are needed, the queue manager creates and starts channels automatically.

If you are planning to have a large number of queue managers in a cluster, you should plan to have queue managers that act as dedicated repositories and have no application traffic.

See “Pianificazione delle code e dei cluster distribuiti” on page 20 for more information.

Actions before you configure the channel initiator

1. Decide if you are using TCP/IP or APPC.
2. If you are using TCP, allocate at least one port for IBM MQ.
3. If you need a a DNS server, configure the server to be highly available if required.
4. If you are using APPC, allocate an LU name, and configure APPC.

Actions after you have configured the channel initiator, before you go into production

1. Plan what connections you will have:
   a. Client connections from remote applications.
   b. MCA channels to and from other queue managers. Typically you have a channel to and from each remote queue manager.
2. Set up clustering, or join an existing clustering environment.
3. Consider whether you need to use multiple TCP stacks, VIPA, or an external router for availability in front of the channel initiator.
4. If you are planning on using TLS:
   a. Set up the key ring

b. Set up certificates

5. If you are planning on using channel authentication:
   a. Decide the criteria for mapping inbound sessions to MCA user IDs
   b. Enable reverse DNS lookup by setting the queue manager parameter **REVDNS**
   c. Review security. For example, delete the default channels, and specify user IDs with only the necessary authority in the **MCAUSER** attribute for a channel.
6. Capture the accounting and statistics SMF records produced by the channel initiator and post process them.
7. Automate the monitoring of job log messages.
8. If necessary, tune your network environment to improve throughput. With TCP, large send and receive buffers improve throughput. You can force MQ to use specific TCP buffer sizes using the commands:

```
RECOVER QMGR(TUNE CHINTCPRBDYNSZ nnnnn)
RECOVER QMGR(TUNE CHINTCPSBDYNSZ nnnnn)
```

   which sets the SO_RCVBUF, and SO_SNDBUF, for the channels to the size in bytes specified in nnnnn.

Related concepts

“Planning for your queue manager” on page 146
When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

z/OS Planning for channel initiator SMF data

You need to plan the implementation of collecting SMF data for the channel initiator.

The channel initiator produces two types of record:

- Statistics data with information about the channel initiator and the tasks within it.
- Channel accounting data with information similar to the DISPLAY CHSTATUS command.

You start collecting statistics data using the command:

```
START TRACE(STAT) CLASS(4)
```

and stop it using the command:

```
STOP TRACE(STAT) CLASS(4)
```

You start collecting accounting data using the command:

```
START TRACE(ACCTG) CLASS(4)
```

and stop it using the command:

```
STOP TRACE(ACCTG) CLASS(4)
```

You can control which channels have accounting data collected for using the **STATCHL** attribute on the channel definition or the queue manager.

- For client channels, you must set **STATCHL** at the queue manager level.
- For automatically defined cluster sender channels, you can control the collection of accounting data with the **STATACLS** queue manager attribute.

The default value of **STATCHL** for the queue manager is OFF. In order to collect channel accounting data you must change the value of **STATCHL** from the default on either the queue manager or channel definition, in addition to starting class 4 accounting trace.

The SMF records are produced when:

- From IBM MQ for z/OS 9.3.0 onwards, the time interval indicated by the CSQ6SYSP **STATIME** or **ACCTIME** parameters has elapsed; or, if **STATIME** or **ACCTIME** is zero on the SMF data collection broadcast. The requests to collect SMF data for the channel initiator and the queue manager are synchronized.
- A `STOP TRACE(ACCTG) CLASS(4)` or `STOP TRACE(STAT) CLASS(4)` command is issued, or
- The channel initiator is shut down. At this point any SMF data is written out.

If a channel stops during the SMF interval, accounting data is written to SMF the next time the SMF processing runs. If a client connects, does some work and disconnects, then reconnects and disconnects, there are two sets of channel accounting data produced.

The statistics data normally fits into one SMF record, however, multiple SMF records might be created if a large number of tasks are in use.

Accounting data is gathered for each channel for which it is enabled, and normally fits into one SMF record. However, multiple SMF records might be created if a large number of channels are active.

The cost of collecting the channel initiator SMF data is small. Typically the increase in CPU usage is under a few percent, and often within measurement error.

Before you use this function you need to work with your z/OS systems programmer to ensure that SMF has the capacity for the additional records, and that they change their processes for extracting SMF records to include the new SMF data.

For channel initiator statistics data, the SMF record type is 115 and sub-type 231.

For channel initiator accounting data, the SMF record type is 116 and sub-type 10.

You can write your own programs to process this data, or use the SupportPac MP1B that contains a program, MQSMF, for printing the data, and creating data in Comma Separated Values (CSV) format suitable for importing into a spread sheet.

If you are experiencing issues with capturing channel initiator SMF data, see Dealing with issues when capturing SMF data for the channel initiator (CHINIT) for further information.

Related tasks

Interpreting IBM MQ performance statistics

Troubleshooting channel accounting data

z/OS Planning your z/OS TCP/IP environment

To get the best throughput through your network, you must use TCP/IP send and receive buffers with a size of 64 KB, or greater. With this size, the system optimizes its buffer sizes.

See What is Dynamic Right Sizing for High Latency Networks? for more information.

You can check your system buffer size by using the following Netstat command, for example:

```
TSO NETSTAT ALL (CLIENT csq1CHIN
```

The results display much information, including the following two values:

```
ReceiveBufferSize: 0000065536
SendBufferSize: 0000065536
```

65536 is 64 KB. If your buffer sizes are less than 65536, you must work with your network team to increase the **TCPSENDBFRSIZE** and **TCPRCVBUFRSIZE** values in the PROFILE DDName in the TCPIP procedure. For example, you might use the following command:

```
TCPCONFIG TCPSENDBFRSZE 65536 TCPRCVBUFRSIZE 65536
```

If you are unable to change your system-wide **TCPSENDBFRSIZE** or **TCPRCVBUFRSIZE** settings, contact your IBM Software Support center.

z/OS Planning your queue sharing group (QSG)

The easiest way to implement a shared queuing environment, is to configure a queue manager, add that queue manager to a QSG, then add other queue managers to the QSG.

A queue sharing group uses Db2 tables to store configuration information. There is one set of tables used by all QSGs that share the same Db2 data sharing group.

Shared queue messages are stored in a structure in a coupling facility (CF). Each QSG has its own set of CF structures. You need to configure the structures to meet your needs.

Messages over 63KB in size cannot be stored in the CF. You need to use either Shared Message Data Sets (SMDS) or Db2 for these messages.

Message profiles and capacity planning

You should understand the message profile of your shared queue messages. The following are examples of factors that you need to consider:

- Average, and maximum message size
- The typical queue depth, and exception queue depth. For example, you might need to have enough capacity to hold messages for a whole day, and the typical queue depth is under 100 messages.

If the message profile changes, you can increase the size of the structures, or implement SMDS, at a later date.

If you want to be able to handle a large peak volume of messages, you can configure IBM MQ to offload messages to SMDS when the usage of the structure reaches user specified thresholds.

You need to decide if you want to duplex the CF structures. This is controlled by the CF structure definition in the CFRM policy:

1. A duplexed structure uses two coupling facilities. If there is a problem with one CF, there is no interruption to the service, and the structure can be rebuilt on a third CF, if one is available. Duplexed structures can significantly impact the performance of operations on shared queues.
2. If the structure is not duplexed, then a problem with the CF means that shared queues on structures in that CF will become unavailable until the structure can be rebuilt in another CF.

   IBM MQ can be configured to automatically rebuild structures in another CF in this case. Persistent messages will be recovered from the logs of the queue managers.

Note that it is easy to change the CF definitions.

You can define a structure so that it can hold nonpersistent messages only, or so that it can hold persistent and nonpersistent messages.

Structures that can hold persistent messages need to be backed up periodically. Back up your CF structures at least every hour to minimize the time needed to recover the structure in the event of a failure. The backup is stored in the log data set of the queue manager performing the backup.

If you are expecting to have a high throughput of messages on your shared queues, it is best practice to have a dedicated queue manager for backing up the CF structures. This reduces the time needed to recover the structures, as a less data needs to be read from queue manager logs.

Channels

To provide a single system image for applications connecting into an IBM MQ QSG, you can define shared input channels. If these are set up, then a connection coming into the queue sharing group environment, can go to any queue manager in the QSG.

You might need to set up a network router, or Virtual IP address (VIPA) for these channels.

You can define shared output channels. A shared output channel instance can be started from any queue manager in the QSG.

See Shared channels for more information.

Security

You protect IBM MQ resources using an external security manager. If you are using RACF®, the RACF profiles are prefixed with the queue manager name. For example, a queue named APPLICATION.INPUT would be protected using a profile in the MQQUEUE class named `qmgrName.APPLICATION.INPUT.`

When using a queue sharing group you can continue to protect resources with profiles prefixed with the queue manager name, or you can prefix profiles with the queue sharing group name. For example `qsgName.APPLICATION.INPUT`.

You should aim to use profiles prefix with the queue sharing group name because this means there is a single definition for all queue managers, saving you work, and preventing a mismatch in definitions between queue managers.

Related concepts

“Planning for your queue manager” on page 146
When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

z/OS Planning your coupling facility and offload storage environment

Use this topic when planning the initial sizes, and formats of your coupling facility (CF) structures, and shared message data set (SMDS) environment or Db2 environment.

This section contains information about the following topics:

Defining coupling facility resources

If you intend to use shared queues, you must define the coupling facility structures that IBM MQ will use in your CFRM policy. To do this you must first update your CFRM policy with information about the structures, and then activate the policy.

Your installation probably has an existing CFRM policy that describes the coupling facilities available. The Administrative data utility is used to modify the contents of the policy based on textual statements you provide. You must add statements to the policy that defines the names of the new structures, the coupling facilities that they are defined in, and what size the structures are.

The CFRM policy also determines whether IBM MQ structures are duplexed and how they are reallocated in failure scenarios. Shared queue recovery contains recommendations for configuring CFRM for resilience to failures that affect the coupling facility.

Deciding your offload storage environment

The message data for shared queues can be offloaded from the coupling facility and stored in either a Db2 table or in an IBM MQ managed data set called a *shared message data set* (SMDS). Messages which are too large to store in the coupling facility (that is, larger than 63 KB) must always be offloaded, and smaller messages can optionally be offloaded to reduce coupling facility space usage.

For more information, see Specifying offload options for shared messages.

Planning your structures

A queue sharing group (QSG) requires a minimum of two structures to be defined. The first structure, known as the administrative structure, is used to coordinate IBM MQ internal activity across the queue sharing group. No user data is held in this structure. It has a fixed name of *qsg-name*CSQ_ADMIN (where *qsg-name* is the name of your queue sharing group). Subsequent structures are known as application structures, and are used to hold the messages on IBM MQ shared queues. Each structure can hold up to 512 shared queues.

An application structure named *qsg-name*CSQSYSAPPL is used for system queues. Defining this structure is optional, but it is required in order to use certain features. By default, the SYSTEM.QSG.CHANNEL.SYNCQ and SYSTEM.QSG.UR.RESOLUTION.QUEUE queues are defined on the *qsg-name*CSQSYAPPL structure.

Using multiple structures

A queue sharing group can connect to up to 64 coupling facility structures. One of these structures must be the administration structure. If it is defined, another of these structures might be the *qsg-name*CSQSYSAPPL structure. You can use up to 63 (62 if *qsg-name*CSQSYSAPPL is defined) structures for message data. You might choose to use multiple application structures for any of the following reasons:

- You have some queues that are likely to hold a large number of messages and so require all the resources of an entire coupling facility.
- You have a requirement for a large number of shared queues, so they must be split across multiple structures because each structure can contain only 512 queues.
- RMF reports on the usage characteristic of a structure suggest that you should distribute the queues it contains across a number of coupling facilities.
- You want some queue data to held in a physically different coupling facility from other queue data for data isolation reasons.
- Recovery of persistent shared messages is performed using structure level attributes and commands, for example BACKUP CFSTRUCT. To simplify backup and recovery, you could assign queues that hold nonpersistent messages to different structures from those structures that hold persistent messages.

When choosing which coupling facilities to allocate the structures in, consider the following points:

- Your data isolation requirements.
- The volatility of the coupling facility (that is, its ability to preserve data through a power outage).
- Failure independence between the accessing systems and the coupling facility, or between coupling facilities.
- The level of coupling facility control code (CFCC) installed on the coupling facility (IBM MQ requires Level 9 or higher).

Planning the size of your structures

The administrative structure

The administrative structure (*qsg-name*CSQ_ADMIN) must be large enough to contain 1000 list entries for each queue manager in the queue sharing group. When a queue manager starts, the structure is checked to see if it is large enough for the number of queue managers currently *defined* to the queue sharing group. Queue managers are considered as being defined to the queue sharing group if they have been added by the CSQ5PQSG utility. You can check which queue managers are defined to the group with the MQSC DISPLAY GROUP command.

Note: When calculating the size of the structure, you should allow for the size of large units of work, in addition to the number of queue managers in the queue sharing group.

Table 22 on page 182 shows the minimum required size for the administrative structure for various numbers of queue managers defined in the queue sharing group. These sizes were established for a CFCC level 14 coupling facility structure; for higher levels of CFCC, they probably need to be larger.

Table 22. Minimum administrative structure sizes

Number of queue managers defined in queue sharing group	Required storage
1	6144 KB
2	6912 KB
3	7976 KB
4	8704 KB
5	9728 KB
6	10496 KB
7	11520 KB
8	12288 KB
9	13056 KB
10	14080 KB
11	14848 KB
12	15616 KB
13	16640 KB
14	17408 KB
15	18176 KB
16	19200 KB
17	19968 KB
18	20736 KB
19	21760 KB
20	22528 KB
21	23296 KB
22	24320 KB
23	25088 KB

Table 22. Minimum administrative structure sizes (continued)	
Number of queue managers defined in queue sharing group	**Required storage**
24	25856 KB
25	27136 KB
26	27904 KB
27	28672 KB
28	29696 KB
29	30464 KB
30	31232 KB
31	32256 KB

When you add a queue manager to an existing queue sharing group, the storage requirement might have increased beyond the size recommended in Table 22 on page 182. If so, use the following procedure to estimate the required storage for the *qsg-name*CSQ_ADMIN structure:

1. Issue MQSC command **DISPLAY CFSTATUS(CSQ_ADMIN)** on an existing member of the queue sharing group.
2. Extract the ENTSMAX information for the CSQ_ADMIN structure.
3. If this number is less than 1000 times the total number of queue managers you want to define in the queue sharing group, increase the structure size.

Application structures

The size of the application structures required to hold IBM MQ messages depends on the likely number and size of the messages to be held on a structure concurrently.

The graph in Figure 42 on page 184 shows how large you should make your CF structures to hold the messages on your shared queues. To calculate the allocation size you need the following information:

- The average size of messages on your queues.
- The total number of messages likely to be stored in the structure.

Find the number of messages along the horizontal axis. Select the curve that corresponds to your message size and determine the required value from the vertical axis. For example, for 200 000 messages of length 1 KB gives a value in the range 256 through 512 MB.

Table 23 on page 184 provides the same information in tabular form.

Figure 42. Calculating the size of a coupling facility structure

Use this table to help calculate how large to make your coupling facility structures:

Table 23. Calculating the size of a coupling facility structure

Number of messages	1 KB	2 KB	4 KB	8 KB	16 KB	32 KB	63 KB
100	6 MB	6 MB	7 MB	7 MB	8 MB	10 MB	14 MB
1000	8 MB	9 MB	12 MB	17 MB	27 MB	48 MB	88 MB
10000	25 MB	38 MB	64 MB	115 MB	218 MB	423 MB	821 MB
100000	199 MB	327 MB	584 MB	1097 MB	2124 MB	4177 MB	8156 MB

Your CFRM policy should include the following statements:

- `INITSIZE` is the size in KB that the structure is allocated with when the first queue manager connects to it.
- `SIZE` is the maximum size that the structure can attain.
- `FULLTHRESHOLD` sets the percentage value of the threshold at which z/OS issues message IXC585E to indicate that the structure is getting full.

A best practice is to ensure that INITSIZE and SIZE are within a factor of 2. For example, with the figures determined previously, you might include the following statements:

```
STRUCTURE NAME(structure-name)
INITSIZE(value from graph in KB, that is, multiplied by 1024)
SIZE(something larger)
FULLTHRESHOLD(85)
```

```
STRUCTURE NAME(QSG1APPLICATION1)
INITSIZE(262144) /* 256 MB */
SIZE(524288)     /* 512 MB */
FULLTHRESHOLD(85)
```

If the structure use reaches the threshold where warning messages are issued, intervention is required. You might use IBM MQ to inhibit MQPUT operations to some of the queues in the structure to prevent applications from writing more messages, start more applications to get messages from the queues, or quiesce some of the applications that are putting messages to the queue.

Alternatively, you can use z/OS facilities to alter the structure size in place. The following z/OS command:

```
SETXCF START,ALTER,STRNAME=structure-name,SIZE=newsize
```

alters the size of the structure to *newsize*, where *newsize* is a value that is less than the value of SIZE specified on the CFRM policy for the structure, but greater than the current coupling facility size.

You can monitor the use of a coupling facility structure with the MQSC DISPLAY CFSTATUS command.

If no action is taken and a queue structure fills up, an MQRC_STORAGE_MEDIUM_FULL return code is returned to the application. If the administration structure becomes full, the exact symptoms depend on which processes experience the error, but they might include the following problems:

- No responses to commands.
- Queue manager failure as a result of problems during commit processing.

The CSQSYSAPPL structure

The *qsg-name*CSQSYSAPPL structure is an application structure for system queues. Table 3 demonstrates an example of how to estimate the message data sizes for the default queues defined on the *qsg-name*CSQSYSAPPL structure.

Table 24. Table showing CSQSYSAPPL usage against sizing.

***qsg-name*CSQSYSAPPL usage**	**Sizing**
SYSTEM.QSG.CHANNEL.SYNCQ	2 messages of 500 bytes per active instance of a shared channel
SYSTEM.QSG.UR.RESOLUTION.QUEUE	1000 messages of 2 KB

The suggested initial structure definition values are as follows:

```
STRUCTURE NAME(qsg-nameCSQSYSAPPL)
INITSIZE(20480)        /* 20 MB */
SIZE(30720)            /* 30 MB */
FULLTHRESHOLD(85)
```

These values can be adjusted depending on your use of shared channels and group units of recovery.

Mapping shared queues to structures

To define an application structure to IBM MQ, use the DEFINE CFSTRUCT command. When you define a structure to IBM MQ, do not include the QSG name prefix in the structure name. For example, to define an application structure to IBM MQ that has the name *qsg-name*APPLICATION1 in the CFRM policy, issue the following command:

```
DEFINE CFSTRUCT(APPLICATION1)
```

The CFSTRUCT attribute of the queue definition is used to map the queue to a structure. Specify the name of the CF structure without the QSG name prefix in this attribute. For example, the following command defines a shared queue on the APPLICATION1 structure:

```
DEFINE QLOCAL(myqueue) QSGDISP(SHARED) CFSTRUCT(APPLICATION1)
```

z/OS *Planning your shared message data set (SMDS) environment*

If you are using queue sharing groups with SMDS offloading, IBM MQ needs to connect to a group of shared message data sets. Use this topic to help understand the data set requirements, and configuration required to store IBM MQ message data.

A *shared message data set* (described by the keyword SMDS) is a data set used by a queue manager to store offloaded message data for shared messages stored in a coupling facility structure.

Note: When defining SMDS data sets for a structure, you must have one for each queue manager.

When this form of data offloading is enabled, the **CFSTRUCT** requires an associated group of shared message data sets, one data set for each queue manager in the queue sharing group. The group of shared message data sets is defined to IBM MQ using the **DSGROUP** parameter on the **CFSTRUCT** definition. Additional parameters can be used to supply further optional information, such as the number of buffers to use and expansion attributes for the data sets.

Each queue manager can write to the data set which it owns, to store shared message data for messages written through that queue manager, and can read all of the data sets in the group.

A list describing the status and attributes for each data set associated with the structure is maintained internally as part of the **CFSTRUCT** definition, so each queue manager can check the definition to find out which data sets are currently available.

This data set information can be displayed using the **DISPLAY CFSTATUS TYPE(SMDS)** command to display current status and availability, and the **DISPLAY SMDS** command to display the parameter settings for the data sets associated with a specified **CFSTRUCT**.

Individual shared message data sets are effectively identified by the combination of the owning queue manager name (usually specified using the **SMDS** keyword) and the **CFSTRUCT** structure name.

This section describes the following topics:

- The DSGROUP parameter
- The DSBLOCK parameter
- Shared message data set characteristics
- Shared message data set space management
- Access to shared message data sets
- Creating a shared message data set
- Shared message data set performance and capacity considerations
- Activating a shared message data set

See DEFINE CFSTRUCT for details of these parameters.

For information on managing your shared message data sets, see Managing shared message data sets for further details.

The DSGROUP parameter

The **DSGROUP** parameter on the **CFSTRUCT** definition identifies the group of data sets in which large messages for that structure are to be stored. Additional parameters may be used to specify the logical block size to be used for space allocation purposes and values for the buffer pool size and automatic data set expansion options.

The **DSGROUP** parameter must be set up before offloading to data sets can be enabled.

- If a new **CFSTRUCT** is being defined at **CFLEVEL(5)** and the option **OFFLOAD(SMDS)** is specified or assumed, then the **DSGROUP** parameter must be specified on the same command.
- If an existing **CFSTRUCT** is being altered to increase the **CFLEVEL** to **CFLEVEL(5)** and the option **OFFLOAD(SMDS)** is specified or assumed, then the **DSGROUP** parameter must be specified on the same command if it is not already set.

The DSBLOCK parameter

Space within each data set is allocated to queues as logical blocks of a fixed size (usually 256 KB) specified using the **DSBLOCK** parameter on the **CFSTRUCT** definition, then allocated to individual messages as ranges of pages of 4 KB (corresponding to the physical block size and control interval size) within each logical block. The logical block size also determines the maximum amount of message data that can be read or written in a single I/O operation, which is the same as the buffer size for the SMDS buffer pool.

A larger value of the **DSBLOCK** parameter can improve performance for very large messages by reducing the number of separate I/O operations. However, a smaller value decreases the amount of buffer storage required for each active request. The default value for the **DSBLOCK** parameter is 256 KB, which provides a reasonable balance between these requirements, so specifying this parameter might not normally be necessary.

Shared message data set characteristics

A shared message data set is defined as a VSAM linear data set (LDS). Each offloaded message is stored in one or more blocks in the data set. The stored data is addressed directly by information in the coupling facility entries, like an extended form of virtual storage. There is no separate index or similar control information stored in the data set itself.

The direct addressing scheme means that for messages which fit into one block, only a single I/O operation is needed to read or write the block. When a message spans more than one block, the I/O operations for each block can be fully overlapped to minimize elapsed time, provided that sufficient buffers are available.

The shared message data set also contains a small amount of general control information, consisting of a header in the first page, which includes recovery and restart status information, and a space map checkpoint area which is used to save the free block space map at queue manager normal termination.

Shared message data set space management

As background information for capacity, performance and operational considerations, it might be useful to understand the concepts of how space in shared message data sets is managed by the queue managers.

Free space in each shared message data set is tracked by its owning queue manager using a space map which indicates the number of pages in use within each logical block. The space map is maintained in main storage while the data set is open and saved in the data set when it is closed normally. (In recovery situations the space map is automatically rebuilt by scanning the messages in the coupling facility structure to find out which data set pages are currently in use).

When a shared message with offloaded message data is being written, the queue manager allocates a range of pages for each message block. If there is a partly used current logical block for the specified queue, the queue manager allocates space starting at the next free page in that block, otherwise it allocates a new logical block. If the whole message does not fit within the current logical block, the queue

manager splits the message data at the end of the logical block and allocates a new logical block for the next message block. This is repeated until space has been allocated for the whole message. Any unused space in the last logical block is saved as the new current logical block for the queue. When the data set is closed normally, any unused pages in current logical blocks are returned to the space map before it is saved.

When a shared message with offloaded message data has been read and is ready to be deleted, the queue manager processes the delete request by transferring the coupling facility entry for the message to a clean-up list monitored by the owning queue manager (which may be the same queue manager). When entries arrive on this list, the owning queue manager reads and deletes the entries and returns the freed ranges of pages to the space map. When all used pages in a logical block have been freed the block becomes available for reuse.

Access to shared message data sets

Each shared message data set must be on shared direct access storage which is accessible to all queue managers in the queue sharing group.

During normal running, each queue manager opens its own shared message data set for read/write access, and opens any active shared message data sets for other queue managers for read-only access, so it can read messages stored by those queue managers. This means that each queue manager userid requires at least UPDATE access to its own shared message data set and READ access to all other shared message data sets for the structure.

If it is necessary to recover shared message data sets using **RECOVER CFSTRUCT**, the recovery process can be executed from any queue manager in the queue sharing group. A queue manager which may be used to perform recovery processing requires UPDATE access to all data sets that it may need to recover

Creating a shared message data set

Each shared message data set should normally be created before the corresponding **CFSTRUCT** definition is created or altered to enable the use of this form of message offloading, as the **CFSTRUCT** definition changes will normally take effect immediately, and the data set will be required as soon as a queue manager attempts to access a shared queue which has been assigned to that structure. A sample job to allocate and pre-format a shared message data set is provided in SCSQPROC(CSQ4SMDS). The job must be customized and run to allocate a shared message data set for each queue manager which uses a CFSTRUCT with OFFLOAD(SMDS).

If the queue manager finds that offload support has been enabled and tries to open its shared message data set but it has not yet been created, the shared message data set will be flagged as unavailable. The queue manager will then be unable to store any large messages until the data set has been created and the queue manager has been notified to try again, for example using the **START SMDSCONN** command.

A shared message data set is created as a VSAM linear data set using an Access Method Services **DEFINE CLUSTER** command. The definition must specify **SHAREOPTIONS(2 3)** to allow one queue manager to open it for write access and any number of queue managers to read it at the same time. The default control interval size of 4 KB must be used. If the data set may need to expand beyond 4 GB, it must be defined using an SMS data class which has the VSAM extended addressability attribute. A shared message data set is eligible to reside in the extended addressing space (EAS) part of an extended address volumes (EAV).

Each shared message data set can either be empty or pre-formatted to binary zeros (using **CSQJUFMT** or a similar utility such as the sample job SCSQPROC(CSQ4SMDS)), before its initial use. If it is empty or only partly formatted when it is opened, the queue manager automatically formats the remaining space to binary zeros.

Shared message data set performance and capacity considerations

Each shared message data set is used to store offloaded data for shared messages written to the associated **CFSTRUCT** by the owning queue manager, from regions within the same system. Each message that is offloaded takes up to 768 bytes of CF storage, made up of 256 bytes for the entry

and 512 bytes for the two elements of header and descriptor. Each offloaded message is stored in one or more pages (physical blocks of size 4 KB) in the data set.

The data set space required for a given number of offloaded messages can therefore be estimated by rounding up the overall message size (including the descriptor) to the next multiple of 4 KB and then multiplying by the number of messages.

As for a page set, when a shared message data set is almost full, it can optionally be expanded automatically. The default behavior for this automatic expansion can be set using the **DSEXPAND** parameter on the **CFSTRUCT** definition. This setting can be overridden for each queue manager using the **DSEXPAND** parameter on the **ALTER SMDS** command. Automatic expansion is triggered when the data set reaches 90% full and more space is required. If expansion is allowed but an expansion attempt is rejected by VSAM because no secondary space allocation was specified when the data set was defined, expansion is retried using a secondary allocation of 20% of the current size of the data set.

Provided that the shared message data set is defined with the extended addressability attribute, the maximum size is only limited by VSAM considerations to a maximum of 16 TB or 59 volumes. This is significantly larger than the 64 GB maximum size of a local page set.

Activating a shared message data set

When a queue manager has successfully connected to an application coupling facility structure, it checks whether that structure definition specifies offloading using an associated **DSGROUP** parameter. If so, the queue manager allocates and opens its own shared message data set for write access, then it opens for read access any existing shared message data sets owned by other queue managers.

When a shared message data set is opened for the first time (before it has been recorded as active within the queue sharing group), the first page will not yet contain a valid header. The queue manager fills in header information to identify the queue sharing group, the structure name and the owning queue manager.

After the header has been completed, the queue manager registers the new shared message data set as active and broadcasts an event to notify any other active queue managers about the new data set.

Every time a queue manager opens a shared message data set it validates the header information to ensure that the correct data set is still being used and that it has not been damaged.

z/OS *Planning your Db2 environment*

If you are using queue sharing groups, IBM MQ needs to attach to a Db2 subsystem that is a member of a data sharing group. Use this topic to help understand the Db2 requirements used to hold IBM MQ data.

IBM MQ needs to know the name of the data sharing group that it is to connect to, and the name of a Db2 subsystem (or Db2 group) to connect to, to reach this data sharing group. These names are specified in the QSGDATA parameter of the CSQ6SYSP system parameter macro (described in Using CSQ6SYSP).

Within the data sharing group, shared Db2 tables are used to hold:

- Configuration information for the queue sharing group.
- Properties of IBM MQ shared and group objects.
- Optionally, data relating to offloaded IBM MQ messages.

IBM MQ provides a single set of sample jobs for defining the necessary Db2 table spaces, tables, and indexes. These jobs make use of Universal Table Spaces (UTS). Earlier versions of the product had two sets of jobs, one for UTS, and one for older types of table space, which have been deprecated by the most recent versions of Db2.

IBM MQ can still be used with older types of table space, and this might be appropriate if you already have an existing queue sharing group. However, if you are creating a new queue sharing group, it should use UTS.

Db2 V12 Function level 508 provides a non disruptive migration process for migrating multi-table table spaces to universal table spaces. You can use this approach to migrate the multi-table table spaces, used

by existing queue sharing groups, to universal table spaces without taking an outage of the whole queue sharing group.

In Db2 V13, use the MOVE TABLE option of the ALTER TABLESPACE statement. See Moving tables from multi-table table spaces to partition-by-growth table spaces for more information.

By default Db2 uses the user ID of the person running the jobs as the owner of the Db2 resources. If this user ID is deleted then the resources associated with it are deleted, and so the table is deleted. Consider using a group ID to own the tables, rather than an individual user ID. You can do this by adding `GROUP=groupname` onto the JOB card, and specifying `SET CURRENT SQLID='groupname'` before any SQL statements.

IBM MQ uses the RRS Attach facility of Db2. This means that you can specify the name of a Db2 group that you want to connect to. The advantage of connecting to a Db2 group attach name (rather than a specific Db2 subsystem), is that IBM MQ can connect (or reconnect) to any available Db2 subsystem on the z/OS image that is a member of that group. There must be a Db2 subsystem that is a member of the data sharing group active on each z/OS image where you are going to run a queue-sharing IBM MQ subsystem, and RRS must be active.

Db2 storage

For most installations, the amount of Db2 storage required is about 20 or 30 cylinders on a 3390 device. However, if you want to calculate your storage requirement, the following table gives some information to help you determine how much storage Db2 requires for the IBM MQ data. The table describes the length of each Db2 row, and when each row is added to or deleted from the relevant Db2 table. Use this information together with the information about calculating the space requirements for the Db2 tables and their indexes in the *Db2 for z/OS Installation Guide*.

Table 25. Planning your Db2 storage requirements

Db2 table name	Length of row	A row is added when:	A row is deleted when:
CSQ.ADMIN_B_QSG	252 bytes	A queue sharing group is added to the table with the ADD QSG function of the CSQ5PQSG utility.	A queue sharing group is removed from the table with the REMOVE QSG function of the CSQ5PQSG utility. (All rows relating to this queue sharing group are deleted automatically from all the other Db2 tables when the queue sharing group record is deleted.)
CSQ.ADMIN_B_QMGR	Up to 3828 bytes	A queue manager is added to the table with the ADD QMGR function of the CSQ5PQSG utility.	A queue manager is removed from the table with the REMOVE QMGR function of the CSQ5PQSG utility.
CSQ.ADMIN_B_STRUCTURE	1454 bytes	The first local queue definition, specifying the QSGDISP(SHARED) attribute, that names a previously unknown structure within the queue sharing group is defined.	The last local queue definition, specifying the QSGDISP(SHARED) attribute, that names a structure within the queue sharing group is deleted.
CSQ.ADMIN_B_SCST	342 bytes	A shared channel is started.	A shared channel becomes inactive.
CSQ.ADMIN_B_SSKT	254 bytes	A shared channel that has the NPMSPEED(NORMAL) attribute is started.	A shared channel that has the NPMSPEED(NORMAL) attribute becomes inactive.

Table 25. Planning your Db2 storage requirements (continued)

Db2 table name	Length of row	A row is added when:	A row is deleted when:
CSQ.ADMIN_B_STRBACKUP	514 bytes	A new row is added to the CSQ.ADMIN_B_STRUCTURE table. Each entry is a dummy entry until the BACKUP CFSTRUCT command is run, which overwrites the dummy entries.	A row is deleted from the CSQ.ADMIN_B_STRUCTURE table.
CSQ.OBJ_B_AUTHINFO	3400 bytes	An authentication information object with QSGDISP(GROUP) is defined.	An authentication information object with QSGDISP(GROUP) is deleted.
CSQ.OBJ_B_QUEUE	Up to 3707 bytes	• A queue with the QSGDISP(GROUP) attribute is defined. • A queue with the QSGDISP(SHARED) attribute is defined. • A model queue with the DEFTYPE(SHAREDYN) attribute is opened.	• A queue with the QSGDISP(GROUP) attribute is deleted. • A queue with the QSGDISP(SHARED) attribute is deleted. • A dynamic queue with the DEFTYPE(SHAREDYN) attribute is closed with the DELETE option.
CSQ.OBJ_B_NAMELIST	Up to 15127 bytes	A namelist with the QSGDISP(GROUP) attribute is defined.	A namelist with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_CHANNEL	Up to 14127 bytes	A channel with the QSGDISP(GROUP) attribute is defined.	A channel with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_STGCLASS	Up to 2865 bytes	A storage class with the QSGDISP(GROUP) attribute is defined.	A storage class with the QSGDISP(GROUP) attribute class is deleted.
CSQ.OBJ_B_PROCESS	Up to 3347 bytes	A process with the QSGDISP(GROUP) attribute is defined.	A process with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_TOPIC	Up to 14520 bytes	A topic object with QSGDISP(GROUP) attribute is defined.	A topic object with QSGDISP(GROUP) attribute is deleted.
CSQ.EXTEND_B_QMGR	Less than 430 bytes	A queue manager is added to the table with the ADD QMGR function of the CSQ5PQSG utility.	A queue manager is removed from the table with the REMOVE QMGR function of the CSQ5PQSG utility.
CSQ.ADMIN_B_MESSAGES	87 bytes	For large message PUT (1 per BLOB).	For large message GET (1 per BLOB).

Table 25. Planning your Db2 storage requirements (continued)			
Db2 table name	**Length of row**	**A row is added when:**	**A row is deleted when:**
CSQ.ADMIN_MSGS_BAUX1 CSQ.ADMIN_MSGS_BAUX2 CSQ.ADMIN_MSGS_BAUX3 CSQ.ADMIN_MSGS_BAUX4		These 4 tables contain message payload for large messages added into one of these 4 tables for each BLOB of the message. BLOBS are up to 511 KB in length, so if the message size is > 711 KB, there will be multiple BLOBs for this message.	

The use of large numbers of shared queue messages of size greater than 63 KB can have significant performance implications on your IBM MQ system. For more information, see SupportPac MP16, Capacity Planning and Tuning for IBM MQ for z/OS, at: SupportPacs for IBM MQ and other project areas.

z/OS Planning for backup and recovery

Developing backup and recovery procedures at your site is vital to avoid costly and time-consuming losses of data. IBM MQ provides means for recovering both queues and messages to their current state after a system failure.

This topic contains the following sections:

Recovery procedures

Develop the following procedures for IBM MQ:

- Creating a point of recovery.
- Backing up page sets.
- Backing up CF structures.
- Recovering page sets.
- Recovering from out-of-space conditions (IBM MQ logs and page sets).
- Recovering CF structures.

See Amministrazione di IBM MQ for z/OS for information about these.

Become familiar with the procedures used at your site for the following:

- Recovering from a hardware or power failure.
- Recovering from a z/OS component failure.

- Recovering from a site interruption, using off-site recovery.

z/OS Tips for backup and recovery

Use this topic to understand some backup and recovery tasks.

The queue manager restart process recovers your data to a consistent state by applying log information to the page sets. If your page sets are damaged or unavailable, you can resolve the problem using your backup copies of your page sets (if all the logs are available). If your log data sets are damaged or unavailable, it might not be possible to recover completely.

Consider the following points:

- Periodically take backup copies
- Do not discard archive logs you might need
- Do not change the DDname to page set association

Periodically take backup copies

A *point of recovery* is the term used to describe a set of backup copies of IBM MQ page sets and the corresponding log data sets required to recover these page sets. These backup copies provide a potential restart point in the event of page set loss (for example, page set I/O error). If you restart the queue manager using these backup copies, the data in IBM MQ is consistent up to the point that these copies were taken. Provided that all logs are available from this point, IBM MQ can be recovered to the point of failure.

The more recent your backup copies, the quicker IBM MQ can recover the data in the page sets. The recovery of the page sets is dependent on all the necessary log data sets being available.

In planning for recovery, you need to determine how often to take backup copies and how many complete backup cycles to keep. These values tell you how long you must keep your log data sets and backup copies of page sets for IBM MQ recovery.

When deciding how often to take backup copies, consider the time needed to recover a page set. The time needed is determined by the following:

- The amount of log to traverse.
- The time it takes an operator to mount and remove archive tape volumes.
- The time it takes to read the part of the log needed for recovery.
- The time needed to reprocess changed pages.
- The storage medium used for the backup copies.
- The method used to make and restore backup copies.

In general, the more frequently you make backup copies, the less time recovery takes, but the more time is spent making copies.

For each queue manager, you should take backup copies of the following:

- The archive log data sets
- The BSDS copies created at the time of the archive
- The page sets
- Your object definitions
- Your CF structures

To reduce the risk of your backup copies being lost or damaged, consider:

- Storing the backup copies on different storage volumes to the original copies.
- Storing the backup copies at a different site to the original copies.

- Making at least two copies of each backup of your page sets and, if you are using single logging or a single BSDS, two copies of your archive logs and BSDS. If you are using dual logging or BSDS, make a single copy of both archive logs or BSDS.

Before moving IBM MQ to a production environment, fully test and document your backup procedures.

Backing up your page sets

You need to back up page sets regularly. Some enterprises back up the page sets twice a day.

You need the active and archive logs since a backup to be able to recover using the backup. You need enough log data to go back four checkpoints if the backup was taken when the queue manager was running.

You can use ADRDSSU FastReplication to back up page sets, and you can do this while the queue manager is active. Note that you need to ensure there is enough space in the storage pool.

Backing up your object definitions

Create backup copies of your object definitions. To do this, use the MAKEDEF feature of the COMMAND function of the utility program (described in Using the COMMAND function of CSQUTIL).

You should do this whenever you take backup copies of your queue manager data sets, and keep the most current version.

Backing up your coupling facility structures

If you have set up any queue sharing groups, even if you are not using them, you must take periodic backups of your CF structures. To do this, use the IBM MQ BACKUP CFSTRUCT command. You can use this command only on CF structures that are defined with the RECOVER(YES) attribute. If any CF entries for persistent shared messages refer to offloaded message data stored in a shared message data set (SMDS) or Db2, the offloaded data is retrieved and backed up together with the CF entries. Shared message data sets should not be backed up separately.

It is recommended that you take a backup of all your CF structures about every hour, to minimize the time it takes to restore a CF structure.

You could perform all your CF structure backups on a single queue manager, which has the advantage of limiting the increase in log use to a single queue manager. Alternatively, you could perform backups on all the queue managers in the queue sharing group, which has the advantage of spreading the workload across the queue sharing group. Whichever strategy you use, IBM MQ can locate the backup and perform a RECOVER CFSTRUCT from any queue manager in the queue sharing group. The logs of all the queue managers in the queue sharing group need to be accessed to recover the CF structure.

Backing up your message security policies

If you are using Advanced Message Security to create a backup of your message security policies, create a backup using the message security policy utility (CSQ0UTIL) to run **dspmqspl** with the -export parameter, then save the policy definitions that are output to the EXPORT DD.

You should create a backup of your message security policies whenever you take backup copies of your queue manager data sets, and keep the most current version.

Do not discard archive logs you might need

IBM MQ might need to use archive logs during restart. You must keep sufficient archive logs so that the system can be fully restored. IBM MQ might use an archive log to recover a page set from a restored backup copy. If you have discarded that archive log, IBM MQ cannot restore the page set to its current state. When and how you discard archive logs is described in Discarding archive log data sets.

You can use the `/cpf DIS USAGE TYPE(ALL)` command to display the log RBA, and log range sequence number (LRSN) that you need to recover your queue manager's page sets and the queue sharing group's structures. You should then use the print log map utility (CSQJU004) to print bootstrap data set (BSDS) information for the queue manager to locate the logs containing the log RBA.

For CF structures, you need to run the CSQJU004 utility on each queue manager in the queue sharing group to locate the logs containing the LRSN. You need these logs and any later logs to be able to recover the page sets and structures.

Do not change the DDname to page set association

IBM MQ associates page set number 00 with DDname CSQP0000, page set number 01 with DDname CSQP0001, and so on, up to CSQP0099. IBM MQ writes recovery log records for a page set based on the DDname that the page set is associated with. For this reason, you must not move page sets that have already been associated with a PSID DDname.

z/OS Recovering page sets

Use this topic to understand the factors involved when recovering pages sets, and how to minimize restart times.

A key factor in recovery strategy concerns the time for which you can tolerate a queue manager outage. The total outage time might include the time taken to recover a page set from a backup, or to restart the queue manager after an abnormal termination. Factors affecting restart time include how frequently you back up your page sets, and how much data is written to the log between checkpoints.

To minimize the restart time after an abnormal termination, keep units of work short so that, at most, two active logs are used when the system restarts. For example, if you are designing an IBM MQ application, avoid placing an MQGET call that has a long wait interval between the first in-syncpoint MQI call and the commit point because this might result in a unit of work that has a long duration. Another common cause of long units of work is batch intervals of more than 5 minutes for the channel initiator.

You can use the DISPLAY THREAD command to display the RBA of units of work and to help resolve the old ones.

How often must you back up a page set?

Frequent page set backup is essential if a reasonably short recovery time is required. This applies even when a page set is very small or there is a small amount of activity on queues in that page set.

If you use persistent messages in a page set, the backup frequency should be in hours rather than days. This is also the case for page set zero.

To calculate an approximate backup frequency, start by determining the target total recovery time. This consists of the following:

1. The time taken to react to the problem.
2. The time taken to restore the page set backup copy.

   If you use SnapShot backup/restore, the time taken to perform this task is a few seconds. For information about SnapShot, see the *DFSMSdss Storage Administration Guide*.
3. The time the queue manager requires to restart, including the additional time needed to recover the page set.

   This depends most significantly on the amount of log data that must be read from active and archive logs since that page set was last backed up. All such log data must be read, in addition to that directly associated with the damaged page set.

   Note: When using *fuzzy backup* (where a snapshot is taken of the logs and page sets while a unit of work is active), it might be necessary to read up to three additional checkpoints, and this might result in the need to read one or more additional logs.

When deciding on how long to allow for the recovery of the page set, the factors that you need to consider are:

- The rate at which data is written to the active logs during normal processing depends on how messages arrive in your system, in addition to the message rate.

  Messages received or sent over a channel result in more data logging than messages generated and retrieved locally.
- The rate at which data can be read from the archive and active logs.

  When reading the logs, the achievable data rate depends on the devices used and the total load on your particular DASD subsystem.

  With most tape units, it is possible to achieve higher data rates for archived logs with a large block size. However, if an archive log is required for recovery, all the data on the active logs must be read also.

z/OS Recovering CF structures

Use this topic to understand the recovery process for CF structures.

At least one queue manager in the queue sharing group must be active to process a RECOVER CFSTRUCT command. CF structure recovery does not affect queue manager restart time, because recovery is performed by an already active queue manager.

The recovery process consists of two logical steps that are managed by the RECOVER CFSTRUCT command:

1. Locating and restoring the backup.
2. Merging all the logged updates to persistent messages that are held on the CF structure from the logs of all the queue managers in the queue sharing group that have used the CF structure, and applying the changes to the backup.

The second step is likely to take much longer because a lot of log data might need to be read. You can reduce the time taken if you take frequent backups, or if you recover multiple CF structures at the same time, or both.

The queue manager performing the recovery locates the relevant backups on all the other queue managers' logs using the data in Db2 and the bootstrap data sets. The queue manager replays these backups in the correct time sequence across the queue sharing group, from just before the last backup through to the point of failure.

The time it takes to recover a CF structure depends on the amount of recovery log data that must be replayed, which in turn depends on the frequency of the backups. In the worst case, it takes as long to read a queue manager's log as it did to write it. So if, for example, you have a queue sharing group containing six queue managers, an hour's worth of log activity could take six hours to replay. In general it takes less time than this, because reading can be done in bulk, and because the different queue manager's logs can be read in parallel. As a starting point, we recommend that you back up your CF structures every hour.

All queue managers can continue working with non-shared queues and queues in other CF structures while there is a failed CF structure. If the administration structure has also failed, at least one of the queue managers in the queue sharing group must be started before you can issue the RECOVER CFSTRUCT command.

Backing up CF structures can require considerable log writing capacity, and can therefore impose a large load on the queue manager doing the backup. Choose a lightly loaded queue manager for doing backups; for busy systems, add an additional queue manager to the queue sharing group and dedicate it exclusively for doing backups.

z/OS Achieving specific recovery targets

Use this topic for guidance on how you can achieve specific recovery target times by adjusting backup frequency.

If you have specific recovery targets to achieve, for example, completion of the queue manager recovery and restart processing in addition to the normal startup time within *xx* seconds, you can use the following calculation to estimate your backup frequency (in hours):

```
Formula (A)
                    Required restart time * System recovery log read rate
                          (in secs)                  (in MB/sec)
Backup frequency  = ------------------------------------------------------
   (in hours)              Application log write rate (in MB/hour)
```

Note: The examples given next are intended to highlight the need to back up your page sets frequently. The calculations assume that most log activity is derived from a large number of persistent messages. However, there are situations where the amount of log activity is not easily calculated. For example, in a queue sharing group environment, a unit of work in which shared queues are updated in addition to other resources might result in UOW records being written to the IBM MQ log. For this reason, the Application log write rate in Formula (A) can be derived accurately only from the observed rate at which the IBM MQ logs fill.

For example, consider a system in which IBM MQ MQI clients generate a total load of 100 persistent messages a second. In this case, all messages are generated locally.

If each message is of user length 1 KB, the amount of data logged each hour is approximately:

```
100 * (1 + 1.3) KB * 3600 = approximately 800 MB

where
      100           = the message rate a second
      (1 + 1.3) KB  = the amount of data logged for
                      each 1 KB of persistent messages
```

Consider an overall target recovery time of 75 minutes. If you have allowed 15 minutes to react to the problem and restore the page set backup copy, queue manager recovery and restart must then complete within 60 minutes (3600 seconds) applying formula (A). Assuming that all required log data is on RVA2-T82 DASD, which has a recovery rate of approximately 2.7 MB a second, this necessitates a page set backup frequency of at least every:

```
3600 seconds * 2.7 MB a second / 800 MB an hour = 12.15 hours
```

If your IBM MQ application day lasts approximately 12 hours, one backup each day is appropriate. However, if the application day lasts 24 hours, two backups each day is more appropriate.

Another example might be a production system in which all the messages are for request-reply applications (that is, a persistent message is received on a receiver channel and a persistent reply message is generated and sent down a sender channel).

In this example, the achieved batch size is one, and so there is one batch for every message. If there are 50 request replies a second, the total load is 100 persistent messages a second. If each message is 1 KB in length, the amount of data logged each hour is approximately:

```
50((2 * (1+1.3) KB) + 1.4 KB + 2.5 KB) * 3600 = approximately 1500 MB

where:
 50                 = the message pair rate a second
 (2 * (1 + 1.3) KB) = the amount of data logged for each message pair
 1.4 KB             = the overhead for each batch of messages
                      received by each channel
 2.5 KB             = the overhead for each batch of messages sent
                      by each channel
```

To achieve the queue manager recovery and restart within 30 minutes (1800 seconds), again assuming that all required log data is on RVA2-T82 DASD, this requires that page set backup is carried out at least every:

```
1800 seconds * 2.7 MB a second / 1500 MB an hour = 3.24 hours
```

Periodic review of backup frequency

Monitor your IBM MQ log usage in terms of MB an hour. Periodically perform this check and amend your page set backup frequency if necessary.

z/OS Backup considerations for other products

If you are using IBM MQ with CICS or IMS then you must also consider the implications for your backup strategy with those products. The data facility hierarchical storage manager (DFHSM) manages data storage, and can interact with the storage used by IBM MQ.

Backup and recovery with DFHSM

The data facility hierarchical storage manager (DFHSM) does automatic space-availability and data-availability management among storage devices in your system. If you use it, you need to know that it moves data to and from the IBM MQ storage automatically.

DFHSM manages your DASD space efficiently by moving data sets that have not been used recently to alternative storage. It also makes your data available for recovery by automatically copying new or changed data sets to tape or DASD backup volumes. It can delete data sets, or move them to another device. Its operations occur daily, at a specified time, and allow for keeping a data set for a predetermined period before deleting or moving it.

You can also perform all DFHSM operations manually. For more information on DFHSM, see the z/OS DFSMS product documentation. If you use DFHSM with IBM MQ, note that DFHSM does the following:

- Uses cataloged data sets.
- Operates on page sets and logs.
- Supports VSAM data sets.

Recovery and CICS

The recovery of CICS resources is not affected by the presence of IBM MQ. CICS recognizes IBM MQ as a non-CICS resource (or external resource manager), and includes IBM MQ as a participant in any syncpoint coordination requests using the CICS resource manager interface (RMI). For more information about CICS recovery and the CICS resource manager interface, see the CICS product documentation.

Recovery and IMS

IMS recognizes IBM MQ as an external subsystem and as a participant in syncpoint coordination. IMS recovery for external subsystem resources is described in the IMS product documentation.

z/OS Preparing for recovery on an alternative site

If a total loss of an IBM MQ computing center, you can recover on another IBM MQ system at a recovery site.

To recover an IBM MQ system at a recovery site, you must regularly back up the page sets and the logs. As with all data recovery operations, the objectives of disaster recovery are to lose as little data, workload processing (updates), and time as possible.

At the recovery site:

- The recovery IBM MQ queue manager **must** have the same name as the lost queue manager.
- Ensure the system parameter module used on the recovery queue manager contains the same parameters as the lost queue manager.

See Administering IBM MQ for z/OS and Troubleshooting IBM MQ for z/OS problems for more information.

z/OS Example of queue manager backup activity

This topic shows as an example of queue manager backup activity.

When you plan your queue manager backup strategy, a key consideration is retention of the correct amount of log data. Managing the logs describes how to determine which log data sets are required, by reference to the system recovery RBA of the queue manager. IBM MQ determines the system recovery RBA using information about the following:

- Currently active units of work.
- Page set updates that have not yet been flushed from the buffer pools to disk.
- CF structure backups, and whether this queue manager's log contains information required in any recovery operation using them.

You must retain sufficient log data to be able to perform media recovery. While the system recovery RBA increases over time, the amount of log data that must be retained only decreases when subsequent backups are taken. CF structure backups are managed by IBM MQ, and so are taken into account when reporting the system recovery RBA. This means that in practice, the amount of log data that must be retained only reduces when page set backups are taken.

Figure 43 on page 200 shows an example of the backup activity on a queue manager that is a member of a queue sharing group, how the recovery RBA varies with each backup, and how that affects the amount of log data that must be retained. In the example the queue manager uses local and shared resources: page sets, and two CF structures, STRUCTURE1 and STRUCTURE2.

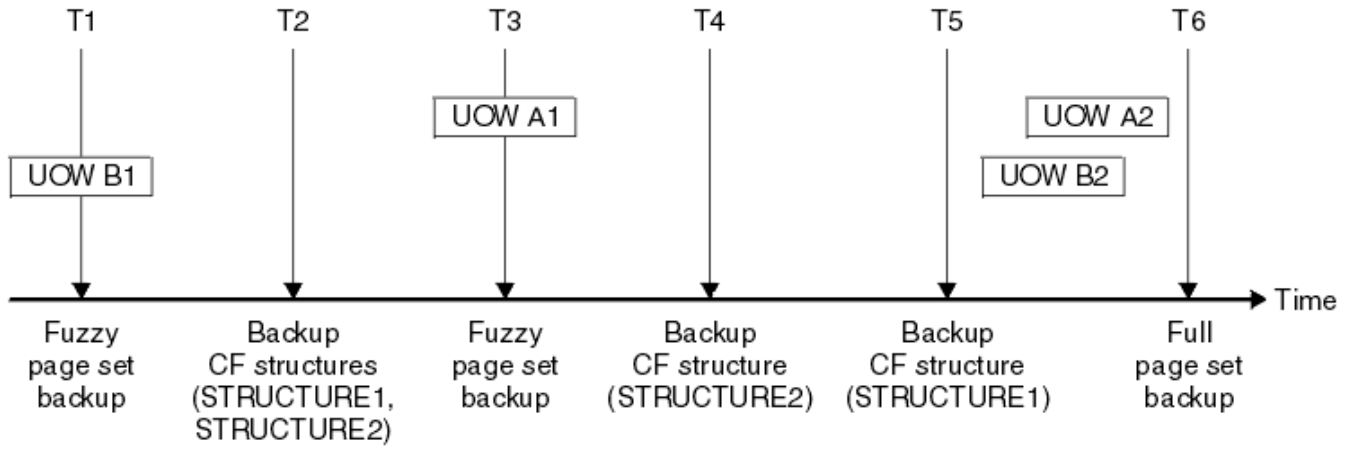


Figure 43. Example of queue manager backup activity

This is what happens at each point in time:

Point in time T1
A fuzzy backup is created of your page sets, as described in How to back up and recover page sets.

The system recovery RBA of the queue manager is the lowest of the following:

- The recovery RBAs of the page sets being backed up at this point.
- The lowest recovery RBA required to recover the CF application structures. This relates to the recovery of backups of STRUCTURE1 and STRUCTURE2 created earlier.
- The recovery RBA for the oldest currently active unit of work within the queue manager (UOWB1).

The system recovery RBA for this point in time is given by messages issued by the DISPLAY USAGE command, which is part of the fuzzy backup process.

Point in time T2
Backups of the CF structures are created. CF structure STRUCTURE1 is backed up first, followed by STRUCTURE2.

The amount of log data that must be retained is unchanged, because the same data as determined from the system recovery RBA at T1 is still required to recover using the page set backups taken at T1.

Point in time T3
Another fuzzy backup is created.

The system recovery RBA of the queue manager is the lowest of the following:

- The recovery RBAs of the page sets being backed up at this point.
- The lowest recovery RBA required to recover CF structure STRUCTURE1, because STRUCTURE1 was backed up before STRUCTURE2.
- The recovery RBA for the oldest currently active unit of work within the queue manager (UOWA1).

The system recovery RBA for this point in time is given by messages issued by the DISPLAY USAGE command, which is part of the fuzzy backup process.

You can now reduce the log data retained, as determined by this new system recovery RBA.

Point in time T4
A backup is taken of CF structure STRUCTURE2. The recovery RBA for the recovery of the oldest required CF structure backup relates to the backup of CF structure STRUCTURE1, which was backed up at time T2.

The creation of this CF structure backup has no effect on the amount of log data that must be retained.

Point in time T5

A backup is taken of CF structure STRUCTURE1. The recovery RBA for recovery of the oldest required CF structure backup now relates to recovery of CF structure STRUCTURE2, which was backed up at time T4.

The creation of this CF structure backup has no effect on amount of log data that must be retained.

Point in time T6

A full backup is taken of your page sets as described in How to back up and recover page sets.

The system recovery RBA of the queue manager is the lowest of the following:

- The recovery RBAs of the page sets being backed up at this point.
- The lowest recovery RBA required to recover the CF structures. This relates to recovery of CF structure STRUCTURE2.
- The recovery RBA for the oldest currently active unit of work within the queue manager. In this case, there are no current units of work.

The system recovery RBA for this point in time is given by messages issued by the DISPLAY USAGE command, which is part of the full backup process.

Again, the log data retained can be reduced, because the system recovery RBA associated with the full backup is more recent.

z/OS Planning your z/OS UNIX environment

Certain processes within the IBM MQ queue manager, channel initiator, and mqweb server use z/OS UNIX System Services (z/OS UNIX) for their normal processing.

The queue manager and channel initiator started task user IDs need an OMVS segment with a UID defined in order to be able to access z/OS UNIX. The user IDs require no special permissions in z/OS UNIX.

Note: Although the queue manager and channel initiator make use of z/OS UNIX facilities (for example, to interface with TCP/IP services), they do not need to access any of the content of the IBM MQ installation directory in the z/OS UNIX file system. As a result, the queue manager and channel initiator do not require any configuration to specify the path for the z/OS UNIX file system.

The mqweb server, which hosts the IBM MQ Console and REST API, makes use of files in the IBM MQ installation directory in the z/OS UNIX file system. It also needs access to another file system which is used to store data such as configuration and log files. The mqweb started task JCL needs to be customized to reference these z/OS UNIX file systems.

The content of the IBM MQ directory in the z/OS UNIX file system is also used by applications connecting to IBM MQ. For example, applications using the IBM MQ classes for Java or IBM MQ classes for JMS interfaces.

See the following topics for the relevant configuration instructions:

- Environment variables relevant to IBM MQ classes for Java
- IBM MQ classes for Java libraries
- Setting environment variables
- Configuring the Java Native Interface (JNI) libraries

z/OS Planning for Advanced Message Security

TLS (or SSL) can be used to encrypt and protect messages flowing on a network, but this does not protect messages when they are on a queue ("at rest"). Advanced Message Security (AMS) protects the messages from the time that they are first put to a queue, until they are got, so that only the intended recipients of the message can read that message. The messages are encrypted and signed during put processing, and unprotected during get processing.

AMS can be configured to protect messages in different ways:

1. A message can be signed. The message is in clear text, but there is a checksum, which is signed. This allows any changes in the message content to be detected. From the signed content, you can identify who signed the data.
2. A message can be encrypted. The contents are not visible to anyone without the decryption key. The decryption key is encrypted for each recipient.
3. A message can be encrypted and signed. They decryption key is encrypted for each recipient, and from the signing you can identify who sent the message.

The encryption and signing use digital certificates and key rings.

You can set up a client to use AMS, so the data is protected before the data is put on the client channel. Protected messages can be sent to a remote queue manager, and you need to configure the remote queue manager to process these messages.

Setting up AMS

An AMS address space is used for doing the AMS work. This has additional security set up, to give access to and protect the use of key rings and certificates.

You configure which queues are to be protected by using a utility program (CSQ0UTIL) to define the security policies for queues.

Once AMS is set up

You need to set up a digital certificate and a key ring for people who put messages, and the people who get messages.

If a user, Alice, on z/OS needs to send a message to Bob, AMS needs a copy of the public certificate for Bob.

If Bob wants to process a message from Alice, AMS needs the public certificate for Alice, or the same certificate authority certificate used by Alice.

Attention: You need to:

- Carefully plan who can put to, or get from, queues
- Identify the people and their certificate names.

It is easy to make mistakes, and problems can be hard to resolve.

Related concepts

“Planning for your queue manager” on page 146
When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

z/OS Planning for Managed File Transfer

Use this section as guidance on how you need to set up your system to run Managed File Transfer (MFT) on z/OS.

z/OS Planning for Managed File Transfer - hardware and software requirements

Use this topic as guidance on how you need to set up hardware and software requirements on your system to run Managed File Transfer (MFT) on z/OS.

Software requirements

Managed File Transfer is written in Java, with some shell scripts and JCL to configure and operate the program.

Important: You must be familiar with z/OS UNIX System Services (z/OS UNIX) in order to configure Managed File Transfer. For example:

- The file directory structure, with names such as `/u/userID/myfile.txt`
- z/OS UNIX commands, for example:
  - `cd` (change directory)
  - `ls` (list)
  - `chmod` (change the file permissions)
  - `chown` (change file ownership or groups which can access the file or directory)

You require the following products in z/OS UNIX to be able to configure and run MFT:

1. Java, for example, in directory `/java/java80_bit64_GA/J8.0_64/`
2. IBM MQ 9.4.0, for example, in directory `/mqm/V9R3M0`
3. If you want to use Db2 for status and history, you need to install Db2 JDBC libraries, for example, in directory `/db2/db2v10/jdbc/libs`.

Product registration

At startup Managed File Transfer checks the registration in `sys1.parmlib`(IFAPRDxx) concatenation. The following code is an example of how you register MFT:

```
PRODUCT OWNER('IBM CORP')
NAME('WS MQ FILE TRANS')
ID(5655-MFT)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('WS MQ FILE TRANS')
STATE(ENABLED)
```

Disk space

The IBM MQ for z/OS Program Directory states the DASD and zFS storage requirements for Managed File Transfer. For download links for the Program Directory for IBM MQ for z/OS, see IBM MQ 9.4 PDF files for product documentation and Program Directories.

z/OS Planning for Managed File Transfer - topologies

Use this topic as guidance on what topology you need on your system to run Managed File Transfer (MFT) on z/OS.

Managed File Transfer queue managers

IBM MQ Managed File Transfer topologies consist of:

Agents, and their associated queue managers
The agent uses system queues hosted on their agent queue manager to maintain state information and receive requests for work.

A command queue manager
This acts as a gateway into an MFT topology. It is connected to the agent queue managers through either sender and receiver channels, or clustering. When certain commands are run, they connect directly to the command queue manager, and send a message to the specified agent. This message is routed through the IBM MQ network to the agent queue manager, where it is picked up by the agent and processed.

A coordination queue manager
This is a central hub that has knowledge of the entire topology. The coordination queue manager is connected to all of the agent queue managers in a topology through either sender and receiver

channels, or using clustering. Agents regularly publish status information to the coordination queue manager, and store their transfer templates there.

It is possible for a single queue manager to perform multiple roles within a topology. For example, the same queue manager can be configured as both the coordination queue manager and the command queue manager for a topology.

If you are using multiple queue managers you need to set up channels between the queue managers. You can either do this by using clustering or by using point-to-point connections.

When using IBM MQ Managed File Transfer for z/OS, there are a number of things to consider when determining which queue managers to use for the different roles within a topology.

Agent queue managers

The agent queue manager for an IBM MQ Managed File Transfer for z/OS agent must be running on z/OS.

If:

- The agent is running Managed File Transfer for z/OS on IBM MQ 9.1 or later
- And, the agent queue manager is licensed for IBM MQ Advanced for z/OS Value Unit Edition (Advanced VUE)

the agent can connect to the queue manager using the CLIENT transport.

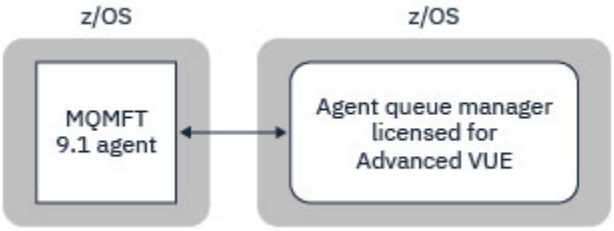


Figure 44. MFT 9.1 agents on z/OS can connect to a queue manager using the CLIENT transport, assuming the queue manager is licensed for Advanced VUE.

If:

- The agent is running Managed File Transfer for z/OS on IBM MQ 9.0 or earlier
- Or, the agent queue manager is running Managed File Transfer for z/OS on IBM MQ 9.0 or later, and the agent queue manager is licensed for either MFT, IBM MQ Advanced for z/OS, or Advanced VUE

the agent must connect to the queue manager using the BINDINGS transport.

Figure 45. MFT 9.0 agents on z/OS and 9.1 agents that have an agent queue manager licensed for either MFT or IBM MQ Advanced, must connect using the BINDINGS transport.

Command queue managers

The Which MFT commands and processes connect to which queue manager topic shows all of the commands that connect to the command queue manager for a Managed File Transfer topology.

Note: When running these commands on z/OS, the command queue manager must also be on z/OS.

If the command queue manager is licensed for Advanced VUE, the commands can connect to the queue manger using the CLIENT transport. Otherwise, the commands must connect to the command queue manager using the BINDINGS transport.

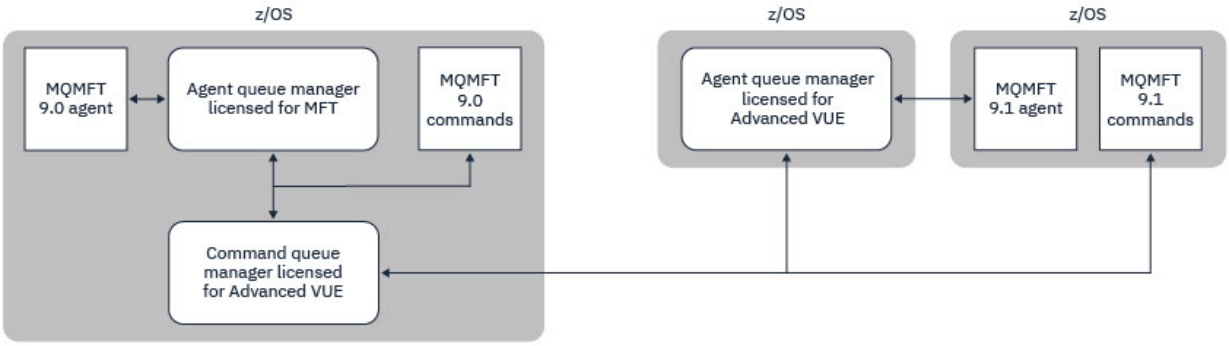


Figure 46. Commands connect to the command queue manager for an MFT topology. When running these commands on z/OS, the command queue manager must also be on z/OS

Coordination queue managers

IBM MQ Managed File Transfer for z/OS agents can be part of a topology where the coordination queue manager is either running on z/OS, or is running on a multiplatform.

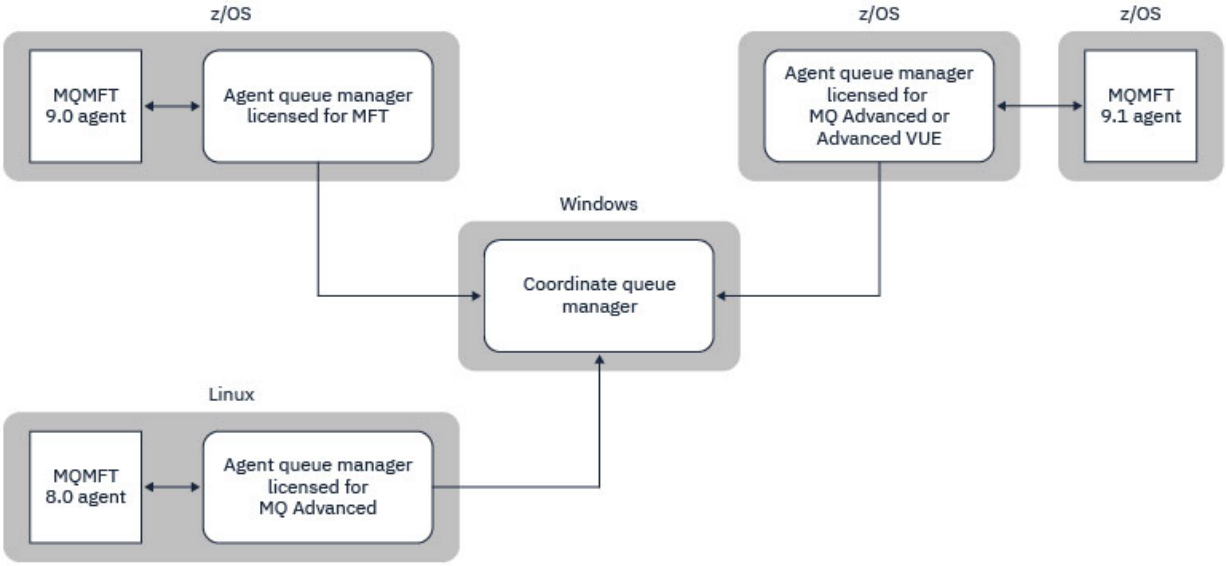


Figure 47. MFT agents running on z/OS can be part of an MFT topology where the coordination queue manager is running on an IBM MQ multiplatform.

The Which MFT commands and processes connect to which queue manager topic shows the commands that connect to the coordination queue manager for a Managed File Transfer topology. It is possible to run these commands on z/OS and have then connect to the coordination queue manager running on a different platform.

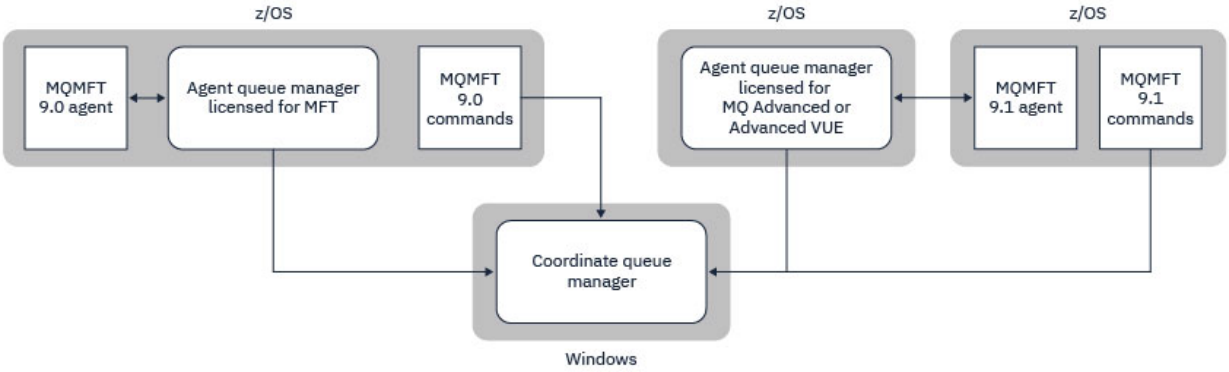


*Figure 48. Certain commands, such as **fteListAgents**, connect directly to the coordination queue manager for an MFT topology.*

How many agents do I need?

The agents do the work in transferring data, and when you make a request to transfer data you specify the name of an agent.

By default an agent can process 25 send and 25 receive requests concurrently. You can configure these processes. See Managed File Transfer configuration options on z/OS for more information.

If the agent is busy then work is queued. The time taken to process a request depends on multiple factors, for example, the amount of data to be sent, the network bandwidth, and the delay on the network.

You might want to have multiple agents to process work in parallel.

You can also control which resources an agent can access, so you might want some agents to work with a limited subset of data.

If you want to process requests with different priority you can use multiple agents and use workload manager to set the priority of the jobs.

Running the agents

Typically the agents are long running processes. The processes can be submitted as jobs that run in batch, or as started tasks.

z/OS Planning for Managed File Transfer - security considerations

Use this topic as guidance on what security considerations you need on your system to run Managed File Transfer (MFT) on z/OS.

Security

You need to identify which user IDs are going to be used for MFT configuration and for MFT operation.

You need to identify the files or queues you transfer, and which user IDs are going to be submitting transfer requests to MFT.

When you customize the agents and logger, you specify the group of users that is allowed to run MFT services, or do MFT administration.

You should set up this group before you start customizing MFT. As MFT uses IBM MQ queues, if you have security enabled in the queue manager, MFT requires access to the following resources:

Table 26. MQADMIN resource class

Name	Access required
QUEUE.SYSTEM.FTE.EVENT.agent_name	Update

Table 26. MQADMIN resource class (continued)	
Name	**Access required**
QUEUE.SYSTEM.FTE.COMMAND.agent_name	Update
CONTEXT.SYSTEM.FTE.COMMAND.agent_name	Update
QUEUE.SYSTEM.FTE.STATE.agent_name	Update
QUEUE.SYSTEM.FTE.DATA.agent_name	Update
QUEUE.SYSTEM.FTE.REPLY.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHAGT1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHTRN1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHOPS1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHSCH1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHMON1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHADM1.agent_name	Update

Table 27. MQQUEUE resource class	
Name	**Access required**
SYSTEM.FTE.AUTHAGT1.agent_name	Update
SYSTEM.FTE.AUTHTRN1.agent_name	Update
SYSTEM.FTE.AUTHOPS1.agent_name	Update
SYSTEM.FTE.AUTHSCH1.agent_name	Update
SYSTEM.FTE.AUTHMON1.agent_name	Update

You can use user sandboxing to determine which parts of the file system the user who requests the transfer can access.

To enable user sandboxing, add the `userSandboxes=true` statement to the *agent.properties* file for the agent that you want to restrict, and add appropriate values to the `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml` file.

See Working with user sandboxes for further information.

This user ID is configured in `UserSandboxes.xml` files.

This XML file has information like user ID, or user ID* and a list of resource that can be used (included), or cannot be used (excluded). You need to define specific user IDs that can access which resources: for example:

Table 28. Example user ID together with access to specific resources			
User ID	**Access**	**Include or Exclude**	**Resource**
Admin*	Read	Include	`/home/user/**`
Admin*	Read	Exclude	`/home/user/private/**`
Sysprog	Read	Include	`/home/user/**`
Admin*	Read	Include	Application.reply.queue

Notes:

1. If `type=queue` is specified, the resource is either a queue name, or `queue@qmgr`.
2. If the resource begins with `//`, the resource is a data set; otherwise the resource is a file in z/OS UNIX.
3. The user ID is the user ID from the MQMD structure, so this might not reflect the user ID that actually puts the message.
4. For requests on the local queue manager you can use MQADMIN CONTEXT.* to limit which users can set this value.
5. For requests coming in over a remote queue manager, you have to assume that the distributed queue managers have security enabled to prevent unauthorized setting of the user ID in the MQMD structure.
6. A user ID of SYSPROG1 on a Linux machine, is the same user ID SYSPROG1 for the security checking on z/OS.

z/OS Planning to use the IBM MQ Console and REST API on z/OS

The IBM MQ Console and REST API are applications that run in a WebSphere Liberty (Liberty) server known as mqweb. The mqweb server runs as a started task. The IBM MQ Console allows a web browser to be used to administer queue managers. The REST API provides a simple programmatic interface for applications to do queue manager administration, and to perform messaging.

Installation and configuration files

You need to install the IBM MQ for z/OS UNIX System Services Web Components feature, which will install the files needed to run the mqweb server in z/OS UNIX System Services (z/OS UNIX). You need to be familiar with z/OS UNIX to be able to configure and manage the mqweb server.

See IBM MQ for z/OS Program Directory PDF files for information on installing IBM MQ for z/OS UNIX System Services Components.

The IBM MQ files in z/OS UNIX are installed with various attributes set that are required for the correct operation of the mqweb server. If you need to copy the IBM MQ z/OS UNIX installation files, for example if you have installed IBM MQ on one system, and run IBM MQ on a different system, you should copy the IBM MQ ZFS created during the installation, and mount it read only at the destination. Copying the files in other ways might cause some file attributes to be lost.

You need to decide upon the location for, and create, a Liberty user directory when you create the mqweb server. This directory contains configuration and log files, and the location can be something similar to `/var/mqm/mqweb`.

Using the IBM MQ Console and REST API with queue managers at different levels

The REST API can directly interact only with queue managers that run at the same Version, Release, and Modification (VRM) as the mqweb server which runs the REST API. For example, the IBM MQ 9.4.0 REST API can directly interact only with local queue managers at IBM MQ 9.4.0, and the IBM MQ 9.3.5 REST API can directly interact only with local queue managers at IBM MQ 9.3.5.

You can use the REST API to administer a queue manager at a different version from the mqweb server by configuring a gateway queue manager. However, you need at least one queue manager at the same version as the mqweb server to act as the gateway queue manager. For more information, see Remote administration using the REST API.

The IBM MQ Console can be used to manage local queue managers that run at the same version as the IBM MQ Console. From IBM MQ 9.3.0, you can also use the IBM MQ Console to administer a queue manager running on a remote system, or at a different version to the IBM MQ Console. For more information, see IBM MQ Console: Adding a remote queue manager.

Migration

If you have only one queue manager, you can run the mqweb server as a single started task, and change the libraries it uses when you migrate your queue manager.

If you have more than one queue manager, during migration you can start mqweb servers at different versions by using started tasks with different names. These names can be any name you want. For example, you can start an IBM MQ 9.3.0 mqweb server using a started task named MQWB0930, and an IBM MQ 9.3.5 mqweb server using a started task named MQWB0935.

Then, when you migrate the queue managers from one version to a later version, the queue managers become available in the mqweb server for the later version, and are no longer available in the mqweb server for the earlier version.

After you have migrated all the queue managers to the later version, you can delete the mqweb server for the earlier version.

HTTP ports

The mqweb server uses up to two ports for HTTP:

- One for HTTPS, with a default value of 9443.
- One for HTTP. HTTP is not enabled by default, but if enabled, has a default value of 9080.

If the default port values are in use, you must allocate other ports. If you have more than one mqweb server running simultaneously for more than one version of IBM MQ, you must allocate separate ports for each version. For more information on setting the ports that the mqweb server uses, see Configuring the HTTP and HTTPS ports.

You can use the following TSO command to display information about a port:

```
NETSTAT TCP tcpip (PORT portNumber)
```

where *tcpip* is the name of the TCP/IP address space, and *portNumber* specifies the number of the port to display information about.

Security - starting the mqweb server

The mqweb server user ID needs certain authorities. For more information, see Authority required by the mqweb server started task user ID.

Security - using the IBM MQ Console and REST API

When you use the IBM MQ Console and REST API, you must authenticate as a user that is included in a configured registry. These users are assigned specific roles that determine the actions the users can perform. For example, to use the messaging REST API, a user must be assigned the `MQWebUser` role. For more information about the available roles for the IBM MQ Console and REST API, and the access that these roles grant, see Roles on the IBM MQ Console and REST API.

For more information about configuring security for the IBM MQ Console and REST API, see IBM MQ Console and REST API security.

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.

Tutti i riferimenti a siti Web non dell'IBM contenuti in questo documento sono forniti solo per consultazione e non rappresenta in alcun modo un'approvazione di tali siti. I materiali reperibili in tali siti Web non fanno parte dei materiali relativi a questo prodotto IBM e l'utilizzo di tali siti è responsabilità dell'utente.

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 IBM 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: