

9.3

IBM MQ ' nun Planlanması

IBM

Not

Bu bilgileri ve desteklediđi ürünü kullanmadan önce, [“Özel notlar” sayfa 205](#) bölümündeki bilgileri okuyun.

Bu basım, yeni basımlarda tersi belirtilmedikçe, IBM® MQ sürüm 9 yayın düzeyi 3 ve sonraki tüm yayınlar ve deđişiklikler için geçerlidir.

IBM'e bilgi gönderdiğinizde, IBM ' e bu bilgileri size hiçbir sorumluluk yüklemeyen uygun gördüğü yöntemlerle kullanması ya da dağıtması için münhasır olmayan bir hak verirsiniz.

© Copyright International Business Machines Corporation 2007, 2024.

İçindekiler

Planlama.....	5
IBM MQ yayın tipleri: planlama ile ilgili önemli noktalar.....	5
IBM MQ ve IBM MQ Appliance şirket içinde GDPR ' ye hazır olma ile ilgili önemli noktalar.....	8
Tek bir kuyruk yöneticisine dayalı mimariler.....	18
Birden çok kuyruk yöneticisine dayalı mimariler.....	19
Dağıtılmış kuyruklarınızın ve kümelerinizin planlanması.....	19
Dağıtılmış yayınlama/abone olma ağınızın planlanması.....	69
Çoklu platformlarda depolama ve performans gereksinimlerinizin planlanması.....	107
Çoklu platformlarda disk alanı gereksinimleri.....	108
Multiplatforms üzerinde dosya sistemi desteğinin planlanması.....	111
Çoklu Platformlar üzerinde MFT için planlama dosya sistemi desteği.....	138
Multiplatforms üzerinde dairesel ya da doğrusal günlük kaydının seçilmesi.....	139
AIX üzerinde paylaşılan bellek.....	139
IBM MQ ve UNIX System V IPC kaynakları.....	140
IBM MQ ve UNIX Süreç Önceliği.....	140
z/OS üzerinde IBM MQ ortamınızın planlanması.....	140
Kuyruk yöneticinizin planlanması.....	141
Kanal başlatıcınızın planlanması.....	168
Kuyruk paylaşım grubunuz (QSG) planlanması.....	173
Yedekleme ve kurtarma planlaması.....	186
z/OS UNIX ortamınızın planlanması.....	195
Advanced Message Security planlaması.....	195
Managed File Transfer planlaması.....	196
z/OS üzerinde IBM MQ Console ve REST API kullanılmasının planlanması.....	202
Özel notlar.....	205
Programlama arabirimi bilgileri.....	206
Ticari Markalar.....	206

IBM MQ mimarisinin planlanması


IBM MQ ortamınızı planlarken, IBM MQ ' un tek ve birden çok kuyruk yöneticisi mimarisi ve noktadan noktaya iletişim ve yayınlama/abone olma ileti sistemi stilleri için sağladığı desteği göz önünde bulundurun. Ayrıca, kaynak gereksinimlerinizi ve günlük kaydı ve yedekleme olanaklarını kullanımınızı da planlayın.

Bu görev hakkında

IBM MQ mimarinizi planlamadan önce, temel IBM MQ kavramlarını tanıyın. Bkz. [IBM MQ Teknik genel bakış](#).

IBM MQ mimarileri, tek bir kuyruk yöneticisi kullanan basit mimarilerden birbirine bağlı kuyruk yöneticilerinden oluşan daha karmaşık ağlara kadar değişir. Birden çok kuyruk yöneticisi, dağıtılmış kuyruğa alma teknikleri kullanılarak birbirine bağlanır. Tek kuyruk yöneticisini ve birden çok kuyruk yöneticisi mimarisini planlama hakkında daha fazla bilgi için aşağıdaki konulara bakın:

- [“Tek bir kuyruk yöneticisine dayalı mimariler” sayfa 18](#)
- [“Birden çok kuyruk yöneticisine dayalı mimariler” sayfa 19](#)
 - [“Dağıtılmış kuyruklarınızın ve kümelerinizin planlanması” sayfa 19](#)
 - [“Dağıtılmış yayınlama/abone olma ağınızın planlanması” sayfa 69](#)

 IBM MQ for z/OS üzerinde, iş yükü dengelemeyi uygulamanızı ve IBM MQ uygulamalarınızın ölçeklenebilir ve yüksek kullanılabilirlikli olmasını sağlamak için paylaşılan kuyrukları ve kuyruk paylaşım gruplarını kullanabilirsiniz. Paylaşılan kuyruklar ve kuyruk paylaşım gruplarına ilişkin bilgi için [Paylaşılan kuyruklar ve kuyruk paylaşım grupları](#) başlıklı konuya bakın.

IBM MQ iki farklı yayın modeli sağlar:

- Long Term Support (LTS) yayın düzeyi, uzun vadeli devreye alma ve maksimum kararlılık gerektiren sistemler için en uygundur.
- Continuous Delivery (CD) yayın düzeyi, IBM MQ için en son işlevsel geliştirmelerden hızla yararlanması gereken sistemler için tasarlanmıştır.

Her iki yayın tipi de aynı şekilde kurulur, ancak anlamanız gereken destek ve geçişle ilgili önemli noktalar vardır. Daha fazla bilgi için bkz. [IBM MQ yayın tipleri ve sürümü oluşturma](#).

Birden çok kuruluş planlaması, depolama ve performans gereksinimleri ve istemcilerin kullanımı hakkında bilgi için diğer alt konulara bakın.

İlgili kavramlar

IBM MQ yayın tipleri ve sürüm oluşturma

[“z/OS üzerinde IBM MQ ortamınızın planlanması” sayfa 140](#)

IBM MQ ortamınızı planlarken, veri kümeleri, sayfa kümeleri, Db2, Coupling Facilities ve günlüğe kaydetme ve yedekleme olanaklarına ilişkin kaynak gereksinimlerini göz önünde bulundurmanız gerekir. IBM MQ ' in çalıştığı ortamı planlamak için bu konuyu kullanın.

[Kullanılabilirlik, kurtarma ve yeniden başlatma](#)

İlgili görevler

[Gereksinimler denetleniyor](#)

[İletilerin kaybolmadığından emin olmak \(günlüğe kaydetme\)](#)

IBM MQ yayın tipleri: planlama ile ilgili önemli noktalar

IBM MQ için iki ana yayın tipi: Long Term Support (LTS) ve Continuous Delivery (CD). Desteklenen her platform için, seçtiğiniz yayın tipi sıralama, kuruluş, bakım ve geçiş işlemlerini etkiler.

Yayın tipleriyle ilgili ayrıntılı bilgi için bkz. [IBM MQ yayın tipleri ve sürüm oluşturma](#).

IBM MQ for Multiplatforms ile ilgili dikkat edilecek noktalar

Multi

Sıralama

Passport Advantage içinde IBM MQ 9.3 için iki ayrı eAssemblies vardır. Biri IBM MQ 9.3.0 Long Term Support yayın düzeyine ilişkin kuruluş görüntülerini, diğeri IBM MQ 9.3.x Continuous Delivery yayın düzeyine ilişkin kuruluş görüntülerini içerir. Yayın düzeyi seçiminize göre eAssembly ' den kuruluş görüntülerini yükleyin.

Tüm IBM MQ sürümleri ve IBM MQ 9.3 hem LTS yayınları hem de CD yayınları için aynı Ürün Tanıtıcısına aittir.

IBM MQ kullanım yetkisi, lisanslı bileşenlerin ve fiyatlandırma ölçülerinin kısıtlamalarına tabi olarak tüm ürün (PID) boyunca uzanır. Bu, IBM MQ 9.3 için LTS yayın düzeyi ve CD yayın düzeyi kuruluş görüntüleri arasında özgürce seçim yapabileceğiniz anlamına gelir.

Kuruluş

Passport Advantagesitesinden bir kuruluş görüntüsünü yükledikten sonra, yalnızca yetki satın aldığınız bileşenleri kurmanız gerekir. Her bir ücretlendirilebilir bileşen için hangi kurulabilir bileşenlerin dahil edilmesiyle ilgili daha fazla bilgi için bkz. [IBM MQ lisans bilgileri](#) .

IBM MQ 9.3.0 LTS yayın düzeyini ve IBM MQ 9.3.x CD yayın düzeyini aynı işletim sistemi görüntüsüne kurabilirsiniz. Bunu yaparsanız, bileşenler, IBM MQ çoklu sürüm desteği tarafından desteklenen ayrı kuruluşlar olarak görünür. Her bir sürümün, o sürümle ilişkilendirilmiş ayrı kuyruk yöneticisi kümeleri vardır.

Her yeni CD yayını, kuruluş görüntüsü olarak sağlanır. Yeni CD yayın düzeyi, var olan bir yayın düzeyinin yanı sıra kurulabilir ya da kuruluş programı tarafından yerine daha önceki bir CD yayın düzeyi yeni yayın düzeyine güncellenebilir.

CD yayınları, en son hata düzeltmeleri ve güvenlik güncellemeleri kümesinin yanı sıra işlevsel geliştirmeler içerir. Her CD yayın düzeyi kümülatiftir ve bu IBM MQsürümünün tüm önceki yayın düzeylerinin yerine geçer. Bu nedenle, kuruluşunuzla ilgili herhangi bir işlev içermiyorsa belirli bir CD yayını atlayabilirsiniz.

Bakım

LTS yayınına, hata düzeltmeleri sağlayan düzeltme paketlerinin uygulanması ve güvenlik yamaları sağlayan toplu güvenlik güncellemeleri (CSU) tarafından hizmet verilir. Düzeltme paketleri ve CSU ' lar belirli aralıklarla kullanıma sunulur ve kümülatiftir.

CD için, CSU ' lar yalnızca sonraki bir sürümde olabilecek en son CD yayın düzeyi için üretilir.

Ara düzeltme uygulamak için IBM Destek ekibi tarafından bazen yönlendirilebilirsiniz. Ara düzeltmeler, acil durum ya da test düzeltmeleri olarak da bilinir ve sonraki bakım teslimatı için bekleyemeyecek acil güncellemeleri uygulamak için kullanılır.

LTS yayın düzeyi ve CD yayın düzeyi arasında geçiş

Kısıtlamalar ve sınırlamalar vardır, ancak genellikle tek bir kuyruk yöneticisi LTS yayın kodunu CD yayın koduna ya da hedef yayın düzeyinin geçişten önce kullanılan daha yüksek olması koşuluyla, CD yayın kodunu LTS yayın koduna kullanmaktan geçirilebilir.

İki yaklaşım mümkündür:

- Var olan bir IBM MQ kuruluşunun güncellenmesi için yeni kod yayın düzeyini kurun. Kuruluşla ilişkili kuyruk yöneticileri, başlatıldığında kodun yeni yayın düzeyini kullanır.
- Yeni kod yayını yeni bir kuruluş olarak kurun ve [setmqm](#) komutunu kullanarak tek tek kuyruk yöneticisi yönetim ortamlarını yeni kuruluşa taşıyın.

Bir kuyruk yöneticisi kodun CD yayın düzeyini çalıştırmaya başladığında, kuyruk yöneticisi komut düzeyi yeni yayın düzeyini gösterecek şekilde güncellenir. Bu, yayın düzeyinde sağlanan yeni işlevlerin

etkinleştirildiği anlamına gelir ve daha düşük bir VRM numarasıyla bir kod yayını kullanarak kuyruk yöneticisini artık yeniden başlatamayamazsınız.

IBM MQ for z/OS ile ilgili dikkat edilecek noktalar

z/OS

Sıralama

IBM MQ for z/OS 9.3 siparişi verilirken, ShopZ' de iki ayrı aksam sunulur. Özellikler, LTS yayın düzeyine ve CD yayın düzeyine karşılık gelir. Her iki özellik de aynı ürün tanıtıcısı (PID) için geçerlidir. Bu, lisanslanan ürün tanıtıcısıdır, bu nedenle bir aksam lisanslandığında, gerekirse alternatif aksamı kullanma yetkisi vardır. Sipariş verirken, LTS yayın düzeyi ya da CD yayın düzeyine karşılık gelen özelliği seçin.

Ürünleri ServerPac' e eklemek üzere seçtiğinizde, ürünler SMP/E tarafından aynı hedef bölgeye kurulamayacağı için, aynı ServerPac siparişindeki hem LTS yayın düzeyini hem de CD yayın düzeyini seçemezsiniz.

Kuruluş

LTS ve CD yayınları ayrı FMID kümelerinde sağlanır. Bu FMID ' lerin aynı SMP/E hedef bölgesine kurulamayacağını unutmayın. Hem LTS hem de CD yayın düzeylerine gereksinim duyarsanız:

- LTS yayın düzeyini ve CD yayın düzeyini ayrı hedef bölgelere kurun.
- İki yayın düzeyi için ayrı hedef ve dağıtım kitaplıkları sağlayın.

Kuyruk yöneticiniz bir kuyruk paylaşım grubundaydıysa, en son CD sürümüne yükseldiğinizde, gruptaki tüm kuyruk yöneticilerini yükseltmeniz gerekir.

Kuyruk yöneticisinin komut düzeyi üç basamaklı VRM düzeyidir. IBM MQ programı, bağlı olduğu kuyruk yöneticisinin komut düzeyini almak için MQIA_COMMAND_LEVEL seçicisini geçirerek MQINQçağırabilir.

Yayınlar farklı FMID ' ler kullandığından, bir CD yayın düzeyini LTS yayın düzeyine ilişkin bir bakım ile güncelleyemezsiniz ya da diğer bir şekilde yuvarlayamazsınız. Benzer şekilde, ürün kodunun bir sürümünü LTS yayın düzeyinden CD yayın düzeyine ya da tersi yönde değiştirmenin bir yolu yoktur. Ancak, bir kuyruk yöneticisini serbest bırakma modelleri arasında geçiş yapabilirsiniz. Bkz. [Migration between LTS release and CD release](#).

Not:

IBM MQ 9.0.x ve IBM MQ 9.1.x CD yayınları, ayrı sürüm ve yayına bağımlı FMID ' lere sahipti. Bu nedenle, 9.0.x CD sürümünden 9.1.x CD sürümüne geçiş için en az bir tam SMP/E kuruluşu gerekir.

IBM MQ for z/OS 9.2.0 sürümünden CD yayın düzeyi, 9sürüm numarasına sahip tüm IBM MQ for z/OS yayınları için aynı kalan bir FMID kümesini kullanır. IBM MQ ürününün her yeni sürümü hem CD hem de LTS yayın düzeyi olarak kullanılabilirdiğinden, ana sürüm sınırını aşarken bile PTF ' leri tek bir SMP/E kuruluşuna uygulayarak CD yayın düzeylerini yükseltebilirsiniz. Örneğin, yalnızca PTF ' leri uygulayarak IBM MQ for z/OS 9.2.0 CD, IBM MQ for z/OS 9.2.2 CD, IBM MQ for z/OS 9.2.4 CD ve IBM MQ for z/OS 9.3.0 CD arasında gidebilirsiniz.

Kuyruk yöneticisi iş günlüğündeki [CSQY000I](#) iletisine bakarak aynı VRM düzeyine sahip bir LTS ve CD yayınını ayırt edebilirsiniz.

Bakım

IBM MQ for z/OS , Bakım için PTF ' leri kullanır.

LTS

PTF ' ler belirli bir yayın düzeyine karşılık gelen belirli bir kitaplık kümesine özgüdür. UNIX System Services özellikleri (JMS ve WEB UI, Connector Pack ve Managed File Transfer) için z/OS PTF 'ler doğrudan Multiplatforms düzeltme paketleriyle ve birikmeli güvenlik güncellemeleriyle (CSU' lar)hızalanır. Bu düzeltmeler kümülatiftir ve eşdeğer Multiplatforms düzeltme paketiyle ya da CSUaynı anda kullanılabilir.

CD

CD CSU ' lar genellikle CD yayınları arasında kullanılamaz, ancak sonraki IBM MQ for z/OS CD yayınında bulunur. + + USERMOD istemek için destek ile de iletişim kurabilirsiniz.

IBM MQ for z/OS üzerindeki diğer düzeltmeler, belirli bölümlere ilişkin ayrı düzeltmelerdir. Bu düzeltmeler, belirli sorunları çözebilir, kümülatif değildir ve bunlar üretildikleri zaman ve oluşturuldukları şekilde kullanıma sunulur.

LTS yayın düzeyi ve CD yayın düzeyi arasında geçiş

Kısıtlamalar ve sınırlamalar vardır, ancak hedef yayın düzeyinin geçişten önce kullanılanı daha yüksek olması koşuluyla, genellikle tek bir kuyruk yöneticisinin LTS yayın kodu kullanılarak CD yayın koduna ya da CD yayın kodu kullanılarak LTS yayın koduna geçişi yapılabilir.

IBM MQ for z/OS 9.2.0'den, gerektiği kadar VRM ile aynı VRM'ye sahip CD ve LTS yayınları arasında ve geriye doğru geçiş yeteneğini etkilemeden geçiş yapabilirsiniz. Örneğin, bir kuyruk yöneticisi IBM MQ for z/OS 9.3.0 LTS adresinde başlatılabilir, daha sonra IBM MQ for z/OS 9.3.0 CD adresinde kapatılıp başlatılabilir, daha sonra IBM MQ for z/OS 9.3.0 LTS adresinde kapatılıp başlatılabilir.

IBM MQ for z/OS geleneksel olarak bir geri dönüş yeteneği (geriye doğru geçiş) sağlamıştır; böylece, bir geçişin ardından bir süre çalıştıktan sonra önceki yayına geri dönebilirsiniz. Bu yetenek LTS yayınları ve 9.3.0 CD gibi 0 değiştiricisine sahip CD yayınları için korunur, ancak bir geçişin kaynağı ya da hedefi, sıfır olmayan bir değiştirici numarasına sahip bir CD yayını (örneğin, 9.2.5 ya da 9.3.1) olduğunda mümkün değildir.

Aşağıda geçerli geçiş senaryoları yer almaktadır ve bu prensibin nasıl çalıştığını göstermektedir:

V 9.3.0 V 9.3.0

Kaynak yayın düzeyi	Hedef yayın düzeyi	Notlar
9.0.0 LTS	9.3.0 LTS ya da 9.3.0 CD 'si	9.0.0 LTS standart desteğin dışında olduğundan geriye doğru geçiş desteklenmez.
9.1.0 LTS	9.3.0 LTS ya da 9.3.0 CD 'si	Geriye doğru geçiş desteklenir.
9.2.0 LTS	9.3.0 LTS ya da 9.3.0 CD 'si	Geriye doğru geçiş desteklenir.
9.2.5 CD 'si	9.3.0 LTS ya da 9.3.0 CD 'si	Kaynak yayın düzeyi, sıfır olmayan bir niteleyiciye sahip CD olduğundan geriye doğru geçiş desteklenmez.
9.3.0 LTS ya da 9.3.0 CD 'si	9.3.1 CD 'si	Hedef yayın düzeyi sıfır olmayan bir niteleyiciye sahip CD olduğundan, geriye doğru geçiş desteklenmez. Geçiş onaylamak için Write to operator with reply CSQY041D yayınlandı.

İlgili görevler

[z/OS üzerinde bakım uygulanması ve kaldırılması](#)

İlgili bilgiler

[karşıdan yükleme IBM MQ 9.3](#)

IBM MQ ve IBM MQ Appliance şirket içinde GDPR ' ye hazır olma ile ilgili önemli noktalar

PID ' ler için:

Dağıtık

- 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

Bildirim:

Bu belgenin amacı, GDPR için hazır olma çalışmalarınıza yardımcı olmaktır. GDPR'ye hazır olma konusunda kuruluşunuza yardımcı olmak için yapılandırabileceğiniz IBM MQ özellikleri hakkında bilgiler ve ürünün kullanımıyla ilgili dikkat etmeniz gereken noktalar sağlar. Müşterilerin özellikleri seçip yapılandırabileceği ve ürünün kendi içinde ya da üçüncü taraf uygulamaları ve sistemleri ile kullanılabileceği birçok yol olduğundan, bu bilgiler çok kapsamlı değildir.

Müşteriler, Avrupa Birliği Genel Veri Koruma Yönetmeliği de dahil olmak üzere çeşitli yasalara ve yasal düzenlemelere uyumluluklarını sağlamaktan kendileri sorumludur. Müşteriler kendi işlerini ve söz konusu yasalara ve yasal düzenlemelere uymak için yapmaları gereken işlemleri etkileyebilecek olan ilgili yasa ve yasal düzenlemelerin belirlenmesi ve yorumlanması ile ilgili olarak yetkili bir hukuk danışmanının önerisini almaktan tek başına sorumludur.

Ürünler, hizmetler ve burada açıklanan diğer yeterlilikler, tüm müşteri durumları için uygun olmayabilir ve bunların sağlanabilirliği kısıtlı olabilir. IBM , yasal, muhasebeyle ilgili ya da denetimle ilgili öneriler sağlamaz ya da hizmetlerinin ya da ürünlerinin, müşterilerin herhangi bir yasaya ya da düzenlemeye uymalarını sağlayacağını beyan ya da garanti etmez.

İçindekiler

1. [GDPR/GVKY](#)
2. [GDPR için Ürün Yapılandırması](#)
3. [Veri Yaşam Çevrimi](#)
4. [Veri Toplama](#)
5. [Veri Depolama](#)
6. [Veri Erişimi](#)
7. [Verilerin İşlenmesi](#)
8. [Verilerin Silinmesi](#)
9. [Verilerin İzlenmesi](#)
10. [Kişisel Verilerin Kullanımını kısıtlama yeteneği](#)
11. [Dosya işleme](#)

GDPR/GVKY

Genel Veri Koruma Yönetmeliği (GDPR; General Data Protection Regulation), Avrupa Birliği ve Avrupa Birliği ("AB") tarafından kabul edilmiş olup 25 Mayıs 2018 tarihinden itibaren yürürlüğe girmiştir.

GDPR neden önemli?

GDPR, bireylerin kişisel verilerinin işlenmesine yönelik daha güçlü bir yasal veri koruma çerçevesi oluşturur. GDPR aşağıdakileri sunar:

- Bireyler için yeni ve geliştirilmiş haklar
- Kişisel verilerin daha geniş bir tanımı
- Veri işleyenler için yeni yükümlülükler
- İhlal halinde ciddi finansal ceza uygulama potansiyeli
- Zorunlu veri ihlali bildirim

GDPR hakkında daha fazla bilgi edinin:

- [AB GDPR bilgi portalı](#)
- [ibm.com/GDPR web sitesi](http://ibm.com/GDPR)

Ürün Yapılandırması - GDPR'ye hazırlıkla ilgili önemli noktalar

Aşağıdaki bölümlerde, kuruluşunuza GDPR 'ye hazır olma konusunda yardımcı olması için IBM MQ ' in yapılandırılmasına ilişkin önemli noktalar sağlanır.

Veri Yaşam Çevrimi

IBM MQ , uygulamaların uygulama tarafından sağlanan verileri zamanuyumsuz olarak değiş tokuş etmesini sağlayan, işlemsel ileti odaklı bir ara katman yazılımı ürünüdür. IBM MQ , uygulamaların bağlanması amacıyla çeşitli ileti sistemi API ' lerini, iletişim kurallarını ve köprülerini destekler. Bu nedenle, IBM MQ , bazıları GDPR ' ye tabi olabilecek birçok veri biçimini değiştirmek için kullanılabilir. IBM MQ ' in veri alışverişinde bulunabileceği birkaç üçüncü kişi ürünü vardır. Bunlardan bazıları IBM' e aittir, ancak diğerleri diğer teknoloji tedarikçileri tarafından sağlanır. [Software Product Compatibility Reports](#) (Yazılım Ürünü Uyumluluk Raporları) web sitesi, ilişkili yazılımların listelerini sağlar. Bir üçüncü kişi ürününün GDPR ' ye hazır olup olmamasına ilişkin dikkat edilmesi gereken noktalar için bu ürünün belgelerine bakmanız gerekir. IBM MQ yöneticileri, IBM MQ ' in veri geçişiyle nasıl etkileşimde bulunacağı, kuyrukların, konuların ve aboneliklerin tanımına göre denetler.

IBM MQ aracılığıyla hangi veri akışı tipleri?

IBM MQ uygulama verileri için zamanuyumsuz ileti sistemi hizmeti sağladığından, kullanım senaryoları uygulama devreye alınmasına göre değiştiğinden bu soruya kesin bir yanıt yoktur. Uygulama iletileri verileri kuyruk dosyalarında (z/OS üzerinde sayfa kümeleri ya da Coupling Facility), günlüklerde ve arşivlerde saklanır ve iletinin kendisi GDPR tarafından yönetilen veriler içerebilir. Uygulama tarafından sağlanan ileti verileri, hata günlükleri, izleme dosyaları ve FFSTler gibi sorun belirleme amacıyla toplanan dosyalara dahil edilebilir. z/OS uygulaması için sağlanan ileti verileri adres alanına ya da Coupling Facility dökümlerine de eklenebilir.

Aşağıda, IBM MQ kullanılarak değiş tokuş edilebilecek bazı tipik kişisel veri örnekleri verilmiştir:

- Müşterinin çalışanları (örneğin; IBM MQ , müşterinin bordro ya da İK sistemleri arasında bağlantı kurmak için kullanılabilir)
- Müşterinin kendi müşterilerinin kişisel verileri (örneğin; IBM MQ , bir müşteri tarafından, satış fırsatlarını almak ve Müşteri İlişkileri Yönetimi sistemi içinde veri depolamak gibi, müşterileriyle ilgili uygulamalar arasında veri alışverişi yapmak için kullanılabilir).
- Müşterinin kendi müşterilerinin hassas kişisel verileri (örneğin; IBM MQ , klinik uygulamalarını bütünleştirirken HL7-based sağlık hizmetleri kayıtları gibi kişisel verilerin değiş tokuş alınmasını gerektiren sektör bağlamında kullanılabilir).

IBM MQ , uygulama tarafından sağlanan ileti verilerine ek olarak aşağıdaki veri tiplerini de işler:

- Kimlik Doğrulama Kimlik Bilgileri (kullanıcı adı ve parolalar, API anahtarları vb.)
- Teknik Olarak Tanımlanmış Kişisel Bilgiler (örneğin, aygıt tanıtıcıları, kullanıma dayalı tanıtıcılar, IP adresi, vb.) -bir kişiye bağlandığında)

IBM ile çevrimiçi iletişim için kullanılan kişisel veriler

IBM MQ müşterileri, öncelikle aşağıdakiler olmak üzere çeşitli yollarla IBM MQ konularına ilişkin çevrimiçi yorumlar/geri beslemeler/istekler gönderebilirler:

- [IBM MQ alanı IBM Developer](#) içindeki sayfalarda genel yorumlar alanı
- [IBM MQ ürün bilgileri IBM Documentation](#) sayfalarındaki genel yorumlar alanı
- [IBM Destek Forumları](#) içindeki genel yorumlar
- [IBM Integration Ideas](#) içindeki genel yorumlar

Genellikle iletişim konusu için kişisel yanıtların etkinleştirilmesi amacıyla yalnızca müşteri adı ve e-posta adresi kullanılır ve kişisel veri kullanımı [IBM Çevrimiçi Gizlilik Bildirimi](#)'ne uygun olur.

Veri Toplama

IBM MQ kişisel verileri toplamak için kullanılabilir. IBM MQ kullanımınızı değerlendirirken ve GDPR 'nin taleplerini karşılama gereksinimlerinizi değerlendirirken, sizin durumunuzda IBM MQ'inden geçen kişisel veri tiplerini göz önünde bulundurmanız gerekir. Aşağıdakiler gibi yönleri göz önünde bulundurmak isteyebilirsiniz:

- Veriler kuyruk yöneticilerinize nasıl ulaşır? (Hangi protokollerde? Veriler şifreli mi? Veriler imzalandı mı?)
- Veriler kuyruk yöneticilerinizden nasıl gönderilir? (Hangi protokollerde? Veriler şifreli mi? Veriler imzalandı mı?)
- Bir kuyruk yöneticisinden geçerken veriler nasıl depolanır? (Herhangi bir ileti sistemi uygulaması, bir ileti kalıcı olmasa da, ileti verilerini durumlu ortama yazma potansiyeline sahiptir. İleti sistemi özelliklerinin, üründen geçen uygulama iletileri verilerinin çeşitli yönlerini nasıl ortaya çıkarabileceğini biliyor musunuz?)
- IBM MQ 'in üçüncü kişi uygulamalarına erişmek için gereksinim duyduğu yerde kimlik bilgileri nasıl toplanır ve depolanır?

IBM MQ 'in kimlik doğrulaması gerektiren diğer sistemler ve hizmetlerle (örneğin, LDAP) iletişim kurması gerekebilir. Gerektiğinde, kimlik doğrulama verileri (kullanıcı kimlikleri, parolalar) bu tür iletişimlerde kullanılmak üzere IBM MQ tarafından yapılandırılır ve depolanır. Mümkün olduğunda, IBM MQ kimlik doğrulaması için kişisel kimlik bilgilerini kullanmaktan kaçınmalısınız. Kimlik doğrulama verileri için kullanılan depolamanın korunmasını göz önünde bulundurun. (Aşağıdaki Veri Depolama bölümüne bakın.)

Veri Depolama

İleti verileri kuyruk yöneticileri arasında dolaşırken, IBM MQ bu verileri doğrudan durumlu ortama aktarmaya devam eder (belki birden çok kopya). IBM MQ kullanıcıları, dururken ileti verilerinin güvenliğini sağlamak isteyebilir.

Aşağıdaki öğeler, IBM MQ 'in uygulama tarafından sağlanan ve kullanıcıların GDPR'ye uygunluğu sağlarken göz önünde bulundurmak isteyecekleri verileri belirttiği alanları vurgular.

- Uygulama İleti Kuyrukları:

IBM MQ , uygulamalar arasında zamanuyumsuz veri alışverişine izin vermek için ileti kuyrukları sağlar. Bir kuyruksa saklanan kalıcı olmayan ve kalıcı olmayan iletiler, durumlu ortama yazılır.

- Dosya Aktarma Aracısı Kuyrukları:

IBM MQ Managed File Transfer , dosya verilerinin güvenilir aktarımını koordine etmek için ileti kuyruklarını kullanır; kişisel verileri içeren dosyalar ve aktarma kayıtları bu kuyruklarda saklanır.

- İletim Kuyrukları:

İletileri kuyruk yöneticileri arasında güvenilir bir şekilde aktarmak için, iletiler iletim kuyruklarında geçici olarak saklanır.

- Gönderilmeyen Mektup Kuyrukları:

Kuyruk yöneticisinde böyle bir kuyruk yapılandırıldıysa, iletilerin hedef kuyruğa konamayacağı ve gitmeyen iletiler kuyruğunda saklanamayacağı bazı durumlar vardır.

- Geriletme Kuyrukları:

JMS ve XMS ileti sistemi arabirimleri, diğere geçerli iletilerin işlenmesine izin vermek için bir dizi geriletme gerçekleştirildikten sonra zehirli iletilerin bir geriletme kuyruğuna taşınmasına olanak sağlayan bir yetenek sağlar.

- **AMS Hata Kuyruğu:**

IBM MQ Advanced Message Security , bir güvenlik ilkesine uymayan iletileri SYSTEM.PROTECTION.ERROR.QUEUE hata kuyruğu, hatalı kuyruğa alma yöntemine benzer bir şekilde.

- **Alıkonan Yayınlar:**

IBM MQ , abone olan uygulamaların önceki bir yayını geri çağırmasına izin vermek için korunan bir yayın özelliği sağlar.

- **Ertelenen Teslimat:**

IBM MQ , iletilerin gelecekteki bir zamanda hedeflerine teslim edilmesini sağlayan JMS 2.0 ve Jakarta Messaging 3.0 teslimat gecikmesi özelliğini destekler. Henüz teslim edilmemiş iletiler, SYSTEM.DDELAY.LOCAL.QUEUE (KUYRUK).

Daha fazla bilgi:

- [Günlüğe kaydetme: İletilerin kaybolmadığından emin olun](#)
- [MFT Agent kuyruk ayarları](#)
- [İleti gönderilmeyen iletiler kuyruğunun kullanılması](#)
- [JMS için IBM MQ sınıflarında zehirli iletilerin işlenmesi](#)
- [AMS hata işleme](#)
- [Alıkonan yayınlar](#)
- [JMS 2.0 teslim gecikmesi](#)

Aşağıdaki öğeler, IBM MQ 'in kullanıcıların GDPR' ye uygunluğu sağlarken dikkate almak isteyecekleri, uygulama tarafından sağlanan verileri dolaylı olarak sürdürebileceği alanları vurgular.

- **İzleme rotası ileti alışverişi:**

IBM MQ , bir iletinin uygulamalar arasında izlediği rotayı kaydeden izleme rotası yetenekleri sağlar. Oluşturulan olay iletileri, IP adresleri gibi teknik olarak tanımlanabilir kişisel bilgileri içerebilir.

- **Uygulama etkinliği izlemesi:**

IBM MQ , uygulamaların ve kanalların ileti sistemi API etkinliklerini kaydeden uygulama etkinliği izlemesini sağlar; uygulama etkinliği izlemesi, uygulama tarafından sağlanan ileti verilerinin içeriğini olay iletilerine kaydedebilir.

- **Hizmet izleme:**

IBM MQ , ileti verilerinin aktığı iç kod yollarını kaydeden hizmet izleme özellikleri sağlar. Bu özelliklerin bir parçası olarak IBM MQ , uygulama tarafından sağlanan ileti verilerinin içeriğini diskte saklanan izleme dosyalarına kaydedebilir.

- **Kuyruk yöneticisi olayları:**

IBM MQ , yetki, komut ve yapılandırma olayları gibi kişisel verileri içerebilecek olay iletileri oluşturabilir.

Daha fazla bilgi:

- [İzleme rotası ileti alışverişi](#)
- [İzleme özelliğinin kullanılması](#)
- [Olay izleme](#)
- [Kuyruk yöneticisi olayları](#)

Uygulamanın sağladığı ileti verilerinin kopyalarına erişimi korumak için aşağıdaki eylemleri göz önünde bulundurun:

- Ayrıcalıklı kullanıcı erişimini dosya sistemindeki IBM MQ verileriyle sınırlandırın; örneğin, UNIX and Linux® platformlarında 'mqm' grubunun kullanıcı üyeliğini kısıtlayın.

- Özel olarak ayrılmış kuyruklar ve erişim denetimi aracılığıyla IBM MQ verilerine uygulama erişimini kısıtlayın. Uygun olduğunda, kuyruklar gibi kaynakların uygulamalar arasında gereksiz yere paylaşılmasından kaçınin ve kuyruk ve konu kaynakları için ayrıntılı erişim denetimi sağlayın.
- Yüksek kullanılabilirlikli (HA) ya da olağanüstü durumdan kurtarma (DR) yapılandırmalarında IBM MQ verilerinin eşlenmiş kopyalarına erişimi kısıtlayın ve eşleme için kullanılan bağlantıların güvenliğini sağlayın.
- İleti verilerinin uçtan uca imzalanmasını ve/veya şifrenmesini sağlamak için IBM MQ Advanced Message Security olanağını kullanın.
- IBM MQ verilerini, izlemesini ya da günlüklerini içerebilecek dizinleri ya da dosya sistemlerini korumak için dosya ya da birim düzeyinde şifrelemeyi kullanın.
- Hizmet izlemesini IBM' e yükledikten sonra, kişisel verileri içerebilecek içerikle ilgili endişeleriniz varsa, hizmet izleme dosyalarınızı ve FFST verilerinizi silebilirsiniz.

Daha fazla bilgi:

- [Ayrıcalıklı kullanıcılar](#)
- [Multiplatforms üzerinde planlama dosya sistemi desteği](#)
- [IBM MQ Appliance üzerinde dosya sistemi şifrelemesi](#)

IBM MQ yöneticisi, kimlik bilgileriyle (kullanıcı adı ve parola, API anahtarları, vb.) bir kuyruk yöneticisini yapılandırabilir LDAP, Salesforce gibi 3rd kişi hizmetleri için Bu veriler genellikle, dosya sistemi izinleriyle korunan kuyruk yöneticisi veri dizininde saklanır.

Bir IBM MQ kuyruk yöneticisi yaratıldığında, veri dizini, IBM MQ ' un yapılandırma dosyalarını okuyabileceği ve bu sistemlere bağlanmak için kimlik bilgilerini kullanabileceği grup tabanlı erişim denetimiyle ayarlanır. IBM MQ yöneticileri ayrıcalıklı kullanıcılar olarak kabul edilir ve bu grubun üyeleri olduğu için dosyalara okuma erişimine sahip olur. Bazı dosyalar gizlenmiş, ancak şifrenmemiş. Bu nedenle, kimlik bilgilerine erişimi tam olarak korumak için aşağıdaki işlemleri göz önünde bulundurmanız gerekir:

- Ayrıcalıklı kullanıcı erişimini IBM MQ verileriyle sınırlandırın; örneğin, UNIX and Linux platformlarında 'mqm' grubunun üyeliğini kısıtlayın.
- Kuyruk yöneticisi veri dizininin içeriğini korumak için dosya ya da birim düzeyinde şifreleme kullanın.
- Üretim yapılandırma dizininin yedeklerini şifreleyin ve bunları uygun erişim denetimleriyle saklayın.
- Güvenlik, komut ve yapılandırma olaylarıyla kimlik doğrulama hatası, erişim denetimi ve yapılandırma değişiklikleri için denetim izleri sağlamayı düşünün.

Daha fazla bilgi:

- [IBM MQ' nun Güvenliğinin Sağlama](#)

Veri Erişimi

IBM MQ kuyruk yöneticisi verilerine, bazıları uzak bağlantı üzerinden erişim için, bazıları da yerel bağlantı üzerinden erişim için tasarlanmış olan aşağıdaki ürün arabirimleri aracılığıyla erişilebilir.

- IBM MQ Konsol [Yalnızca Uzak]
- IBM MQ Yönetim REST API 'si [Yalnızca Uzak]
- IBM MQ Messaging REST API [Yalnızca Uzak]
- MQI [Yerel ve Uzak]
- JMS [Yerel ve Uzak]
- XMS [Yerel ve Uzak]
- IBM MQ Telemetry (MQTT) [Yalnızca Uzak]
- IBM MQ Light (AMQP) [Yalnızca Uzak]
- IBM MQ IMS köprüsü [Yalnızca Yerel]
- IBM MQ CICS köprüsü [Yalnızca Yerel]

- IBM MQ MFT İletişim Kuralı köprüleri [Yalnızca Uzak]
- IBM MQ Connect:Direct köprüleri [Yalnızca Uzak]
- IBM MQ Köprüden Salesforce ' a [Yalnızca Uzak]
- IBM MQ Köprüden Blockchain [Yalnızca Uzak]
- IBM MQ MQAI [Yerel ve Uzak]
- IBM MQ PCF komutları [Yerel ve Uzak]
- IBM MQ MQSC komutları [Yerel ve Uzak]
- IBM MQ Explorer [Yerel ve Uzak]
- IBM MQ Kullanıcı Çıkışları [Yalnızca Yerel]
- IBM MQ Internet Pass-Thru [Yalnızca Uzak]
- Red Hat® OpenShift® İzleme (Prometheus) metrikleri (metrikler kuyruk yöneticisi istatistiklerine ilişkin sayısal verilerdir)
- **Removed** **Deprecated** Üst düzey izleme verilerini merkezi bir kaynağa gönderen IBM Cloud Pak for Integration Operations Dashboard bütünleşmesi (yalnızcaCP4I). Bu özelliğin IBM MQ Operator 2.3.0sürümünde kullanımdan kaldırıldığını ve IBM MQ Operator 2.4.0sürümünde kaldırıldığını unutmayın.
- IBM MQ Appliance Dizisel Konsol [Yalnızca Yerel]
- IBM MQ Appliance SSH [Yalnızca Uzak]
- IBM MQ Appliance REST API [Yalnızca Uzak]
- IBM MQ Appliance Web UI [Yalnızca Uzak]
- **V 9.3.3** IBM MQ Kafka Bağlayıcılar (Kafka Connect) [Yerel ve Uzak]

Arabirimler, kullanıcıların bir IBM MQ kuyruk yöneticisinde ve üzerinde saklanan iletilerde değişiklik yapmalarını sağlamak üzere tasarlanmıştır. Yönetim ve ileti sistemi işlemleri, bir istek yapıldığında üç aşamaya dahil olmak üzere güvence altına alınır;

- Kimlik Doğrulaması
- Rol eşleme
- Yetkilendirme

Kimlik denetimi:

İleti ya da denetim işlemi yerel bir bağlantıdan istendiyse, bu bağlantının kaynağı aynı sistemde çalışan bir işlemdir. İşlemi çalıştıran kullanıcı, işletim sistemi tarafından sağlanan kimlik doğrulama adımlarını geçmiş olmalıdır. Bağlantının yapıldığı sürecin sahibinin kullanıcı adı kimlik olarak bildirilir. Örneğin, bu, bir uygulamanın başlatıldığı kabuğu çalıştıran kullanıcının adı olabilir. Yerel bağlantılar için olası kimlik doğrulama biçimleri şunlardır:

1. Bildirili kullanıcı adı (yerel işletim sistemi)
2. İsteğe bağlı kullanıcı adı ve parola (OS, LDAP ya da özel 3rd taraf havuzları)
3. Yalnızca Güvenlik Simgesi (JWT) IBM MQ ve yalnızca IBM MQ 9.3.4 içinden

Denetim işlemi uzak bir bağlantıdan istendiyse, IBM MQ ile iletişim bir ağ arabirimi üzerinden kurulur. Ağ bağlantıları aracılığıyla kimlik doğrulaması için aşağıdaki kimlik formları sunulabilir;

1. Bildiri kullanıcı adı (uzak işletim sisteminden)
2. Kullanıcı adı ve parola (OS, LDAP ya da özel 3rd taraf havuzları)
3. Kaynak ağ adresi (IP adresi gibi)
4. X.509 Sayısal Sertifika (karşılıklı SSL/TLS kimlik doğrulaması)
5. Güvenlik belirteçleri (LTPA2 simgesi ya da JWT simgesigibi)
6. Diğer özel güvenlik (3rd kişi çıkışları tarafından sağlanan yetenek)

7. SSH anahtarları

IBM MQ' in IBM Cloud Pak for Integration ile bütünleştirilmesi, IBM MQ Console için yeni bir kimlik doğrulama tipi ekler: Cloud Pak ile Tekli Oturum Açma. (Yalnızca CP4I)

Rol eşleme:

Rol eşleme aşamasında, kimlik doğrulama aşamasında sağlanan kimlik bilgileri diğer bir kullanıcı kimliğiyle eşlenebilir. Eşlenen kullanıcı kimliğinin devam etmesine izin verildiği sürece (örneğin, yönetimle görevli kullanıcılar kanal kimlik doğrulama kuralları tarafından engellenebilir), eşlenen kullanıcı kimliği IBM MQ kaynaklarına ilişkin etkinlikler yetkilendirilirken son aşamaya taşınır.

Yetki:

IBM MQ , farklı kullanıcıların kuyruklar, konular ve diğer kuyruk yöneticisi nesnelere gibi farklı ileti kaynakları için farklı yetkilere sahip olmalarını sağlar.

Günlüğe kaydetme etkinliği:

IBM MQ ' in bazı kullanıcılarının MQ kaynaklarına erişim için bir denetim kaydı yaratmaları gerekebilir. İstenen denetim günlüklerine örnek olarak, değişikliği kimin istediğine ek olarak değişikliğe ilişkin bilgiler de içeren yapılandırma değişiklikleri gösterilebilir.

Bu gereksinimi gerçekleştirmek için aşağıdaki bilgi kaynakları kullanılabilir:

1. Bir yönetici komutu başarıyla çalıştırıldığında, IBM MQ kuyruk yöneticisi komut olayları üretecek şekilde yapılandırılabilir.
2. Bir IBM MQ kuyruk yöneticisi, bir kuyruk yöneticisi kaynağı yaratıldığında, değiştirildiğinde ya da silindiğinde yapılandırma olayları üretecek şekilde yapılandırılabilir.
3. IBM MQ kuyruk yöneticisi, bir kaynak için yetki denetimi başarısız olduğunda bir yetki olayı üretecek şekilde yapılandırılabilir.
4. Başarısız yetki denetimlerinin kuyruk yöneticisi hata günlüklerine yazıldığını gösteren hata iletileri.
5. IBM MQ Console, kimlik doğrulama, yetkilendirme denetimleri başarısız olduğunda ya da kuyruk yöneticileri oluşturulduğunda, başlatıldığında, durdurulduğunda ya da silindiğinde denetleme iletilerini günlüklerine yazar.
6. IBM MQ Appliance , kullanıcı oturum açmaları ve sistem değişikliklerini kaydetmek için denetleme iletilerini günlüklerine yazar.

IBM MQ kullanıcıları, bu tür çözümleri göz önünde bulundurarak aşağıdaki noktaları dikkate almak isteyebilir:

- Olay iletileri kalıcı değildir; bu nedenle, kuyruk yöneticisi bilgileri yeniden başlattığında bilgiler kaybolur. Herhangi bir olay izleyicisi, var olan iletileri sürekli olarak kullanacak ve içeriği kalıcı ortama aktarabilecek şekilde yapılandırılmalıdır.
- IBM MQ ayrıcalıklı kullanıcılarının devre dışı bırakılan olaylar için yeterli ayrıcalıkları vardır, günlükleri temizleyin ya da kuyruk yöneticilerini silin.

IBM MQ verilerine erişimin güvenliğini sağlama ve bir denetim izi sağlama hakkında daha fazla bilgi için aşağıdaki konulara bakın:

- [IBM MQ güvenlik mekanizmaları](#)
- [Yapılandırma olayları](#)
- [Komut olayları](#)
- [Hata günlüklerinin kullanılması](#)

Verilerin İşlenmesi

Genel Anahtar Altyapısı kullanılarak şifreleme:

Bağlantıların TLS kullandığını belirterek IBM MQ ile ağ bağlantılarının güvenliğini sağlayabilirsiniz; bu, bağlantının başlangıç tarafının karşılıklı kimlik doğrulamasını da sağlayabilir.

İletim mekanizmaları tarafından sağlanan PKI güvenlik olanaklarının kullanılması, IBM MQ ile veri işleminin güvenli kılınmasına yönelik ilk adımdır. Ancak, daha fazla güvenlik özelliği etkinleştirilmeden, tüketen bir uygulamanın davranışı, iletinin kökenini doğrulamadan veya geçiş sırasında değiştirilip değiştirilmediğini doğrulamadan ona teslim edilen tüm iletileri işlemektir.

Advanced Message Security (AMS) yeteneklerini kullanmak üzere lisanslanan IBM MQ kullanıcıları, güvenlik ilkelerinin tanımlanması ve yapılandırılması aracılığıyla uygulamaların iletelerde tutulan kişisel verileri nasıl işleyeceklerini denetleyebilirler. Güvenlik ilkeleri, uygulamalar arasında ileti verilerine dijital imzalama ve/veya şifreleme uygulanmasını sağlar.

İletilerin gerçek olduğundan emin olmak için iletileri kullanırken dijital imzayı zorunlu tutmak ve doğrulamak için güvenlik ilkeleri kullanılabilir. AMS şifrelemesi, ileti verilerinin okunabilir bir formdan kodlanmış bir sürüme dönüştürülmesi için bir yöntem sağlar. Bu yöntem, yalnızca amaçlanan alıcı ya da iletise ve doğru şifre çözme anahtarına erişimi varsa başka bir uygulama tarafından çözülebilir.

Ağ bağlantılarınızın güvenliğini sağlamak için SSL ve sertifikaların kullanılmasıyla ilgili ek bilgi için IBM MQ ürün belgelerinde aşağıdaki konulara bakın:

- [IBM MQ](#)
- [AMS ' ye Genel Bakış](#)

Verilerin Silinmesi

IBM MQ , ürüne sağlanan verileri silmek için komutlar ve kullanıcı arabirimi işlemleri sağlar. Bu, IBM MQ kullanıcılarının belirli kişilerle ilgili verileri silebileceği anlamına gelir (bu gerekiyorsa).

- GDPR Müşteri Verilerini silmeye uymak için göz önünde bulundurulması gereken IBM MQ davranış alanları
 - Uygulama kuyruğunda saklanan ileti verilerini silmek için:
 - İleti alışverişi API 'sini ya da araçlarını kullanarak ya da ileti süre bitimini kullanarak tek tek iletileri kaldırma.
 - İletilerin kalıcı olmadığını, kalıcı olmayan ileti sınıfının normal olduğu bir kuyrukta tutulduğunu ve kuyruk yöneticisini yeniden başlattığı belirtiliyordu.
 - Yönetim tarafından kuyruk temizleniyor.
 - Kuyruk siliniyor.
 - Bir konuda saklanan alıkonan yayın verilerini silmek için:
 - İletilerin kalıcı olmadığını belirtme ve kuyruk yöneticisini yeniden başlatma.
 - Alıkonan verileri yeni verilerle ya da ileti süre bitimini kullanarak değiştirme.
 - Yönetimsel olarak konu dizesi temizleniyor.
 - Yüksek kullanılabilirlik ya da olağanüstü durumdan kurtarma için tüm kuyruk yöneticisini ve eşlenen kopyaları silerek bir kuyruk yöneticisinde saklanan verileri silin.
 - İzleme dizinindeki dosyaları silerek, Hizmet izleme komutlarıyla saklanan verileri silin.
 - Hatalar dizinindeki dosyaları silerek saklanan FFST verilerini silin.
 - Adres alanını ve Coupling Facility dökümlerini silin (z/OSüzerinde).
 - Bu tür verilerin arşivini, yedeklemesini ya da diğer kopyalarını silin.
- GDPR Hesap Verilerini silmeye uymak için göz önünde bulundurulması gereken IBM MQ davranış alanları
 - IBM MQ tarafından kuyruk yöneticilerine ve 3rd taraf hizmetlerine bağlanmak için depolanan hesap verilerini ve tercihlerini silmek için bunları silebilirsiniz (arşiv, yedekleme ya da başka bir şekilde eşlenmiş kopyaları da içinde olmak üzere):
 - Kimlik bilgilerini saklayan kuyruk yöneticisi kimlik doğrulama bilgileri nesnelere.
 - Kullanıcı tanıtıcılarına başvuran kuyruk yöneticisi yetkisi kayıtları.

- Belirli IP adreslerini, sertifika DN ' lerini ya da kullanıcı tanıtıcılarını eşleyen ya da engelleyen kuyruk yöneticisi kanalı kimlik doğrulama kuralları.
- Kuyruk yöneticisi ve dosya sunucularıyla kimlik doğrulaması için IBM MQ Managed File Transfer Agent, Logger ve MQ Explorer MFT Plugin tarafından kullanılan kimlik bilgileri dosyaları.
- X.509 , SSL/TLS bağlantıları ya da IBM MQ Advanced Message Security (AMS) tarafından kullanılabilen anahtar depolarından bir kişiyi temsil eden ya da bu kişiye ilişkin bilgileri içeren sayısal sertifikalar.
- Sistem günlük dosyalarında bu hesaplara başvuru da dahil olmak üzere, IBM MQ Appliance' den tek tek kullanıcı hesapları.
- IBM MQ Explorer çalışma alanı meta verileri ve Eclipse ayarları.
- IBM MQ Explorer Parola Deposu, Parola Tercihleri' nde belirtildiği şekilde.
- IBM MQ Konsol ve mqweb sunucusu yapılandırma dosyaları.
- Salesforce bağlantı verileri yapılandırma dosyaları.
- Blockchain bağlantı verileri yapılandırma dosyaları.
- IBM MQ Internet Pass-Thru yapılandırma dosyaları ve anahtar depoları.

Daha fazla bilgi:

- IBM MQ Bridge 'in Salesforceolarak yapılandırılması
- IBM MQ ' nun Blockchain ile kullanılmak üzere yapılandırılması
- MFT ve IBM MQ bağlantı kimlik doğrulaması
- ProtocolBridgeCredentials.xml dosyasını kullanarak bir dosya sunucusuna ilişkin kimlik bilgilerini eşleme
- IBM MQ Console kullanıcılarını ve rollerini yapılandırma

Verilerin İzlenmesi

IBM MQ , kullanıcıların uygulamaların ve kuyruk yöneticilerinin nasıl performans gösterdiğini daha iyi anlamak için kullanabilecekleri bir dizi izleme özelliği sağlar.

IBM MQ , kuyruk yöneticisi hata günlüklerinin yönetilmesine yardımcı olan bir dizi özellik de sağlar.

Daha fazla bilgi:

- IBM MQ ağınızın izlenmesi
- Tanımlama iletisi hizmetleri
- QMErrorLog hizmeti
- IBM MQ Appliance izleme ve raporlama

Kişisel Verilerin Kullanımını Kısıtlama yeteneği

IBM MQ , bu belgede özetlenen olanakları kullanarak bir son kullanıcının kişisel verilerinin kullanımını kısıtlamasını sağlar.

IBM MQ ileti kuyrukları, bir veritabanıyla aynı şekilde kalıcı veri deposu olarak kullanılmamalıdır; bu, özellikle GDPR ' ye tabi uygulama verileri işlenirken geçerlidir.

Verilerin arama sorgusuyla bulunabileceği bir veritabanından farklı olarak, bir iletinin kuyruğunu, ileti ve ilinti tanıtıcılarını bilmediğiniz sürece ileti verilerini bulmak zor olabilir.

Bir kişinin verilerini içeren iletiler kolayca tanımlanabilir ve bulunabilir; ileti verilerine erişmek ya da ileti verilerini değiştirmek için standart IBM MQ ileti sistemi özellikleri kullanılabilir.

Dosya işleme

1. IBM MQ Managed File Transfer , aktarılan dosyalarda kötü amaçlı yazılım taraması gerçekleştirmez. Dosyalar olduğu gibi aktarılır ve aktarma sırasında dosya verilerinin değiştirilmediğinden emin olmak için bir bütünlük denetimi gerçekleştirilir. Kaynak ve hedef sağlama toplamları, aktarım durumu yayınının bir parçası olarak yayınlanır. Son kullanıcıların, MFT dosyayı aktarmadan önce ve MFT bir dosyayı uzak uç noktaya teslim ettikten sonra ortamlarına uygun şekilde kötü amaçlı yazılım taraması gerçekleştirmeleri önerilir.
2. IBM MQ Managed File Transfer , MIME tipine ya da dosya uzantısına dayalı olarak işlem yapmaz. MFT , dosyaları okur ve baytları tam olarak giriş dosyasından okunduğu gibi aktarır.

Tek bir kuyruk yöneticisine dayalı mimariler

En basit IBM MQ mimarileri, tek bir kuyruk yöneticisinin yapılandırılmasını ve kullanılmasını içerir.

IBM MQ mimarinizi planlamadan önce, temel IBM MQ kavramlarını tanıyın. Bkz. [IBM MQ Teknik genel bakış](#).

Aşağıdaki kısımlarda, tek bir kuyruk yöneticisini kullanan birkaç olası mimari açıklanmıştır:

- [“Bir hizmete erişen yerel uygulamalarla tek kuyruk yöneticisi” sayfa 18](#)
- [“Bir hizmete istemci olarak erişen uzak uygulamaları içeren tek kuyruk yöneticisi” sayfa 18](#)
- [“Yayınlama/abone olma yapılandırmasına sahip tek kuyruk yöneticisi” sayfa 18](#)

Bir hizmete erişen yerel uygulamalarla tek kuyruk yöneticisi

Tek bir kuyruk yöneticisine dayalı ilk mimari, bir hizmete erişen uygulamaların hizmeti sağlayan uygulamalarla aynı sistemde çalıştığı yerdir. IBM MQ kuyruk yöneticisi, hizmeti isteyen uygulamalar ile hizmeti sağlayan uygulamalar arasında zamanuyumsuz iletişim sağlar. Bu, uygulamalardan biri uzun bir süre çevrimdışı olsa bile uygulamalar arasındaki iletişimin devam edebileceği anlamına gelir.

Bir hizmete istemci olarak erişen uzak uygulamaları içeren tek kuyruk yöneticisi

Tek bir kuyruk yöneticisine dayalı ikinci mimari, hizmeti sağlayan uygulamalardan uzaktan çalışan uygulamalara sahiptir. Uzak uygulamalar, hizmetler için farklı sistemlerde çalışıyor. Uygulamalar, istemciler olarak tek bir kuyruk yöneticisine bağlanır. Bu, bir hizmete erişimin tek bir kuyruk yöneticisi aracılığıyla birden çok sisteme sağlanabileceği anlamına gelir.

Bu mimarinin bir sınırlaması, bir uygulamanın çalışması için bir ağ bağlantısının kullanılabilir olması gerektirir. Uygulama ile kuyruk yöneticisi arasındaki ağ bağlantısı üzerinden etkileşim zamanuyumludur.

Yayınlama/abone olma yapılandırmasına sahip tek kuyruk yöneticisi

Tek bir kuyruk yöneticisini kullanan diğer bir mimari, yayınlama/abone olma yapılandırması kullanmaktır. Yayınlama/abone olma mesajında, bilgi sağlayıcıyı bu bilgilerin kullanıcılarından ayırabilir. Bu, uygulamaların hedef uygulamayla ilgili bilgileri (örneğin, iletilerin yerleştirileceği kuyruk adı) bilmesi gereken, önceden tanımlanan mimarilerde noktadan noktaya ileti alışverişi biçemlerinden farklıdır. IBM MQ yayınlama/abone olma özelliğini kullanarak gönderen uygulama, bilgilerin konusuna dayalı olarak belirtilen bir konuyu içeren bir ileti yayınlar. IBM MQ , bir abonelik aracılığıyla ilgili konuya ilgi duyan uygulamalara iletinin dağıtımını işler. Alan uygulamaların, bunları alacak iletilerin kaynağı hakkında herhangi bir şey bilmeleri gerekmez. Daha fazla bilgi için bkz. [Yayınlama/abone olma ileti sistemi](#) ve [Tek bir kuyruk yöneticisi yayınlama/abone olma yapılandırması örneği](#).

İlgili kavramlar

[IBM MQ ' a Giriş](#)

İlgili görevler

[“IBM MQ mimarisinin planlanması” sayfa 5](#)

IBM MQ ortamınızı planlarken, IBM MQ ' un tek ve birden çok kuyruk yöneticisi mimarisi ve noktadan noktaya iletişim ve yayınlama/abone olma ileti sistemi stilleri için sağladığı desteği göz önünde bulundurun. Ayrıca, kaynak gereksinimlerinizi ve günlük kaydı ve yedekleme olanaklarını kullanımınızı da planlayın.

[Çoklu platformlarda kuyruk yöneticilerinin yaratılması ve yönetilmesi](#)

Birden çok kuyruk yöneticisine dayalı mimariler

Birden çok kuyruk yöneticisinin yapılandırılmasını ve kullanılmasını içeren bir IBM MQ mimarisi yaratmak için dağıtılmış ileti kuyruklama tekniklerini kullanabilirsiniz.

IBM MQ mimarinizi planlamadan önce, temel IBM MQ kavramlarını tanıyın. Bkz. [IBM MQ Teknik genel bakış](#).

IBM MQ mimarisi, ek kuyruk yöneticileri eklenerek, hizmet sağlayan uygulamalarda değişiklik yapılmaksızın değiştirilebilir.

Uygulamalar, kuyruk yöneticisiyle aynı makinede barındırılabilir ve daha sonra başka bir sistemdeki başka bir kuyruk yöneticisinde barındırılan bir hizmetle zamanuyumsuz iletişim kurabilir. Diğer bir seçenek olarak, bir hizmete erişen uygulamalar, başka bir kuyruk yöneticisinde hizmete zamanuyumsuz erişim sağlayan bir kuyruk yöneticisine istemci olarak bağlanabilir.

Farklı kuyruk yöneticilerini ve kuyruklarını bağlayan rotalar, dağıtılmış kuyruğa alma teknikleri kullanılarak tanımlanır. Mimari içindeki kuyruk yöneticileri kanallar kullanılarak bağlanır. Kanallar, kuyruk yöneticilerinin yapılandırılmasına bağlı olarak iletileri bir kuyruk yöneticisinden diğerine otomatik olarak taşımak için kullanılır.

IBM MQ ağı planlamasına ilişkin üst düzey genel bilgiler için bkz. [“Dağıtılmış kuyruk yöneticisi ağlarının tasarlanması” sayfa 21](#).

IBM MQ mimarinize ilişkin kanalların planlanması hakkında bilgi için bkz. [IBM MQ dağıtılmış kuyruğa alma teknikleri](#).

Dağıtılmış kuyruk yönetimi, kuyruk yöneticileri arasındaki iletişimi oluşturmanızı ve izlemenizi sağlar. Dağıtılmış kuyruk yönetimi hakkında daha fazla bilgi için bkz. [Dağıtılmış kuyruk yönetimine giriş](#).

İlgili görevler

[“IBM MQ mimarisinin planlanması” sayfa 5](#)

IBM MQ ortamınızı planlarken, IBM MQ ' un tek ve birden çok kuyruk yöneticisi mimarisi ve noktadan noktaya iletişim ve yayınlama/abone olma ileti sistemi stilleri için sağladığı desteği göz önünde bulundurun. Ayrıca, kaynak gereksinimlerinizi ve günlük kaydı ve yedekleme olanaklarını kullanımınızı da planlayın.

[Çoklu platformlarda kuyruk yöneticilerinin yaratılması ve yönetilmesi](#)

Dağıtılmış kuyruklarınızın ve kümelerinizin planlanması

Dağıtılmış kuyruk yöneticilerinin barındırdığı kuyrukları el ile bağlayabilir ya da bir kuyruk yöneticisi kümesi yaratabilir ve ürünün kuyruk yöneticilerini sizin için bağlamasına izin verebilirsiniz. Dağıtılmış ileti sistemi ağınıza için uygun bir topoloji seçmek üzere el ile denetim, ağ boyutu, değişiklik sıklığı, kullanılabilirlik ve ölçeklenebilirlik gereksinimlerini göz önünde bulundurmanız gerekir.

Başlamadan önce

Bu görev, dağıtılmış ileti sistemi ağlarının ne olduğunu ve bunların nasıl çalıştığını anladığınızı varsayar. Teknik bir genel bakış için bkz. [Dağıtılmış kuyruğa alma ve kümeler](#).

Bu görev hakkında

Dağıtılmış bir ileti sistemi ağı yaratmak için, kanalları farklı kuyruk yöneticilerinin barındırdığı kuyrukları bağlayacak şekilde el ile yapılandırabilir ya da bir kuyruk yöneticisi kümesi yaratabilirsiniz. Kümeleme,

kuyruk yöneticilerinin, ek kanal tanımlamaları ya da uzak kuyruk tanımları ayarlarına gerek kalmadan birbirleriyle iletişim kurmalarını sağlayarak yapılandırma ve yönetimini basitleştirir.

Dağıtılmış yayınlama/abone olma ağınız için uygun bir topoloji seçmek üzere aşağıdaki geniş soruları göz önünde bulundurmanız gerekir:

- Ağınızdaki bağlantılar üzerinde ne kadar manuel denetime ihtiyacınız var?
- Ağınız ne kadar büyük olacak?
- Ne kadar dinamik olacak?
- Kullanılabilirlik ve ölçeklenebilirlik gereksinimleriniz nelerdir?

Yordam

- Ağınızdaki bağlantılar üzerinde ne kadar manuel denetime gereksinim duyacağınızı göz önünde bulundurun.

Yalnızca birkaç bağlantıya ihtiyacınız varsa ya da tek tek bağlantıların çok kesin olarak tanımlanması gerekiyorsa, büyük olasılıkla ağı el ile oluşturmanız gerekir.

Mantıksal olarak ilişkili birden çok kuyruk yöneticisine gereksinim duyarsanız ve verileri ve uygulamaları paylaşmanız gerekiyorsa, bunları bir kuyruk yöneticisi kümesinde birlikte gruplamayı düşünmeniz gerekir.

- Ağınızın ne kadar büyük olması gerektiğini tahmin edin.
 - a) Kaç kuyruk yöneticilerine ihtiyacınız olduğunu tahmin edin. Kuyrukların birden çok kuyruk yöneticisinde barındırılabilmesini unutmayın.
 - b) Bir küme kullanmayı düşünüyorsanız, tam havuz olarak işlev görecektir iki fazlıdan kuyruk yöneticisi ekleyin.

Daha büyük ağlar için, bağlantıların el ile yapılandırılması ve bakımı çok zaman alabilir ve bir küme kullanmayı düşünmeniz gerekir.

- Ağ etkinliğinin ne kadar dinamik olacağını göz önünde bulundurun.

Performansı yüksek kuyruk yöneticilerine barındırılacak meşgul kuyrukları planlayın.

Kuyrukların sık sık yaratılıp silinmesini bekliyorsanız, bir küme kullanmayı düşünün.

- Kullanılabilirlik ve ölçeklenebilirlik gereksinimlerinizi göz önünde bulundurun.
 - a) Kuyruk yöneticilerinin yüksek kullanılabilirliğini garanti edemediğinize karar verin. Bu durumda, bu gereksiminin kaç kuyruk yöneticisi için geçerli olduğunu tahmin edin.
 - b) Kuyruk yöneticilerinizden bazılarının diğerlerinden daha az yetenekli olup olmadığını göz önünde bulundurun.
 - c) Bazı kuyruk yöneticilerinize yönelik iletişim bağlantılarının diğerlerine göre daha kırılabilir olup olmadığını göz önünde bulundurun.
 - d) Birden çok kuyruk yöneticisine kuyruk barındırmayı göz önünde bulundurun.

El ile yapılandırılan ağlar ve kümelerin her ikisi de yüksek düzeyde kullanılabilir ve ölçeklenebilir olacak şekilde yapılandırılabilir. Bir küme kullanıyorsanız, iki fazlıdan kuyruk yöneticisini tam havuz olarak tanımlamanız gerekir. İki tam havuza sahip olmak, tam havuzlardan birinin kullanılmaması durumunda kümenin çalışmaya devam etmesini sağlar. Tam havuz kuyruğu yöneticilerinin güçlü, yüksek performanslı ve iyi ağ bağlantısına sahip olduğundan emin olun. Başka bir iş için tam havuz kuyruğu yöneticilerini kullanmayı planlamayın.

- Bu hesaplamalara dayalı olarak, kuyruk yöneticileri arasında bağlantıları el ile mi yapılandıracağınıza, yoksa bir kümeyi mi kullanacağınıza karar vermenize yardımcı olmak için sağlanan bağlantıları kullanın.

Sonraki adım

Artık dağıtılmış ileti sistemi ağınızı yapılandırmaya hazırsınız.

İlgili görevler

[Dağıtılmış kuyruğa alma yapılandırılıyor](#)

Dağıtılmış kuyruk yöneticisi ağlarının tasarlanması

IBM MQ , Kuyruk Yöneticilerini ve Kanallarını kullanarak uygulamalar arasında ve ağlar üzerinden veri gönderir ve alır. Ağ planlaması, bu sistemleri bir ağ üzerinden bağlamak için bir çerçeve oluşturmak üzere gereksinimlerin tanımlanmasını içerir.

Sisteminiz ile iletişim kurmanız gereken diğer sistemler arasında kanallar yaratılabilir. Doğrudan bağlantınız olmayan sistemlere bağlanmak için çok sekmeli kanallar yaratılabilir. Senaryolarda açıklanan ileti kanalı bağlantıları, Şekil 1 sayfa 21’inde bir ağ şeması olarak gösterilir.

Farklı fiziksel ağlardaki sistemler ya da bir güvenlik duvarı üzerinden iletişim kuran kanallar arasında kanallar oluşturmanız gerekiyorsa, IBM MQ Internet Pass-Thru komutunu kullanarak yapılandırmayı basitleştirebilirsiniz. Daha fazla bilgi için bkz. [IBM MQ Internet Pass-Thru](#).

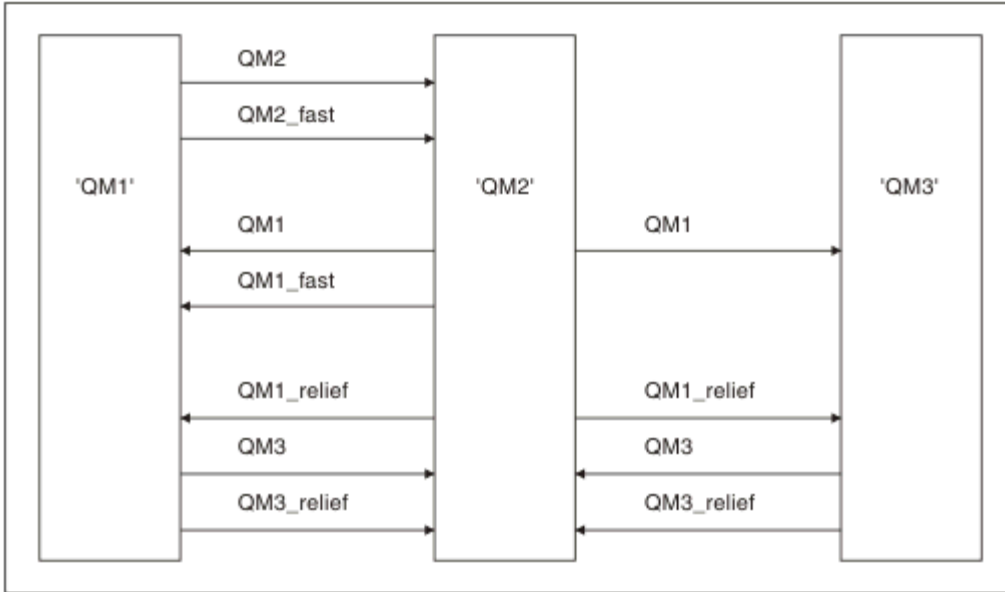
Kanal ve iletim kuyruğu adları

İletim kuyruklarına herhangi bir ad verilebilir. Ancak karışıklığı önlemek için, bunlara hedef kuyruk yöneticisi adlarıyla ya da kuyruk yöneticisi diğer adlarıyla aynı adları verebilirsiniz. Bu, iletim kuyruğunu kullandıkları rotayla ilişkilendirerek, ara (çok atlı) kuyruk yöneticileri aracılığıyla yaratılan koşut rotalara ilişkin net bir genel bakış sağlar.

Kanal isimleri için çok açık değil. Örneğin, QM2 için Şekil 1 sayfa 21 içindeki kanal adları, gelen ve giden kanallar için farklı olmalıdır. Tüm kanal adları iletim kuyruğu adlarını içerebilir, ancak bunların benzersiz olması için nitelenmiş olmaları gerekir.

Örneğin, QM2’de QM1’ den gelen bir QM3 kanalı ve QM3’ e giden bir QM3 kanalı vardır. Adları benzersiz kılmak için ilki QM3_from_QM1, ikincisi QM3_from_QM2 olarak adlandırılabilir. Bu şekilde, kanal adları, adın ilk bölümünde iletim kuyruğu adını gösterir. Adın ikinci bölümünde yön ve bitişik kuyruk yöneticisi adı gösterilir.

Şekil 1 sayfa 21 için önerilen kanal adları tablosu Çizelge 1 sayfa 22’inde verilmiştir.




Şekil 1. Tm kanalları gösteren ağ şeması

Çizelge 1. Kanal adları örneği

Rota adı	Kanal barındıran kuyruk yöneticileri	İletim kuyruğu adı	Önerilen kanal adı
QM1	QM1 & QM2	QM1 (QM2)	QM1.from.QM2
QM1	QM2 & QM3	QM1 (QM3)	QM1.from.QM3
QM1_fast	QM1 & QM2	QM1_fast (QM2)	QM1_fast.from.QM2
QM1_relief	QM1 & QM2	QM1_relief (QM2)	QM1_relief.from.QM2
QM1_relief	QM2 & QM3	QM1_relief (QM3)	QM1_relief.from.QM3
QM2	QM1 & QM2	QM2 (QM1)	QM2.from.QM1
QM2_fast	QM1 & QM2	QM2_fast (QM1)	QM2_fast.from.QM1
QM3	QM1 & QM2	QM3 (QM1)	QM3.from.QM1
QM3	QM2 & QM3	QM3 (QM2)	QM3.from.QM2
QM3_relief	QM1 & QM2	QM3_relief (QM1)	QM3_relief.from.QM1
QM3_relief	QM2 & QM3	QM3_relief (QM2)	QM3_relief.from.QM2

Not:

1.  IBM MQ for z/OS' da, kuyruk yöneticisi adları dört karakterle sınırlıdır.
2. Ağınızdaki tüm kanalları benzersiz bir şekilde adlandırın. Çizelge 1 sayfa 22 içinde gösterildiği gibi, kanal adında kaynak ve hedef kuyruk yöneticisi adları da dahil olmak üzere, bunu yapmak için iyi bir yoldur.

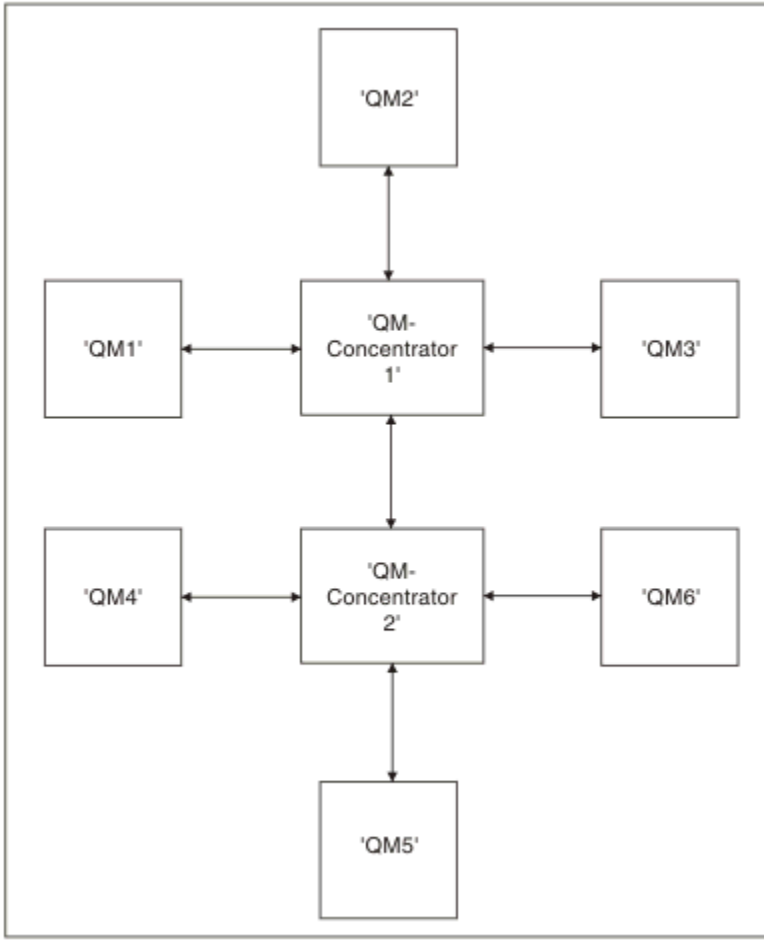
Ağ planlayıcısı

Bir ağ oluşturulması, planları ekibin diğer üyeleri tarafından uygulanan *ağ planlayıcısının* başka bir daha üst düzey işlevi olduğunu varsayar.

Yaygın olarak kullanılan uygulamalar için, Şekil 2 sayfa 23 içinde gösterildiği gibi, yerel erişim siteleri arasında geniş bant bağlantıları kullanarak, ileti trafiğinin yoğunluğu için yerel erişim siteleri açısından düşünmek daha ekonomiktir.

Bu örnekte iki ana sistem ve bir dizi uydu sistemi bulunmaktadır. Gerçek yapılandırma, işle ilgili önemli noktalara bağlıdır. Uygun merkezlerde bulunan iki yoğunlaştırıcı kuyruk yöneticisi vardır. Her QM-yoğunlaştırıcının yerel kuyruk yöneticilerine ileti kanalları vardır:

- QM-concentrator 1, üç yerel kuyruk yöneticisinin her biri için ileti kanallarına sahiptir: QM1, QM2ve QM3. Bu kuyruk yöneticilerini kullanan uygulamalar QM-yoğunlaştırıcılar aracılığıyla birbirleriyle iletişim kurabilir.
- QM-yoğunlaştırıcı 2, üç yerel kuyruk yöneticisine, QM4, QM5ve QM6ileti kanallarına sahiptir. Bu kuyruk yöneticilerini kullanan uygulamalar QM-yoğunlaştırıcılar aracılığıyla birbirleriyle iletişim kurabilir.
- QM-yoğunlaştırıcıların kendi aralarında ileti kanalları vardır, böylece kuyruk yöneticisindeki herhangi bir uygulamanın başka bir kuyruk yöneticisindeki herhangi bir uygulamayla ileti değiş tokuşu yapmasına izin verir.



Şekil 2. QM-yoğunlaştırıcıları gösteren ağ çizgesi

Kümelerin tasarlanması

Kümeler, hem ilk yapılandırmayı hem de devam eden yönetimi basitleştiren bir şekilde birbirine bağlanan kuyruk yöneticileri için bir mekanizma sağlar. Kümelerin doğru çalıştığından ve gerekli kullanılabilirlik ve yanıt verebilirlik düzeylerine ulaşabildiğinden emin olmak için kümeler dikkatli bir şekilde tasarlanmalıdır.

Başlamadan önce

Kümeleme kavramlarına giriş için aşağıdaki konulara bakın:

- [Dağıtılmış kuyruğa alma ve kümeler](#)
- [“Kümeleme ve dağıtılmış kuyruğa alma karşılaştırması” sayfa 29](#)
- [Kümenin bileşenleri](#)

Kuyruk yöneticisi kümenizi tasarlarken bazı kararlar almanız gerekir. Önce kümedeki hangi kuyruk yöneticilerinin küme bilgilerinin tam havuzlarını tutacağına karar vermeniz gerekir. Yaratmış olduğunuz herhangi bir kuyruk yöneticisi bir kümede çalışabilir. Bu amaçla istediğiniz sayıda kuyruk yöneticisi seçebilirsiniz, ancak ideal sayı ikidir. Tam havuzları tutmak üzere kuyruk yöneticilerini seçme hakkında bilgi için bkz. [“Tam havuzları tutmak için küme kuyruğu yöneticilerinin seçilmesi” sayfa 32.](#)

Kümenizi tasarlamaya ilişkin ek bilgi için aşağıdaki konulara bakın:

- [“Örnek kümeler” sayfa 38](#)
- [“Kümenin düzenlenmesi” sayfa 33](#)
- [“Küme adlandırma kuralları” sayfa 33](#)

- [z/OS](#) “Kuyruk paylaşımı grupları ve kümeleri” sayfa 34
- “Çakışan kümeler” sayfa 35

Sonraki adım

Kümeleri yapılandırma ve kümelerle çalışma hakkında daha fazla bilgi için aşağıdaki konulara bakın:

- [Kümeye iletişim kurulması](#)
- [Kuyruk yöneticisi kümesinin yapılandırılması](#)
- [İletileri kümelere/kümelerden yönlendirme](#)
- [İş yükü yönetimi için kümelerin kullanılması](#)

Kümenizi yapılandırmanıza yardımcı olacak daha fazla bilgi için bkz. “Kümeleme İpuçları” sayfa 36.

Birden çok küme iletim kuyruğunun nasıl kullanılmasının planlanması

İletim kuyruklarını belirttik olarak tanımlayabilir ya da sistemin iletim kuyruklarını sizin için oluşturmasını sağlayabilirsiniz. İletim kuyruklarını kendiniz tanımlarsanız, kuyruk tanımları üzerinde daha fazla denetime sahip oluyalınız kaldı? [z/OS](#) z/OS' ta, iletilerin tutulduğu sayfa kümesi üzerinde daha fazla denetime sahipsiniz.

İletim kuyruklarının tanımlanması

İletim kuyruklarını tanımlamak için iki y " ntemler vardır:

- Otomatik olarak, kuyruk yöneticisi özniteliği DEFCLXQ kullanılarak aşağıdaki gibi:

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

DEFCLXQ (SCTQ), tüm küme gönderen kanalları için varsayılan iletim kuyruğunun SYSTEM.CLUSTER.TRANSMIT.QUEUE. Bu varsayılan değerdir.

DEFCLXQ (KANAL), varsayılan olarak her bir küme gönderen kanalının SYSTEM.CLUSTER.TRANSMIT.kanal adı. Her iletim kuyruğu otomatik olarak kuyruk yöneticisi tarafından tanımlanır. Ek bilgi için bkz. “Otomatik olarak tanımlanan küme iletim kuyrukları” sayfa 25 .

- CLCHNAME özniteliği için belirlenen bir değerle iletim kuyruğu tanımlayarak, el ile. CLCHNAME özniteliği, iletim kuyruğunu hangi küme gönderen kanallarının kullanması gerektiğini belirtir.

[z/OS](#) z/OS üzerinde bir iletim kuyruğunu el ile tanımlıyorsanız, ek bilgi için bkz. “El ile tanımlanan küme iletim kuyruklarının planlanması” sayfa 27 .

Hangi güvenliğe ihtiyacım var?

Otomatik ya da el ile bir anahtar başlatmak için bir kanal başlatma yetkiniz olmalıdır.

İletim kuyruğu olarak kullanılan kuyruğu tanımlamak için, kuyruğu tanımlamak üzere standart IBM MQ yetkinizin olması gerekir.

Değişikliği uygulamak için uygun bir zaman ne zaman?

Kümeyi gönderen kanallar tarafından kullanılan iletim kuyruğunu değiştirirken, aşağıdaki noktaları dikkate alarak, güncellemenin gerçekleştirileceği bir süre ayırmanız gerekir:

- Bir kanalın iletim kuyruğunu değiştirmesi için gereken süre, eski iletim kuyruğundaki toplam ileti sayısına, kaç iletinin taşınması gerektiğine ve iletilerin büyüklüğüne bağlıdır.
- Uygulamalar, değişiklik gerçekleşirken iletileri iletim kuyruğuna yerleştirmeye devam edebilir. Bu, geçiş süresi içinde bir artışa neden olabilir.
- Tercihen, iş yükü düşük olduğunda, herhangi bir iletim kuyruğunun ya da DEFCLXQ parametresinin CLCHNAME parametresini istediğiniz zaman değiştirebilirsiniz.

Hemen bir şey olmadığını unutmayın.

- Değişiklikler yalnızca bir kanal başlatıldığında ya da yeniden başlatıldığında oluşur. Kanal başlatıldığında, yürürlükteki konfigürasyonu denetler ve gerekirse yeni bir iletim kuyruğuna geçer.
- Bir küme gönderen kanalının bir iletim kuyruğuyla ilişkisini değiştirebilecek birkaç değişiklik vardır:
 - CLCHNAME değerini değiştirerek, CLCHNAME değerini daha az özel ya da boş bırakın.
 - CLCHNAME özniteliğini değiştirerek CLCHNAME değerini daha özel hale getirin.
 - CLCHNAME belirlenmiş bir kuyruk siliniyor.
 - Kuyruk yöneticisi özniteliği DEFCLXQ değiştiriliyor.


-Anahtar ne kadar sürecek?

Geçiş süresi boyunca, kanala ilişkin iletiler bir iletim kuyruğundan diğerine taşınır. Bir kanalın iletim kuyruğunu değiştirmesi için gereken süre, eski iletim kuyruğundaki toplam ileti sayısına ve kaç iletinin taşınması gerektiğine bağlıdır.

Birkaç bin ileti içeren kuyruklar için, iletilerin taşınması bir saniyebakalım! Gerçek süre, iletilerin sayısına ve boyutuna bağlıdır. Kuyruk yöneticiniz iletileri her saniye birden çok megabayt olarak taşıyabilmelidir.

Uygulamalar, değişiklik gerçekleşirken iletileri iletim kuyruğuna yerleştirmeye devam edebilir. Bu, geçiş süresi içinde bir artışa neden olabilir.

Değişikliğin yürürlüğe girmesi için etkilenen her kümeyi gönderen kanal yeniden başlatılmalıdır. Bu nedenle, kuyruk yöneticisi meşgul olmadığında ve küme iletim kuyruklarında birkaç ileti saklandığında, iletim kuyruğu konfigürasyonunun değiştirilmesi en iyi yöntem olur.

runswch1 komutu  ya da z/OSüzerinde CSQUTIL ' de SWITCH CHANNEL (*) STATUS komutu , , küme gönderen kanallarının durumunu sorgulamak için kullanılabilir ve bekleyen değişiklikler iletim kuyruğu yapılandırmasında beklemede.

Değişiklik nasıl uygulanır?

Birden çok küme iletim kuyruğunda otomatik ya da el ile değişiklik yapmaya ilişkin ayrıntılar için [Sistemin birden çok küme iletim kuyrukları kullanarak uygulanması](#) başlıklı konuya bakın.

Değişiklik geri alınmıyor



Sorunlarla karşılaşırsanız değişiklikleri nasıl geri çevireceğinize ilişkin ayrıntılar için bkz. [z/OS ' da bir iletim kuyruğunda yapılan değişikliğin geri alınması](#) bir) .

Otomatik olarak tanımlanan küme iletim kuyrukları

İletim kuyruklarını sizin için sistemin oluşturmasını sağlayabilirsiniz.

Başlamadan önce



Küme iletim kuyruklarını z/OSüzerinde el ile ayarlamak için bkz. [“El ile tanımlanan küme iletim kuyruklarının planlanması” sayfa 27.](#)

Bu görev hakkında

Bir kanalla ilişkilendirilmiş el ile tanımlanmış bir küme iletim kuyruğu yoksa ve DEFCLXQ (KANAL) değerini belirlerseniz, kanal başlatıldığında kuyruk yöneticisi küme gönderen kanalı için otomatik olarak kalıcı dinamik bir kuyruk tanımlar. Model kuyruğu SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE , SYSTEM.CLUSTER.TRANSMIT.ChannelName.

Önemli: 

Kuyruk yöneticisi IBM MQ 8.0' e geçilirse, kuyruk yöneticisinde SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE.

Önce bu kuyruğu tanımlayın, ALTER QMGR DEFCLXQ (KANAL) komutunun yürürlüğe girmesi için.

Aşağıdaki JCL, model kuyruğunu tanımlamak için kullanabileceğiniz bir kod örneğidir:

```
//CLUSMODL JOB MSGCLASS=H,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=(MVCC)
//MQCMD EXEC PGM=CSQUTIL,REGION=4096K,PARM='CDLK'
//STEPLIB DD DISP=SHR,DSN=SCEN.MQ.V000.COM.BASE.SCSQAUTH
// DD DISP=SHR,DSN=SCEN.MQ.V000.COM.BASE.SCSQANLE
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(CMDINP)
/*
//CMDINP DD *
DEFINE QMODEL( 'SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE' ) +
QSGDISP( QMGR ) +

* COMMON QUEUE ATTRIBUTES
DESCR( 'SYSTEM CLUSTERING TRANSMISSION MODEL QUEUE' ) +
PUT( ENABLED ) +
DEFPRTY( 5 ) +
DEFFPSIST( YES ) +

* MODEL QUEUE ATTRIBUTES
DEFTYPE( PERMDYN ) +

* LOCAL QUEUE ATTRIBUTES
GET( ENABLED ) +
SHARE +
DEFSOPT( EXCL ) +
MSGDLVSQ( PRIORITY ) +
RETINTVL( 99999999 ) +
MAXDEPTH( 99999999 ) +
MAXMSGL( 4194304 ) +
NOHARDENBO +
BOTHRESH( 0 ) +
BOQNAME( ' ' ) +
STGCLASS( 'REMOTE' ) +
USAGE( XMITQ ) +
INDXTYPE( CORRELID ) +
CFSTRUCT( ' ' ) +
MONQ( OFF ) ACCTQ( OFF ) +

* EVENT CONTROL ATTRIBUTES
QDPMAXEV( ENABLED ) +
QDPHIEV( DISABLED ) +
QDEPTHHI( 80 ) +
QDPLOEV( DISABLED ) +
QDEPTHLO( 40 ) +
QSVCIIEV( NONE ) +
QSVCIINT( 99999999 ) +

* TRIGGER ATTRIBUTES
TRIGGER +
TRIGTYPE( FIRST ) +
TRIGMPRI( 0 ) +
TRIGDPH( 1 ) +
TRIGDATA( ' ' ) +
PROCESS( ' ' ) +
INITQ( ' ' )
/*
```

Yordam

1. DEFCLXQ kuyruk yöneticisi öznitelikliğini kullanın.

Bu öznitelikle ilgili ek bilgi için [ALTER QMGR](#) başlıklı konuya bakın.

İki seçenek vardır:

SCTQ

Bu seçenek varsayılan değerdir ve tek bir SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Kanal

Birden çok küme iletim kuyruğu kullandığınız anlamına gelir.


2. Yeni ilişkilendirmeye geçmek için:

- Kanalı durdurun ve yeniden başlatın.
- Kanal, yeni iletim kuyruğu tanımını kullanır.
- İletiler, eski kuyruktan yeni iletim kuyruğuna bir geçiş anahtarı işlemiyle aktarılır.

Tüm uygulama iletilerinin eski tanımlamaya konduğunu unutmayın.

Eski kuyruktaki ileti sayısı sıfıra ulaştığında, yeni iletiler doğrudan yeni iletim kuyruğuna yerleştirilir.

3. Geçiş işleminin tamamlanmasını izlemek için:

- a) Kanal tarafından başlatılan bir iletim kuyruğu anahtarı artalarda çalışır ve denetimciniz, tamamlandığı zamanı belirlemek için kuyruk yöneticisi iş günlüğünü izleyebilir.
- b) Anahtarın ilerleme durumunu göstermek için iş günlüğündeki iletileri izleyin.
- c) Yalnızca bu iletim kuyruğunu kullanmak istediğinizi kanalların kullandığından emin olmak için, DIS CLUSQMGR (*) komutunu verin; burada örneğin, iletim kuyruğunu tanımlayan iletim kuyruğu özelliği APPQMGR.CLUSTER1.XMITQ olur.
- d)  CSQUTIL altında SWITCH CHANNEL (*) STATUS komutunu kullanın.
Bu seçenek, bekleyen değişiklikleri ve iletim kuyrukları arasında kaç iletinin taşınması gerektiğini gösterir.

Sonuçlar

Küme iletim kuyruğunuzu ya da kuyruklarınızı ayarladınız.

İlgili görevler

[“El ile tanımlanan küme iletim kuyruklarının planlanması” sayfa 27](#)

IBM MQ for z/OS' ta, iletim kuyruklarını kendiniz tanımlarsanız, tanımlamalar ve iletilerin tutulduğu sayfa takımı üzerinde daha fazla denetime sahipsiniz.

İlgili başvurular

[ALTER QMGR](#)

[CLUSQMGR ' YI GÖRÜNTÜLE](#)



El ile tanımlanan küme iletim kuyruklarının planlanması

IBM MQ for z/OS' ta, iletim kuyruklarını kendiniz tanımlarsanız, tanımlamalar ve iletilerin tutulduğu sayfa takımı üzerinde daha fazla denetime sahipsiniz.

Başlamadan önce

Küme iletim kuyruklarını otomatik olarak ayarlamak için bkz. [“Otomatik olarak tanımlanan küme iletim kuyrukları” sayfa 25.](#)

Bu görev hakkında

Denetimciniz bir iletim kuyruğunu el ile tanımlar ve hangi küme gönderen kanalının ya da kanalların iletim kuyruğu olarak bu kuyruğu kullanacağını tanımlamak için CLCHNAME kuyruk özneteliğini kullanır.

CLCHNAME ' in birden çok kanal için tek bir kuyruğun kullanılmasına izin vermek için başında ya da sonunda bir genel arama karakteri içerebileceğine dikkat edin.

Yordam

1. Örneğin, aşağıdakileri girin:

```
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)
```

İpucu: İletim kuyruklarınız için kullanacağınız sayfa takımını (ve arabellek havuzunu) planlamanız gerekir. Farklı kuyruklar için farklı sayfa takımlarına sahip olabilir ve bunlar arasında yalıtım sağlayabilirsiniz; böylece, bir sayfanın doldurulması diğer sayfa kümelerindeki iletim kuyruklarını etkilemez.

Her bir kanalın uygun kuyruğu nasıl seçeceğine ilişkin bilgi için [Küme iletim kuyruklarıyla ve küme gönderen kanallarıyla çalışma](#) başlıklı konuya bakın.

Kanal başlatıldığında, ilişkilendirmesini yeni iletim kuyruğuna değiştirir. Hiçbir iletimin kaybolmadığından emin olmak için kuyruk yöneticisi, iletileri eski küme iletim kuyruğundan yeni iletim kuyruğuna otomatik olarak sırayla aktarır.

2. Yeni ilişkilendirmeye geçmek için CSQUTIL SWITCH işlevini kullanın.

Ek bilgi için [Küme gönderen kanallarıyla \(SWITCH\) ilişkili iletim kuyruğunun değiştirilmesi](#) başlıklı konuya bakın.

a) İletim kuyruğu değiştirilecek kanalı ya da kanalları DURDURULDU durumunda olacak şekilde durdurun.

Örneğin:

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

b) İletim kuyruğundaki CLCHNAME (XXXX) özniteliğini değiştirin.

c) İletileri değiştirmek ya da neler olduğunu izlemek için SWITCH işlevini kullanın.

Komutu kullan

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

İletileri kanalı başlatmadan taşımak için.

d) Kanalı ya da kanalları başlatın ve kanalın doğru kuyrukları kullanıp kullanmadığını denetleyin.

Örneğin:

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

İpucu: Aşağıdaki işlem CSQUTIL SWITCH işlevini kullanır. Daha fazla bilgi için bkz. [Küme gönderen kanallarıyla \(SWITCH\) ilişkili iletim kuyruğunun değiştirilmesi](#).

Bu işlevi kullanmanız gerekmez, ancak bu işlevin kullanılması daha fazla seçenek sağlar:

- SWITCH CHANNEL (*) STATUS yönteminin kullanılması, küme gönderen kanallarının geçiş durumunu tanımlamanın kolay bir yolunu sağlar. Yöneticinizin şu anda hangi kanalların değişmekte olduğunu görmesini ve bu kanallar bir sonraki başlatılışında yürürlüğe girecek bir anahtara sahip kanalları görmesini sağlar.

Bu yetenek olmadan, denetimcinizin birden çok DISPLAY komutu kullanması ve sonuç çıkışını bu bilgileri belirleyecek şekilde işlemesi gerekir. Yöneticiniz, bir yapılandırma değişikliğinin gerekli sonucu olduğunu da doğrulayabilir.

- Anahtarı başlatmak için CSQUTIL kullanılırsa, CSQUTIL bu işlemin ilerleyişini izlemeye devam eder ve yalnızca anahtar tamamlandığında sona erer.

Bu, bu işlemlerin toplu olarak gerçekleştirilmesini çok daha kolay hale getirir. Ayrıca, birden çok kanal arasında geçiş yapmak için CSQUTIL çalıştırılırsa, CSQUTIL bu işlemleri sıralı olarak gerçekleştirir; bunun işletmeniz için koşut olarak çalışan birden çok anahtardan daha az etkisi olabilir.

Sonuçlar

z/OSüzerinde küme iletim kuyruğunuzu ya da kuyruklarınızı ayarladınız.

Erişim denetimi ve birden çok küme iletim kuyruğu

Bir uygulama iletileri uzak küme kuyruklarına koyduğunda üç denetleme kipi arasından seçim yapın. Kipler, küme kuyruğuna göre uzaktan denetleniyor, SYSTEM . CLUSTER . TRANSMIT . QUEUEile yerel

olarak denetleniyor ya da küme kuyruğuna ya da küme kuyruğu yöneticisine ilişkin yerel tanımlarla karşılaştırılıyor.


IBM MQ , kullanıcının uzak bir kuyruğa ileti koyma izni olduğunu yerel olarak ya da yerel olarak ve uzaktan denetleme seçeneği sunar. Tipik bir IBM MQ uygulaması yalnızca yerel denetimi kullanır ve yerel kuyruk yöneticisinde yapılan erişim denetimlerine güvenen uzak kuyruk yöneticisine dayanır. Uzaktan denetim kullanılmazsa, ileti, uzak ileti kanalı işleminin yetkisiyle hedef kuyruğa yerleştirilir. Uzaktan denetimi kullanmak için, alan kanalın koyma yetkisini bağlam güvenliğine ayarlamamız gerekir.

Uygulamanın açtığı kuyruğa ilişkin yerel denetimler yapılır. Dağıtılmış kuyrukta, uygulama genellikle bir uzak kuyruk tanımlamasını açar ve uzak kuyruk tanımlamasına ilişkin erişim denetimleri yapılır. İleti tam yöneltme üstbilgisiyle yerleştirilirse, iletim kuyruğuna ilişkin denetimler yapılır. Bir uygulama yerel kuyruk yöneticisinde olmayan bir küme kuyruğunu açarsa, denetlenmesi gereken yerel nesne yoktur. Erişim denetimi denetimleri küme iletim kuyruğuna (SYSTEM . CLUSTER . TRANSMIT . QUEUE) göre yapılır. Birden çok küme iletim kuyruğunda bile, uzak küme kuyruklarına ilişkin yerel erişim denetimi SYSTEM . CLUSTER . TRANSMIT . QUEUE için yapılır.

Yerel ya da uzaktan denetim seçeneği, iki uç arasında bir seçimdir. Uzaktan kontrol etmek ince taneli. Her kullanıcının herhangi bir küme kuyruğuna yerleştirilebilmesi için kümedeki her kuyruk yöneticisinde bir erişim denetimi profili olmalıdır. Yerel olarak kontrol etmek kaba taneli. Her kullanıcının, bağlı olduğu kuyruk yöneticisindeki küme iletim kuyruğu için tek bir erişim denetimi tanımına gereksinimi vardır. Bu tanımla, herhangi bir kümedeki herhangi bir kuyruk yöneticisindeki herhangi bir küme kuyruğuna ileti yerleştirebilirler.

Yöneticilerin, küme kuyrukları için erişim denetimi oluşturmaları için başka bir yöntemi vardır. **setmqaut** komutunu kullanarak kümedeki herhangi bir kuyruk yöneticisinde bir küme kuyruğu için güvenlik profili yaratabilirsiniz. Tanıtım, uzak bir küme kuyruğunu yerel olarak açtığınızda, yalnızca kuyruk adını belirtirseniz etkili olur. Uzak kuyruk yöneticisi için bir tanım da ayarlayabilirsiniz. Bunu yaparsanız, kuyruk yöneticisi, küme kuyruğunu açan bir kullanıcının tanımını tam olarak nitelenmiş bir ad belirterek denetleyebilir.

Yeni tanımlar yalnızca kuyruk yöneticisi kısmı **ClusterQueueAccessControl** değerini RQMName olarak değiştirirseniz çalışır. Varsayılan değer Xm1tq' dir. Varolan uygulamaların küme kuyruklarını kullandığı tüm küme kuyrukları için tanım yaratmanız gerekir. Tanım yaratmadan RQMName olarak değiştirirseniz, uygulamalar başarısız olabilir.

İpucu: Küme kuyruğu erişim denetimi, uzak kuyruğa alma için geçerli değildir. Yerel tanımlara ilişkin erişim denetimleri yapılmaya devam eder. Bu değişiklikler, küme kuyruklarında ve küme konularında erişim denetimini yapılandırmak için aynı yaklaşımı izleyebileceğiniz anlamına gelir. 

Değişiklikler, küme kuyruklarına ilişkin erişim denetimi yaklaşımını da z/OS ile daha yakından hizalar. z/OS üzerinde erişim denetimi ayarlama komutları farklıdır, ancak her ikisi de erişimi nesnenin kendisine karşı değil, bir tanımla karşılaştırır.

İlgili kavramlar

“Kümeleme: Birden çok küme iletim kuyruğu kullanan uygulama yalıtımı” sayfa 46

Bir kümedeki kuyruk yöneticileri arasındaki ileti akışlarını yalıtabilirsiniz. Farklı küme gönderen kanalları tarafından iletilen iletileri farklı küme iletim kuyruklarına yerleştirebilirsiniz. Yaklaşımı tek bir kümede ya da çakışan kümelerle kullanabilirsiniz. Bu konuda, kullanmak üzere bir yaklaşım seçerken size yol gösterecek örnekler ve bazı en iyi uygulamalar sağlanmaktadır.

İlgili görevler

[Ayar ClusterQueueAccessControl](#)

Kümeleme ve dağıtılmış kuyruğa alma karşılaştırması

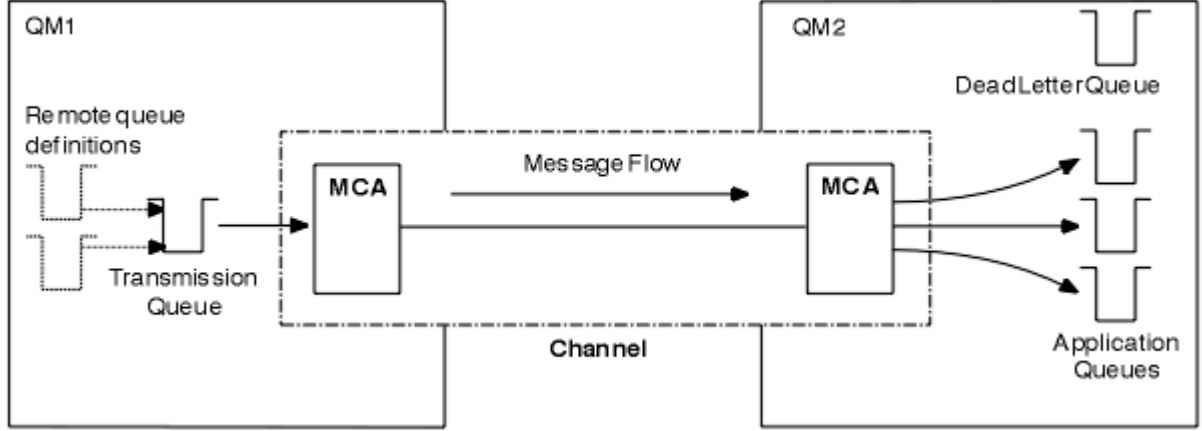
Dağıtımlı kuyruğa alma ve kümeleme kullanarak kuyruk yöneticilerine bağlanmak için tanımlanması gereken bileşenleri karşılaştırın.

Kümeleri kullanmıyorsanız, kuyruk yöneticileriniz bağımsızdır ve dağıtılmış kuyruğa alma özelliğini kullanarak iletişim kurar. Bir kuyruk yöneticisinin başka bir kuyruk yöneticisine ileti göndermesi gerekiyorsa, aşağıdakileri tanımlamanız gerekir:

- Bir iletim kuyruğu

- Uzak kuyruk yöneticisine bir kanal

Şekil 3 sayfa 30 , dağıtılmış kuyruğa alma için gerekli bileşenleri gösterir.



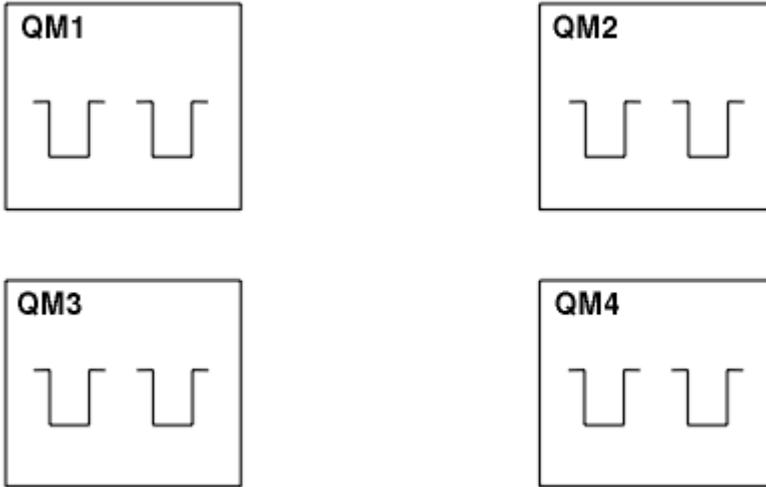
Şekil 3. Dağıtılmış kuyruğa alma

Bir kümedeki kuyruk yöneticilerini gruplarsanız, herhangi bir kuyruk yöneticisindeki kuyruklar kümedeki diğer herhangi bir kuyruk yöneticisi tarafından kullanılabilir. Herhangi bir kuyruk yöneticisi, aynı kümedeki diğer herhangi bir kuyruk yöneticisine belirtik tanımlamaları olmadan ileti gönderebilir. Her hedef için kanal tanımlarını, uzak kuyruk tanımlarını ya da iletim kuyruklarını belirtmezsiniz. Bir kümedeki her kuyruk yöneticisinin, kümedeki başka bir kuyruk yöneticisine ileti iletebileceği tek bir iletim kuyruğu vardır. Bir kümedeki her kuyruk yöneticisinin yalnızca aşağıdakileri tanımlaması gerekir:

- İletilerin alınacağı bir küme-alıcı kanalı
- Kendisini tanıtan ve küme hakkında bilgi alan bir küme-gönderen kanalı

Bir kümenin dağıtılmış kuyruğa alınmasına karşı ayarlanmasına ilişkin tanımlar

Her birinin iki kuyruğu olan dört kuyruk yöneticisini gösteren Şekil 4 sayfa 30' e bakın. Bu kuyruk yöneticilerine dağıtılmış kuyruğa alma yöntemini kullanarak bağlanmak için kaç tanımlama gerektiğini göz önünde bulundurun. Bir kümeyle aynı ağı kurmak için kaç tanımlama gerektiğini karşılaştırın.



Şekil 4. Dört kuyruk yöneticilerinden oluşan bir ağ

Dağıtılmış kuyruğa alma kullanılarak bir ağ ayarlanmasına ilişkin tanımlar

Şekil 3 sayfa 30 içinde gösterilen ağı dağıtılmış kuyruğa alma özelliğini kullanarak ayarlamak için aşağıdaki tanımlamalara sahip olabilirsiniz:

<i>Çizelge 2. Dağıtılmış kuyruğa alma tanımlamaları</i>		
Açıklama	Kuyruk yöneticisi başına sayı	Toplam sayı
Her diğer kuyruk yöneticisine ileti gönderilecek bir kanala ilişkin gönderen kanal tanımı	3	12
Diğer her kuyruk yöneticisinden ileti alınacağı bir kanala ilişkin alıcı kanal tanımlaması	3	12
Diğer her kuyruk yöneticisine bir iletim kuyruğuna ilişkin iletim kuyruğu tanımı	3	12
Her yerel kuyruk için yerel kuyruk tanımlaması	2	8
Bu kuyruk yöneticisinin iletileri koymak istediği her uzak kuyruk için bir uzak kuyruk tanımlaması	6	24

Soysal alıcı-kanal tanımlamalarını kullanarak bu tanımlama sayısını azaltabilirsiniz. Tanımlama sayısı üst sınırı, bu ağ için toplam 68 olan her kuyruk yöneticisinde 17 olabilir.

Kümeleri kullanarak bir ağ kurmak için tanımlar

Şekil 3 sayfa 30 içinde gösterilen ağ kümeleri kullanarak ayarlamak için aşağıdaki tanımlamalara gereksinim duyarsınız:

<i>Çizelge 3. Kümeleme tanımlamaları</i>		
Açıklama	Kuyruk yöneticisi başına sayı	Toplam sayı
Havuz kuyruğu yöneticisine ileti gönderilecek bir kanala ilişkin küme gönderen kanal tanımı	1	4
Kümedeki diğer kuyruk yöneticilerinden ileti alınacağı bir kanala ilişkin küme alıcı kanal tanımlaması	1	4
Her yerel kuyruk için yerel kuyruk tanımlaması	2	8

Bu kuyruk yöneticisi kümesini (iki tam havuzlu) ayarlamak için, her kuyruk yöneticisinde toplam on altı tanımlama olmak üzere dört tanımlama gerekir. Ayrıca, iki kuyruk yöneticisine ilişkin kuyruk yöneticisi tanımlamalarını, kümeye ilişkin tam havuz kuyruğu yöneticileri yapmak için değiştirmeniz gerekir.

Yalnızca bir CLUSSDR ve bir CLUSRCVR kanal tanımlaması gereklidir. Küme tanımlandığında, diğer kuyruk yöneticilerine herhangi bir kesinti olmadan kuyruk yöneticileri (havuz kuyruğu yöneticileri dışında) ekleyebilir ya da kaldırabilirsiniz.

Küme kullanılması, çok sayıda kuyruk yöneticisi içeren bir ağ kurmak için gereken tanımlama sayısını azaltır.

Daha az tanımlama ile hata riski daha azdır:

- Nesne adları her zaman eşleşiyor; örneğin, bir gönderen-alıcı çiftindeki kanal adı.
- Kanal tanımında belirlenen iletim kuyruğu adı, uzak kuyruk tanımında belirlenen doğru iletim kuyruğu tanımıyla ya da iletim kuyruğu adıyla her zaman eşleşir.
- QREMOTE tanımlaması her zaman uzak kuyruk yöneticisindeki doğru kuyruğu gösterir.

Küme ayarlandıktan sonra, başka bir kuyruk yöneticisinde sistem yönetimi işi yapmak zorunda kalmadan küme kuyruklarını bir kuyruk yöneticisinden diğerine taşıyabilirsiniz. Kanal, uzak kuyruk ya da iletim

kuyruğu tanımlarını silmeyi ya da değiştirmeyi unutmak mümkün değildir. Var olan ağda herhangi bir kesinti olmadan bir kümeye yeni kuyruk yöneticileri ekleyebilirsiniz.

Tam havuzları tutmak için küme kuyruğu yöneticilerinin seçilmesi

Her bir kümede, tam havuzları tutmak için en az bir ve tercihen iki kuyruk yöneticisi seçmeniz gerekir. İki tam havuz, en istisnai koşullar dışında herkes için yeterlidir. Mümkünse, çakışan kesintileri olmayan ve coğrafi olarak merkezi konumda olan, sağlam ve kalıcı bağlantılı platformlarda barındırılan kuyruk yöneticilerini seçin. Ayrıca, sistemleri tam havuz anasistemleri olarak ayırmayı ve diğer görevler için bu sistemleri kullanmamayı da düşünün.

Tam havuzlar , kümenin durumunun tam bir resmini sağlayan kuyruk yöneticileridir. Bu bilgileri paylaşmak için, her bir tam havuz CLUSSDR kanallarıyla (ve ilgili CLUSRCVR tanımlamalarıyla) kümedeki diğer tüm havuzlara bağlanır. Bu kanalları el ile tanımlamanız gerekir.



Şekil 5. Bağlı iki tam havuz.

Kümedeki diğer her kuyruk yöneticisi, *kısmi havuz* kümenin durumu hakkında şu anda bildiklerini gösteren bir resim sağlar. Bu kuyruk yöneticileri, kendilerine ilişkin bilgileri yayınlar ve diğer kuyruk yöneticilerine ilişkin bilgileri, kullanılabilir iki tam havuzu kullanarak isteyebilirler. Seçilen bir tam havuz yoksa, başka bir havuz kullanılır. Seçilen tam havuz yeniden kullanılabilir olduğunda, adım adım devam edecek şekilde diğerlerinden en son yeni ve değiştirilen bilgileri toplar. Tüm tam havuzlar hizmet dışı kalırsa, diğer kuyruk yöneticileri kısmi havuzlarındaki bilgileri kullanır. Ancak, bunlar sahip oldukları bilgileri kullanmakla sınırlıdır; yeni bilgiler ve güncelleme istekleri işlenemez. Tam havuzlar ağa yeniden bağlandığında, tüm havuzları (tam ve kısmi) güncel yapmak için iletiler değiş tokuş edilir.

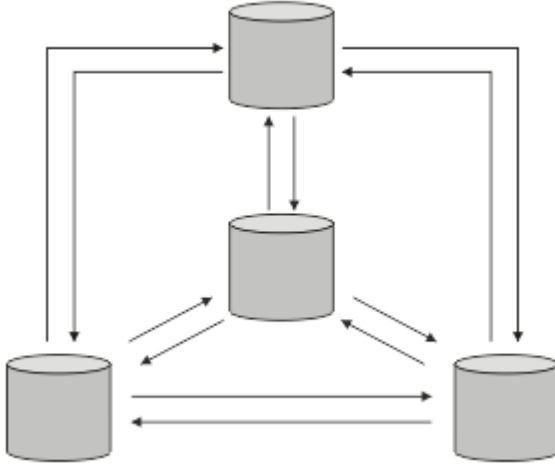
Tam havuzların ayrılmasını planlarken aşağıdaki noktaları göz önünde bulundurun:

- Tam havuzları tutmak için seçilen kuyruk yöneticilerinin güvenilir ve yönetilen olması gerekir. Güçlü ve kalıcı bağlantılı bir platformda barındırılan kuyruk yöneticilerini seçin.
- Tam havuzlarınızı barındıran sistemler için planlanan kesintileri göz önünde bulundurun ve bunların çakışan kesintilere sahip olmadığından emin olun.
- Ağ başarımını göz önünde bulundurun: Coğrafi olarak merkezi bir konumda olan ya da kümedeki diğer kuyruk yöneticileriyle aynı sistemi paylaşan kuyruk yöneticilerini seçin.
- Bir kuyruk yöneticisinin birden çok kümenin üyesi olup olmadığını göz önünde bulundurun. Bu avantajın, kuyruk yöneticisinin ne kadar meşgul olmasını beklediğiniz ile dengelenmesi koşuluyla, birkaç kümeye ilişkin tüm havuzları barındırmak için aynı kuyruk yöneticisini kullanmak yönetsel olarak uygun olabilir.
- Bazı sistemleri yalnızca tam havuzları içerecek şekilde ayırmayı ve bu sistemleri başka görevler için kullanmamayı düşünün. Bu, bu sistemlerin yalnızca kuyruk yöneticisi yapılandırması için bakım gerektirmesini ve diğer iş uygulamalarının bakımı için hizmetten kaldırılmamasını sağlar. Ayrıca, havuzu koruma görevinin sistem kaynakları için uygulamalarla rekabet etmemesini sağlar. Bu, tam havuzların küme durumunun korunmasında çok daha yüksek bir iş yüküne sahip olduğu büyük kümeler (örneğin, binden fazla kuyruk yöneticisinden oluşan kümeler) için özellikle yararlı olabilir.

İkide fazla tam havuz olması mümkündür, ancak nadiren önerilir. Nesne tanımlamaları (kuyruklar, konular ve kanallar) kullanılabilir tüm tam havuzlara aksa da, istekler yalnızca kısmi bir havuzdan en çok iki tam havuza akar. Bu, ikiden fazla tam havuz tanımlandığında ve herhangi iki tam havuz kullanılamaz olduğunda, bazı kısmi havuzların bekledikleri güncellemeleri almayabileceği anlamına gelir. Bkz. [MQ Cluster: Neden yalnızca iki Tam Havuz?](#)

İkisinden fazla tam havuz tanımlamayı yararlı bulabileceğiniz bir durum, var olan tam havuzları yeni donanım ya da yeni kuyruk yöneticilerine geçirmendir. Bu durumda, önceki tam havuzları kaldırmadan önce, yeni tam havuzları tanıtmanız ve bunların tam olarak doldurulduğunu doğrulamanız gerekir. Bir tam

havuz eklediğinizde, bunu CLUSSDR kanallarıyla diğer her tam havuza doğrudan bağlamanız gerektiğini unutmayın.



Şekil 6. İki'den birden fazla bağlı tam havuz

İlgili bilgiler

MQ Cluster ürünleri: Neden yalnızca iki Tam Havuz?

Bir MQ kümesi ne kadar büyük olabilir?

Kümenin düzenlenmesi

Bağlantı oluşturulacağı kuyruk yöneticilerini seçin. Performans etkisini, kuyruk yöneticisi sürümünü ve birden çok CLUSSDR kanalının istenip istenmediğini göz önünde bulundurun.

Kuyruk yöneticilerinin tam havuzları tutmasını seçtiğinizde, hangi kuyruk yöneticilerinin hangi tam havuza bağlanacağına karar vermeniz gerekir. CLUSSDR kanal tanımlaması, bir kuyruk yöneticisini kümedeki diğer tam havuzları öğrendiği tam bir havuza bağlar. Bundan sonra, kuyruk yöneticisi iletileri herhangi iki tam havuza gönderir. Her zaman önce CLUSSDR kanal tanımına sahip olduğu bir kanal kullanmayı dener. Bir kuyruk yöneticisini tam havuza bağlamayı seçebilirsiniz. Seçerken, yapılandırmanızın topolojisini ve kuyruk yöneticilerinin fiziksel ya da coğrafi konumunu göz önünde bulundurun.

Tüm küme bilgileri iki tam havuza gönderildiğinden, ikinci bir CLUSSDR kanal tanımlama yapmak istediğiniz durumlar olabilir. Çok sayıda tam havuzu geniş bir alana dağılmış bir kümede ikinci bir CLUSSDR kanalı tanımlayabilirsiniz. Daha sonra, bilgilerinizin hangi iki tam havuza gönderildiğini denetleyebilirsiniz.

Küme adlandırma kuralları

Kuyruk yöneticisinin ait olduğu kümeyi tanıtan bir adlandırma kuralı kullanarak aynı kümedeki kuyruk yöneticilerini adlandırmayı düşünün. Kanal adları için benzer bir adlandırma kuralı kullanın ve kanal özelliklerini açıklayacak şekilde genişletin.

MQ Cluster ürünlerini adlandırırken en iyi uygulamalar

Küme adları en çok 48 karakter olabilese de, diğer nesnelere adlandırma kuralları uygulanırken görece olarak kısa küme adları yararlı olur. Bkz. [“Küme kanalı adlarını seçerken en iyi uygulamalar”](#) sayfa 34.

Bir küme adı seçerken, genellikle 'içerik' yerine kümenin 'amacını' (uzun ömürlü olması muhtemeldir) temsil etmek yararlı olur. Örneğin, 'QM1_QM2_QM3_CLUS' yerine 'B2BPROD' veya 'ACTTEST'.

Küme Kuyruğu Yöneticisi adlarını seçerken en iyi uygulamalar

Yeni bir küme ve üyeleri sıfırdan oluşturuyorsanız, kuyruk yöneticileri için küme kullanımını yansıtan bir adlandırma kuralı düşünün. Her kuyruk yöneticisinin farklı bir adı olmalıdır. Ancak, bir kümedeki kuyruk yöneticilerine mantıksal gruplamaların (örneğin, 'ACTTQM1, ACTTQM2) tanımlanmasına ve anımsamasına yardımcı olması için benzer adlar verebilirsiniz.

Görece kısa kuyruk yöneticisi adları (örneğin, 8 karakterden kısa), kanal adları için sonraki bölümde açıklanan kuralı ya da benzeri bir şeyi kullanmayı seçerseniz yardımcı olur.

Küme kanalı adlarını seçerken en iyi uygulamalar

Kuyruk yöneticileri ve kümelerin adları en çok 48 karakterden oluşabildiğinden ve bir kanal adı 20 karakterle sınırlandırıldığından, bir projenin ortasında adlandırma kuralını değiştirmek zorunda kalmamak için ilk adlandırma nesnelere dikkatli olun (önceki bölüme bakın).

Kanalları tanımlarken, kümedeki herhangi bir kuyruk yöneticisinde otomatik olarak yaratılan küme gönderen kanallarının adlarını, kümede alıcı kuyruk yöneticisinde yapılandırılan karşılık gelen küme alıcı kanalından aldığı ve bu nedenle benzersiz ve anlamlı olmaları gerektiğini unutmayın. *Kümede uzak kuyruk yöneticileriyle ilgili olarak.*

Yaygın bir yaklaşım, küme adının önünde bulunan kuyruk yöneticisi adını kullanmaktır. Örneğin, küme adı CLUSTER1 ve kuyruk yöneticileri QM1, QM2 ise, küme alıcı kanalları CLUSTER1.QM1, CLUSTER1.QM2 olur.

Kanalların farklı öncelikleri varsa ya da farklı iletişim kuralları kullanıyorsa, bu kuralı genişletebilirsiniz. Örneğin:

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

Bu örnekte, S1 ilk SNA kanalı, N3 üç ağ önceliğine sahip NetBIOS kanalı ve T4 IPV4 ağı kullanan TCP/IP olabilir.

Paylaşılan kanal tanımlarını adlandırma

Tek bir kanal tanımı birden çok küme arasında paylaşılabilir; bu durumda, burada önerilen adlandırma kurallarının değiştirilmesi gerekir. Ancak, Kanal tanımlarının yönetilmesi konusunda açıklandığı gibi, her durumda her küme için ayrı kanallar tanımlanması tercih edilir.

Eski kanal adlandırma kuralları

Küme ortamlarının dışında geçmişte 'FROMQM.TO.TARGETQM' adlandırma kuralını kullanmak yaygındır, bu nedenle var olan kümelerin benzer bir şey kullandığını bulabilirsiniz (CLUSTER.TO.TARGET gibi). Yeni bir küme adlandırma şemasının parçası olarak bu önerilmez; kanal adınız içinde 'yararlı' bilgileri iletmek için kullanılacak karakterleri daha da azaltır.

z/OS IBM MQ for z/OS üzerindeki kanal adları

VTAM sosyal kaynaklarını ya da *Dinamik Etki Alanı Ad Sunucusu* (DDNS) sosyal adlarını tanımlayabilirsiniz. Sosyal adları kullanarak bağlantı adları tanımlayabilirsiniz. Ancak, bir küme-günlük nesnesi tanımlaması yarattığınızda, sosyal bir bağlantı adı kullanmayın.

Küme-günlük nesnesi tanımlamaları için sosyal bağlantı adlarının kullanılmasıyla ilgili sorun şöyledir: Sosyal CONNAME ile bir CLUSRCVR tanımlarsanız, CLUSSDR kanallarınızın amaçladığınız kuyruk yöneticilerini gösterdiğine ilişkin bir garanti yoktur. Başlangıçtaki CLUSSDR, kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisini gösteriyor olabilir, tam havuz bulunduran bir kuyruk yöneticisini göstermeyebilir. Bir kanal bağlantıyı yeniden denemeye başlarsa, aynı sosyal ada sahip farklı bir kuyruk yöneticisine yeniden bağlanarak ileti akışını kesintiye uğratabilir.

z/OS Kuyruk paylaşımı grupları ve kümeleri

Paylaşılan kuyruklar küme kuyrukları ve bir kuyruk paylaşım grubundaki kuyruk yöneticileri de küme kuyruğu yöneticileri olabilir.

IBM MQ for z/OS 'da kuyruk yöneticilerini kuyruk paylaşım grupları içinde gruplayabilirsiniz. Bir kuyruk paylaşım grubundaki bir kuyruk yöneticisi, en çok 32 kuyruk yöneticisi tarafından paylaşılacak bir yerel kuyruk tanımlayabilir.

Paylaşılan kuyruklar küme kuyrukları da olabilir. Ayrıca, bir kuyruk paylaşım grubundaki kuyruk yöneticileri de bir ya da daha fazla kümede olabilir.

VTAM soysal kaynaklarını ya da *Dinamik Etki Alanı Ad Sunucusu* (DDNS) soysal adlarını tanımlayabilirsiniz. Soysal adları kullanarak bağlantı adları tanımlayabilirsiniz. Ancak, bir küme-günlük nesnesi tanımlaması yarattığınızda, soysal bir bağlantı adı kullanmayın.

Küme-günlük nesnesi tanımlamaları için soysal bağlantı adlarının kullanılmasıyla ilgili sorun şöyledir: Soysal CONNAME ile bir CLUSRCVR tanımlarsanız, CLUSSDR kanallarınızın amaçladığınız kuyruk yöneticilerini gösterdiğine ilişkin bir garanti yoktur. Başlangıçtaki CLUSSDR , kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisini gösteriyor olabilir, tam havuz bulunduran bir kuyruk yöneticisini göstermeyebilir. Bir kanal bağlantıyı yeniden denemeye başlarsa, aynı soysal ada sahip farklı bir kuyruk yöneticisine yeniden bağlanarak ileti akışını kesintiye uğratabilir.

Grup dinleyici kapısını kullanan bir CLUSRCVR kanalı başlatılmıyor; böyle bir durumda, CLUSRCVR ' nin her seferinde hangi kuyruk yöneticisine bağlanacağı belirlenemez. Kümeyle ilgili bilgilerin paylaşılmadığı küme sistemi kuyrukları. Her kuyruk yöneticisinin kendine ait bir kuyruk yöneticisi vardır.

Küme kanalları yalnızca uygulama iletilerini aktarmak için değil, kümenin kurulumuna ilişkin iç sistem iletilerini aktarmak için de kullanılır. Kümedeki her kuyruk yöneticisinin bu iç sistem iletilerini kümelemenin doğru bir şekilde katılması için alması gerekir; bu nedenle, bu iletilerin alınacağı benzersiz CLUSRCVR kanalı gerekir.

Paylaşılan bir CLUSRCVR , kuyruk paylaşım grubundaki (QSG) herhangi bir kuyruk yöneticisinde başlayabilir ve bu nedenle QSG kuyruk yöneticilerine iç sistem iletilerinin tutarsız bir şekilde sağlanmasına neden olabilir; yani, kümeyle doğru bir şekilde katılamaz. Paylaşılan CLUSRCVR kanallarının kullanılmadığından emin olmak için, herhangi bir girişim CSQX502E iletilisiyle başarısız olur.

Çakışan kümeler

Çakışan kümeler, ek yönetim yetenekleri sağlar. Çakışan kümeleri yönetmek için gereken komut sayısını azaltmak için ad listesi kullanın.

Çakışan kümeler oluşturabilirsiniz. Çakışan kümeleri tanımlamanızın birçok nedeni vardır; örneğin:

- Farklı kuruluşların kendi yönetimine sahip olmasına izin vermek için.
- Bağımsız uygulamaların ayrı olarak yönetilmesine izin vermek için.
- Hizmet sınıfları yaratmak için.

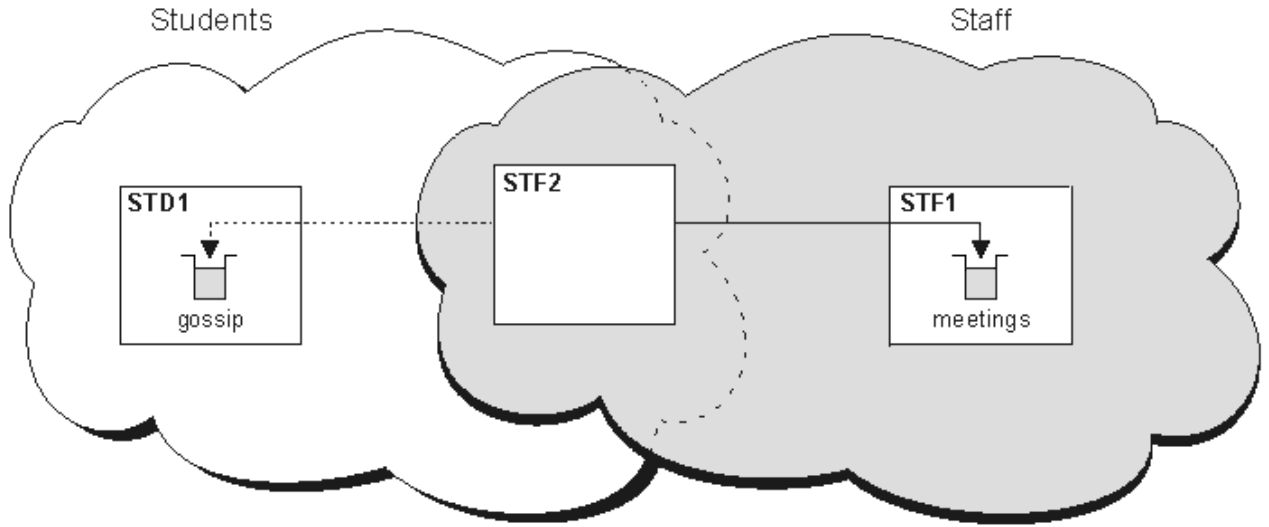
Şekil 7 sayfa 36 içinde, kuyruk yöneticisi STF2 her iki kümenin de bir üyesidir. Bir kuyruk yöneticisi birden çok kümenin üyesiyse, gereksinim duyduğunuz tanımlama sayısını azaltmak için ad listelerinden yararlanabilirsiniz. Ad listesi, küme adları gibi bir ad listesi içerir. Kümeleri adlandırmak için bir ad listesi oluşturabilirsiniz. Her iki küme için de tam havuz kuyruğu yöneticisi yapmak üzere STF2 için ALTER QMGR komutunda ad değerini belirtin.

Ağınızda birden çok küme varsa, bunlara farklı adlar vermeniz gerekir. Aynı ada sahip iki küme birleştirilirse, bunları yeniden ayırmak mümkün değildir. Kümeler ve kanallara farklı adlar vermek de iyi bir fikirdir. Bunlar, DISPLAY komutlarındaki çıkışa baktığınızda daha kolay ayırt edilir. Kuyruk yöneticisi adlarının doğru çalışması için bir küme içinde benzersiz olması gerekir.

Hizmet sınıflarının tanımlanması

Her bir personel ve öğrenci için kuyruk yöneticisi olan bir üniversite düşünün. Çalışanlar arasındaki mesajlar, yüksek öncelikli ve yüksek bant genişliğine sahip kanallarda seyahat etmek için. Öğrenciler arasındaki mesajlar daha ucuz ve daha yavaş kanallarda seyahat etmek için. Bu ağı, geleneksel dağıtılmış kuyruğa alma tekniklerini kullanarak kurabilirsiniz. IBM MQ , hedef kuyruk adını ve kuyruk yöneticisi adını inceleyerek hangi kanalların kullanılacağını seçer.

Personel ve öğrenciler arasında açıkça ayırım yapmak için, kuyruk yöneticilerini Şekil 7 sayfa 36 içinde gösterildiği gibi iki küme halinde gruplayabilirsiniz. IBM MQ , iletileri yalnızca o kümede tanımlanan kanallar üzerinden personel kümesindeki toplantı kuyruğuna taşır. Öğrenciler kümesindeki dedikodu kuyruğuna ilişkin iletiler, o kümede tanımlanan kanallar üzerinden gider ve uygun hizmet sınıfını alır.



Şekil 7. Hizmet sınıfları

Kümeleme İpuçları

Kümelemeyi kullanmadan önce sistemleriniz ya da uygulamalarınızda bazı değişiklikler yapmanız gerekebilir. Dağıtılmış kuyruğa alma davranışından hem benzerlikler hem de farklılıklar vardır.

- Küme kuyruklarına erişmeleri için bir kümenin dışındaki kuyruk yöneticilerine el ile yapılandırma tanımlamaları eklemelisiniz.
- İki kümeyi aynı adla birleştirirseniz, bunları yeniden ayıramazsınız. Bu nedenle, tüm kümelere benzersiz bir ad vermeniz önerilir.
- Bir ileti bir kuyruk yöneticisine ulaşır, ancak bu iletiyi alacak kuyruk yoksa, ileti gitmeyen ileti kuyruğuna konulur. Bir gitmeyen iletiler kuyruğu yoksa, kanal başarısız olur ve yeniden dener. Gönderilmeyen iletiler kuyruğunun kullanımı, dağıtılmış kuyruğa alma ile aynıdır.
- Kalıcı iletilerin bütünlüğü korunur. Kümelerin kullanılması sonucunda iletiler yinelenmez ya da kaybolmaz.
- Kümelerin kullanılması sistem yönetimini azaltır. Kümeler, dağıtılmış kuyruğa alma özelliğini kullanarak düşünebileceğinizden çok daha fazla kuyruk yöneticisiyle daha büyük ağların bağlanmasını kolaylaştırır. Bir kümedeki her kuyruk yöneticisi arasında iletişimi etkinleştirmeyi denerseniz, çok fazla ağ kaynağı kullanma riski vardır.
- Kuyruk yöneticilerini bir ağaç yapısında gösteren IBM MQ Explorer kullanırsanız, büyük kümelerin görünümü zahmetli olabilir.
- **Multi** Dağıtım listelerinin amacı, aynı iletiyi birden çok hedefe göndermek için tek bir MQPUT komutu kullanmaktır. Dağıtım listeleri IBM MQ for Multiplatforms üzerinde desteklenir. Kuyruk yöneticisi kümeleriyle dağıtım listelerini kullanabilirsiniz. Bir kümede, tüm iletiler MQPUT zamanında genişletilir. Bu avantaj, ağ trafiği açısından, kümelene memiş bir ortamda olduğu kadar büyük değildir. Dağıtım listelerinin yararı, çok sayıda kanal ve iletim kuyruğunun el ile tanımlanması gerekmemesidir.
- İş yükünüzü dengelemek için kümeleri kullanacaksanız uygulamalarınızı inceleyin. İletilerin belirli bir kuyruk yöneticisi tarafından mı, yoksa belirli bir sırada mı işlenmesini gerektirdiğini görün. Bu tür uygulamaların mesaj yakınlıkları olduğu söylenmektedir. Karmaşık kümelerde kullanabilmek için önce uygulamalarınızı değiştirmeniz gerekebilir.
- İletilerin belirli bir hedefe gönderilmesini zorlamak için MQOPEN üzerinde MQ00_BIND_ON_OPEN seçeneğini kullanmayı seçebilirsiniz. Hedef kuyruk yöneticisi kullanılamıyorsa, kuyruk yöneticisi yeniden kullanılabilir oluncaya kadar iletiler teslim edilmez. Yineleme riski nedeniyle iletiler başka bir kuyruk yöneticisine yöneltilmez.
- Bir kuyruk yöneticisi bir küme havuzunu barındırmak için anasistem adını ya da IP adresini bilmeniz gerekir. Kümeye katılan diğer kuyruk yöneticilerinde CLUSSDR tanımlamasını yaptığınızda CONNAME değiştirilmesinde bu bilgileri belirtmeniz gerekir. DHCP kullanıyorsanız, DHCP sistemi her yeniden

başlattığınızda yeni bir IP adresi ayırabileceği için IP adresi değiştirilebilir. Bu nedenle, CLUSSDR tanımlarında IP adresini belirtmemelisiniz. Tüm CLUSSDR tanımları IP adresi yerine anasistem adını belirtse de, tanımlar yine de güvenilir olmaz. DHCP, anasisteme ilişkin DNS dizin girişini yeni adresle güncellemez. Kuyruk yöneticilerini DHCP kullanan sistemlerde tam havuz olarak aday göstermeniz gerekiyorsa, DNS dizinini güncel tutacağını garanti eden yazılımı kurun.

- Kanallarınız için bağlantı adları olarak soysal adlar (VTAM soysal kaynakları ya da Dinamik Etki Alanı Adı Sunucusu (DDNS) soysal adları) kullanmayın. Bunu yaparsanız, kanallarınız beklenenden farklı bir kuyruk yöneticisine bağlanabilir.
- Bir iletiyi yalnızca yerel küme kuyruğundan alabilirsiniz, ancak bir kümedeki herhangi bir kuyruğa ileti yerleştirebilirsiniz. MQGET komutunu kullanmak için bir kuyruk açarsanız, kuyruk yöneticisi yerel kuyruğu açar.
- Basit bir IBM MQ kümesi oluşturduysanız, uygulamalarınızdan herhangi birini değiştirmenize gerek yoktur. Uygulama, MQOPEN çağrısında hedef kuyruğu adlayabilir ve kuyruk yöneticisinin yerini bilmesine gerek yoktur. İş yükü yönetimi için bir küme oluşturursanız, uygulamalarınızı gözden geçirmeniz ve gerektiğinde değiştirmeniz gerekir.
- DISPLAY CHSTATUS ve DISPLAY QSTATUS **runmqsc** komutlarını kullanarak bir kanala ya da kuyruğa ilişkin geçerli izleme ve durum verilerini görüntüleyebilirsiniz. İzleme bilgileri, sistemin performansını ve durumunu ölçmeye yardımcı olmak için kullanılabilir. İzleme, kuyruk yöneticisi, kuyruk ve kanal öznitelikleri tarafından denetlenir. MONACLS kuyruk yöneticisi özniteliğiyle otomatik olarak tanımlanan küme gönderen kanallarının izlenmesi mümkündür.

İlgili kavramlar

Kümeler

“Kümeleme ve dağıtılmış kuyruğa alma karşılaştırması” sayfa 29

Dağıtılmış kuyruğa alma ve kümeleme kullanarak kuyruk yöneticilerine bağlanmak için tanımlanması gereken bileşenleri karşılaştırın.

Bir kümenin bileşenleri

İlgili görevler

Kuyruk yöneticisi kümesinin yapılandırılması

Yeni bir küme ayarlama

Kuyruk yöneticisi havuzları ne kadar süre bilgi saklar?

Kuyruk yöneticisi havuzları bilgileri 30 gün boyunca saklar. Otomatik işlem, kullanılmakta olan bilgileri verimli bir şekilde yeniler.

Bir kuyruk yöneticisi kendisiyle ilgili bazı bilgiler gönderdiğinde, tam ve kısmi havuz kuyruğu yöneticileri bilgileri 30 gün boyunca saklar. Örneğin, bir kuyruk yöneticisi yeni bir kuyruk yaratılmasını duyurduğunda bilgi gönderilir. Bu bilgilerin süresinin dolmasını önlemek için kuyruk yöneticileri 27 gün sonra kendileriyle ilgili tüm bilgileri otomatik olarak yeniden gönderirler. Kısmi bir havuz, 30 günlük yaşam süresi boyunca yeni bir bilgi isteği gönderirse, süre bitimi orijinal 30 gün olarak kalır.

Bilgilerin süresi dolduğunda, bilgiler havuzdan hemen kaldırılmaz. Bunun yerine 60 günlük bir ek süre için tutuluyor. Ek süre içinde güncelleme alınmazsa, bilgiler kaldırılır. Ek süre, bir kuyruk yöneticisinin süre bitim tarihinde geçici olarak hizmet dışı kalmasına izin verir. Bir kuyruk yöneticisinin bir kümeyle bağlantısı 90 günden fazla kesilirse, kümenin bir parçası olmayı durdurur. Ancak, ağa yeniden bağlanırsa, kümenin bir parçası olur. Tam havuzlar, diğer kuyruk yöneticilerinden gelen yeni istekleri karşılamak için süresi dolmuş bilgileri kullanmaz.

Benzer şekilde, bir kuyruk yöneticisi tam havuzdan güncel bilgiler için bir istek gönderdiğinde, istek 30 gün sürer. 27 gün sonra IBM MQ isteği denetler. 27 gün boyunca başvurulduysa, otomatik olarak yenilenir. Yoksa, süresi sona ermek üzere bırakılır ve yeniden gerekirse kuyruk yöneticisi tarafından yenilenir. Süresi dolan istekler, atıl kuyruk yöneticilerinden bilgi alınmasını önler.

Not: APAR PH43191 için PTF '1' yi yüklemeniz ve kurmanız gerekir; bu PTF, aboneliğin süre bitimini hesaplarken sistem hatalarını düzeltir. Bu hatalar aboneliğin erken süre bitimine neden olabilir (CSQX456I iletilisinin yayınlanmasıyla sonuçlanabilir) ya da nesnenin süresi dolduktan sonra sona erebilir (bu, hatalı MQRC 2085 (MQRC_UNKNOWN_OBJECT) hatalarıyla sonuçlanır).

Büyük kümeler için, birçok kuyruk yöneticisi aynı anda kendileriyle ilgili tüm bilgileri otomatik olarak yeniden gönderirse bu yıkıcı olabilir. Bkz. [Büyük bir kümede yenilenmesi, kümenin performansını ve kullanılabilirliğini etkileyebilir.](#)

İlgili kavramlar

“Kümeleme: REFRESH CLUSTER en iyi uygulamalarını kullanma” sayfa 67

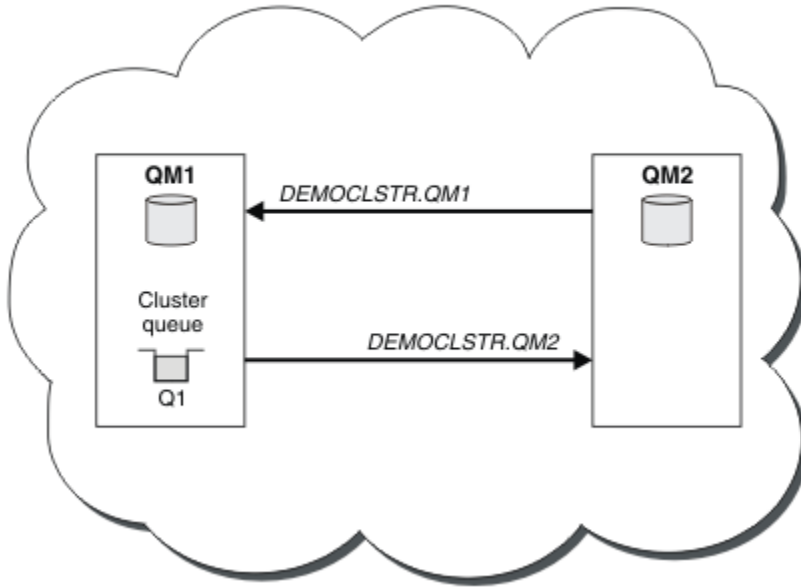
Bir kümeyle ilgili yerel olarak tutulan tüm bilgileri atmak ve kümedeki tüm havuzlardan bu bilgileri yeniden oluşturmak için **REFRESH CLUSTER** komutunu kullanabilirsiniz. Kural dışı durumlar dışında bu komutu kullanmanıza gerek yoktur. Bunu kullanmanız gerekiyorsa, nasıl kullanacağınıza ilişkin özel konular vardır. Bu bilgiler, müşterilerden gelen testlere ve geribildirime dayalı bir kılavuzdur.

Örnek kümeler

İlk örnek, iki kuyruk yöneticisinin olası en küçük kümesini gösterir. İkinci ve üçüncü örnekler, üç kuyruk yöneticisi kümesinin iki sürümünü gösterir.

Olası en küçük küme yalnızca iki kuyruk yöneticisi içerir. Bu durumda, her iki kuyruk yöneticisi de tam havuz içerir. Kümeyi ayarlamak için yalnızca birkaç tanımlama gerekir, ancak her kuyruk yöneticisinde yüksek düzeyde bir özerklik vardır.

DEMOCLSTR



Şekil 8. İki kuyruk yöneticisinden oluşan küçük bir küme

- Kuyruk yöneticileri, LONDON ve NEWYORK gibi uzun adlara sahip olabilir. z/OS IBM MQ for z/OS' da, kuyruk yöneticisi adları dört karakterle sınırlıdır.
- Her kuyruk yöneticisi genellikle ayrı bir makinede yapılandırılır. Ancak, aynı makinede birden çok kuyruk yöneticisi olabilir.

Benzer bir örnek küme ayarlamaya ilişkin yönergeler için [Yeni bir küme ayarlamabaşlıklı](#) konuya bakın.

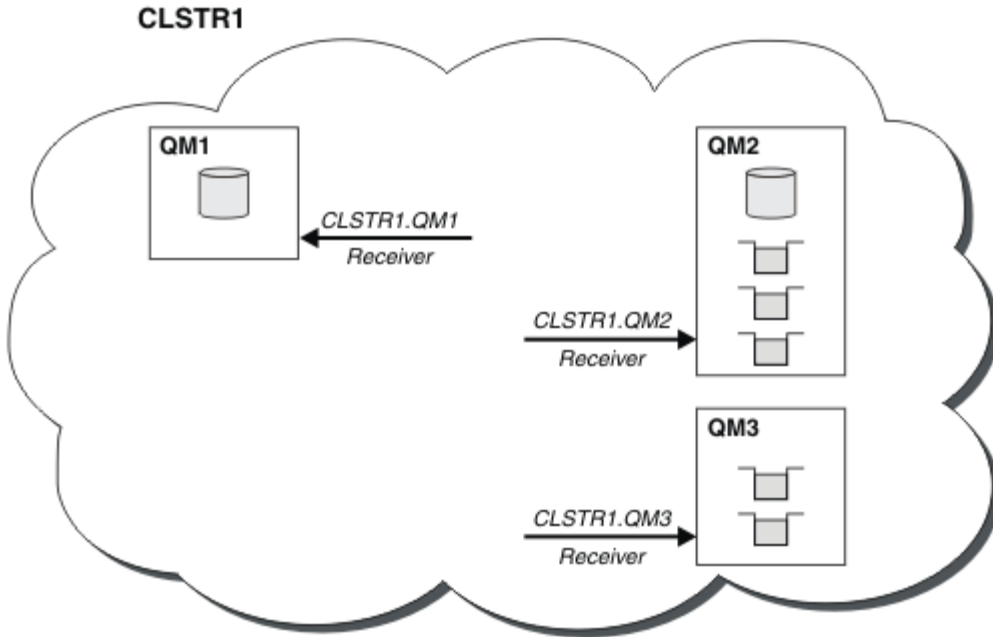
Şekil 9 sayfa 39 , CLSTR1adlı bir kümenin bileşenlerini gösterir.

- Bu kümede, QM1, QM2 ve QM3 olmak üzere üç kuyruk yöneticisi vardır.
- Kümedeki tüm kuyruk yöneticilerine ve kümeyle ilgili nesnelere ilişkin bilgilerin QM1 ve QM2 anasistem havuzları. Bunlara *tam havuz kuyruğu yöneticilerinden*ir. Havuzlar, çizgede gölgeli silindirelerle gösterilir.
- QM2 ve QM3 , kümedeki diğer herhangi bir kuyruk yöneticisinin erişebileceği bazı kuyrukları barındırır. Kümedeki diğer herhangi bir kuyruk yöneticisinin erişebileceği kuyruklara *küme kuyrukları* adı verilir. Küme kuyrukları, şemada gölgeli kuyruklar tarafından temsil edilir. Küme kuyruklarına kümenin

herhangi bir yerinden erişilebilir. IBM MQ kümeleme kodu, küme kuyruklarına ilişkin uzak kuyruk tanımlamalarının, bunlara gönderme yapan herhangi bir kuyruk yöneticisinde yaratılmasını sağlar.

Dağıtılmış kuyruğa alma sırasında olduğu gibi bir uygulama, kümedeki herhangi bir kuyruk yöneticisinde bir küme kuyruğuna ileti koymak için MQPUT çağrısı kullanır. Bir uygulama, yalnızca kuyruğun bulunduğu kuyruk yöneticisindeki bir küme kuyruğundan ileti almak için MQGET çağrısı kullanır.

- Her kuyruk yöneticisinin, ileti alabileceği *cluster_name*. *queue_manager_name* adlı bir kanalın alma sonu için el ile yaratılmış bir tanımlaması vardır. Alıcı kuyruk yöneticisinde, *cluster_name*. *queue_manager_name* bir küme-alıcı kanalıdır. Küme alıcı kanalı, dağıtılmış kuyruğa alma işlemi için kullanılan alıcı kanalına benzer; kuyruk yöneticisine ilişkin iletileri alır. Ayrıca, kümeyle ilgili bilgi de alır.

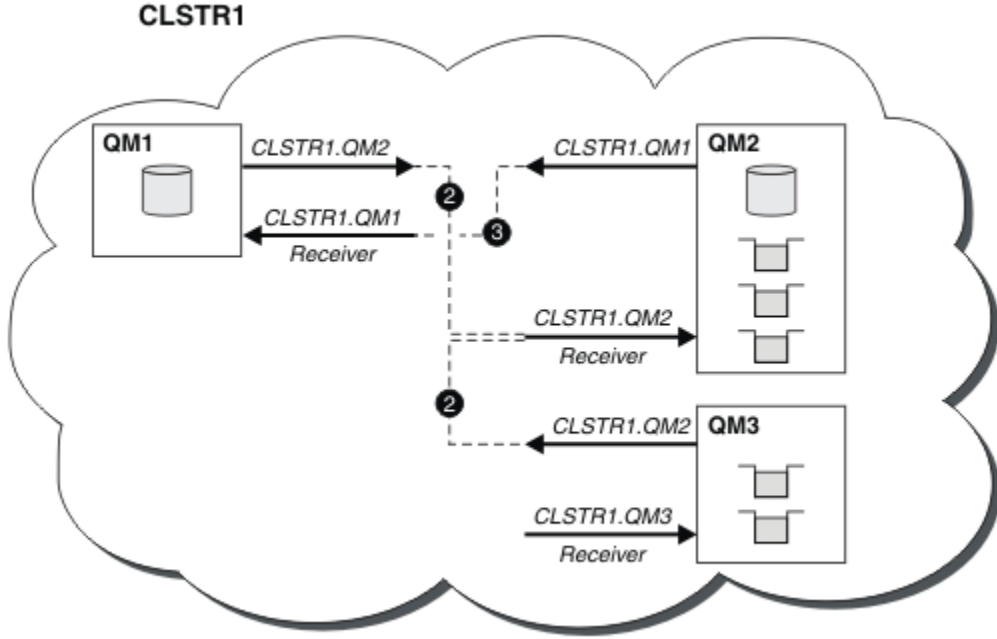


Şekil 9. Kuyruk yöneticileri kümesi

- Şekil 10 sayfa 40 ' da her kuyruk yöneticisi, bir kanalın gönderme sonuna ilişkin bir tanımlamaya da sahiptir. Tam havuz kuyruğu yöneticilerinden birinin küme alıcı kanalına bağlanır. Gönderen kuyruk yöneticisinde *cluster_name*. *queue_manager_name*, bir küme gönderen kanaldır. QM1 ve QM3, CLSTR1.QM2' e bağlanan küme gönderen kanallarına sahiptir, bkz. noktalı çizgi "2".

QM2, CLSTR1.QM1' e bağlanan bir küme gönderen kanalına sahiptir, bkz. noktalı çizgi "3". Küme gönderen kanal, dağıtılmış kuyruğa alma işlemi için kullanılan bir gönderen kanal gibidir; iletileri alan kuyruk yöneticisine gönderir. Ayrıca, kümeyle ilgili bilgi de gönderir.

Hem küme alıcı ucu hem de bir kanalın küme gönderen ucu tanımlandığında, kanal otomatik olarak başlar.

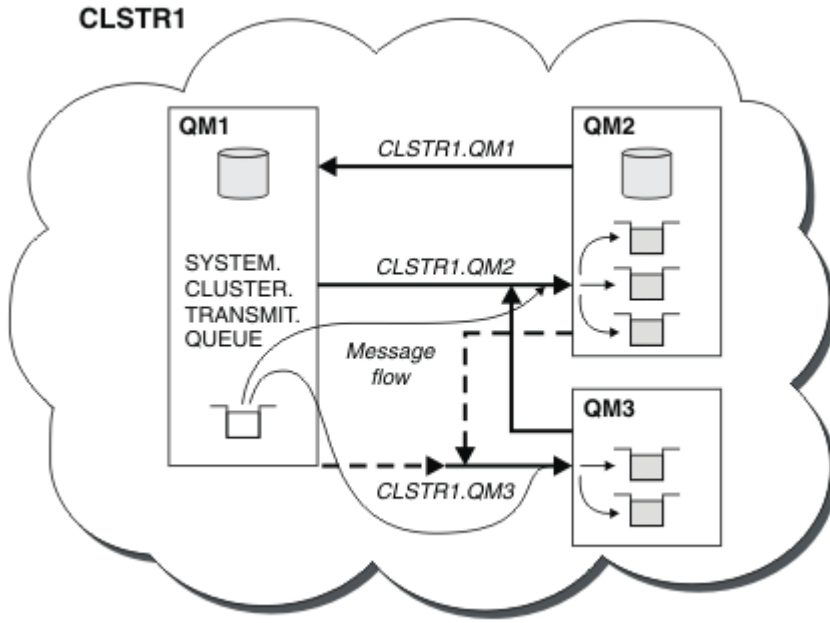


Şekil 10. Gönderen kanalları olan kuyruk yöneticileri kümesi

Yerel kuyruk yöneticisinde küme gönderen kanalı tanımlanması, kuyruk yöneticisini tam havuz kuyruğu yöneticilerinden birine tanıtır. Tam havuz kuyruk yöneticisi, tüm havuzundaki bilgileri buna göre günceller. Daha sonra otomatik olarak özgün kuyruk yöneticisine bir küme gönderen kanalı yaratır ve kümeyle ilgili kuyruk yöneticisi bilgilerini gönderir. Bu nedenle bir kuyruk yöneticisi bir kümeyi öğrenir ve bir küme bir kuyruk yöneticisini öğrenir.

Şekil 9 sayfa 39' e yeniden bakın. QM3 kuyruk yöneticisine bağlı bir uygulamanın QM2adresindeki kuyruklara bazı iletiler göndermek istediğini varsayın. QM3 ' in bu kuyruklara ilk kez erişmesi gerektiği zaman, bu kuyrukları tam bir havuza başvurarak keşfeder. Bu durumda tam havuz, gönderen kanalı CLSTR1.QM2kullanılarak erişilen QM2havuzudur. Havuzdaki bilgilerle, bu kuyruklar için otomatik olarak uzak tanımlamalar yaratabilir. Kuyruklar QM1üzerdeyse, QM2 tam bir havuz olduğundan bu mekanizma yine de çalışır. Tam havuz, kümedeki tüm nesnelere eksiksiz bir kaydını içerir. Bu ikinci durumda, QM3 otomatik olarak QM1üzerindeki küme alıcı kanalına karşılık gelen ve ikisi arasında doğrudan iletişime izin veren bir küme gönderen kanal oluşturacaktır.

Şekil 11 sayfa 41 içinde otomatik olarak oluşturulan iki küme gönderen kanalı ile aynı küme gösterilir. Küme gönderen kanallar, küme alıcı kanalı CLSTR1.QM3ile birleşen iki kesikli çizgiyle temsil edilir. Ayrıca, QM1 ' un iletilerini göndermek için kullandığı küme iletim kuyruğunu da (SYSTEM.CLUSTER.TRANSMIT.QUEUE) gösterir. Kümedeki tüm kuyruk yöneticilerinin, aynı kümedeki başka bir kuyruk yöneticisine ileti gönderebilecekleri bir küme iletim kuyruğu vardır.



Şekil 11. Otomatik tanımlı kanalları gösteren bir kuyruk yöneticileri kümesi

Not: Diğer çizgeler, yalnızca el ile tanımlama yaptığınız kanalların alma uçlarını gösterir. Gönderme uçları çoğunlukla gerektiğinde otomatik olarak tanımlandığından atlanır. Çoğu küme gönderen kanalının otomatik tanımlanması, kümelerin işlevi ve verimliliği için çok önemlidir.

İlgili kavramlar

[“Kümeleme ve dağıtılmış kuyruğa alma karşılaştırması” sayfa 29](#)

Dağıtılmış kuyruğa alma ve kümeleme kullanarak kuyruk yöneticilerine bağlanmak için tanımlanması gereken bileşenleri karşılaştırın.

[Bir kümenin bileşenleri](#)

İlgili görevler

[Kuyruk yöneticisi kümesinin yapılandırılması](#)

[Yeni bir küme ayarlama](#)

Kümeleme: En iyi uygulamalar

Kümeleme, birbirine bağlanan kuyruk yöneticileri için bir mekanizma sağlar. Bu bölümde açıklanan en iyi uygulamalar, müşterilerin testlerine ve geribildirimlerine dayalıdır.

Başarılı bir küme kurulumu, iyi planlamaya ve iyi uygulama yönetimi ve ağ tasarımı gibi IBM MQ temellerinin kapsamlı bir şekilde anlaşılmasına bağlıdır. Devam etmeden önce, ilgili konulardaki bilgileri bildiğinizden emin olun.

İlgili kavramlar

[Dağıtılmış kuyruğa alma ve kümeler](#)

[Kümeler](#)

İlgili görevler

[“Kümelerin tasarlanması” sayfa 23](#)

Kümeler, hem ilk yapılandırmayı hem de devam eden yönetimi basitleştiren bir şekilde birbirine bağlanan kuyruk yöneticileri için bir mekanizma sağlar. Kümelerin doğru çalıştığından ve gerekli kullanılabilirlik ve yanıt verebilirlik düzeylerine ulaşabildiğinden emin olmak için kümeler dikkatli bir şekilde tasarlanmalıdır.

[Kümeleri izleme](#)

Kümeleme: çakışan kümeler için dikkat edilmesi gereken özel noktalar

Bu konuda, IBM MQ kümelerini planlamaya ve yönetmeye ilişkin yönergeler sağlanır. Bu bilgiler, müşterilerden gelen testlere ve geribildirimlere dayalı bir kılavuздur.

Küme sahipliği

Aşağıdaki bilgileri okumadan önce çakışan kümeleri tanıyın. Gerekli bilgiler için [“Çakışan kümeler” sayfa 35 ve Kümeler arasında ileti yollarını yapılandırma başlıklı konuya](#) bakın.

Çakışan kümelerden oluşan bir sistemi yapılandırırken ve yönetirken, en iyisi aşağıdakilere uymaktır:

- IBM MQ kümeleri daha önce açıklandığı gibi 'gevşek bir şekilde eşlenmiş' olsalar da, bir kümeyi tek bir yönetim birimi olarak kabul etmek yararlıdır. Bu kavram, tek tek kuyruk yöneticilerindeki tanımlamalar arasındaki etkileşim, kümenin düzgün çalışması için kritik önem taşıdığından kullanılır. Örneğin: İş yükü dengeli küme kuyrukları kullanılırken, tek bir yöneticinin ya da ekibin, küme genelinde dağıtılan tanımlamalara bağlı olarak iletilere ilişkin olası hedeflerin tam kümesini anlaması önemlidir. Daha önemsiz olarak, küme gönderen/alıcı kanal çiftleri boyunca uyumlu olmalıdır.
- Bu önceki kavram göz önünde bulundurulur; birden çok kümenin (ayrı ekipler/kişiler tarafından yönetileceği) bir araya geldiği durumlarda, ağ geçidi kuyruk yöneticilerinin yerinde denetim yönetiminde net ilkelere sahip olmak önemlidir.
- Çakışan kümeleri tek bir ad alanı olarak kabul etmek yararlıdır: Kanal adları ve kuyruk yöneticisi adları tek bir küme boyunca benzersiz olmalıdır. Yönetim, topolojinin tamamında benzersiz olduğunda çok daha kolaydır. Uygun bir adlandırma kuralına uymanız en iyi yoldur; olası kurallar [“Küme adlandırma kuralları” sayfa 33](#) içinde açıklanmıştır.
- Bazen idari ve sistem yönetimi işbirliği esastır. Örneğin, çakışması gereken farklı kümelere sahip kuruluşlar arasındaki işbirliği. Kimin neye sahip olduğuna ilişkin net bir anlayış ve uygulanabilir kurallar ve kurallar, kümelerle örtüşme sırasında kümelemenin sorunsuz çalışmasına yardımcı olur.

Çakışan kümeler: Ağ Geçitleri

Genel olarak, tek bir kümeyi yönetmek birden çok kümeye göre daha kolaydır. Bu nedenle çok sayıda küçük küme oluşturmak (örneğin her uygulama için bir tane) genel olarak kaçınılması gereken bir şeydir.

Ancak, hizmet sınıfları sağlamak için çakışan kümeler uygulayabilirsiniz. Örneğin:

- Daha küçük olanın Yayınlama/Abone Olma için olduğu eş merkezli kümeleriniz varsa. Ek bilgi için [Sistemleri nasıl boyutlandırılır](#) başlıklı konuya bakın.
- Bazı kuyruk yöneticileri farklı ekipler tarafından yönetiliyorsa. Daha fazla bilgi için önceki [“Küme sahipliği” sayfa 42](#) bölümüne bakın.
- Bir organizasyonel ya da coğrafi bakış açısından mantıklı ise.
- Eşdeğer kümeler ad çözümlemesi ile çalışıyorsa, örneğin var olan bir kümede TLS uygulanıyor.

Üst üste binen kümelerden güvenlik avantajı yoktur; iki farklı ekip tarafından yönetilen kümelerin örtüşmesine, ekiplerin yanı sıra topolojiyi de etkin bir şekilde birleştirmesine izin verir:

- Böyle bir kümede ilan edilen herhangi bir ada diğer küme tarafından erişilebilir.
- Bir kümede ilan edilen herhangi bir ad, uygun mesajları çekmek için diğerinde duyurulabilir.
- Ağ geçidinin bitişiğindeki bir kuyruk yöneticisinde duyurulmayan herhangi bir nesne, ağ geçidinin üyesi olduğu herhangi bir kümeden çözümlenebilir.

Ad alanı, her iki kümenin birleşmesi ve tek bir ad alanı olarak ele alınmalıdır. Bu nedenle, çakışan bir kümenin sahipliği her iki kümenin tüm yöneticileri arasında paylaşılır.

Bir sistem birden çok küme içerdiğinde, iletileri bir kümedeki kuyruk yöneticilerinden başka bir kümedeki kuyruk yöneticilerine yönlendirme gereksinimi olabilir. Bu durumda, birden çok kümenin bir şekilde birbirine bağlanması gerekir: İzlenmesi gereken iyi bir kalıp, kümeler arasında ağ geçidi kuyruk yöneticilerinin kullanılmasıdır. Bu düzenleme, noktadan noktaya iletişim kanallarının yönetilmesi zor bir ağ oluşturulmasını önler ve güvenlik politikaları gibi sorunları yönetmek için iyi bir yer sağlar. Bu düzenlemeyi gerçekleştirmenin iki farklı yolu vardır:

1. İkinci bir küme alıcı tanımlaması kullanarak her iki kümeye bir (ya da daha çok) kuyruk yöneticisi yerleştirin. Bu düzenleme daha az yönetim tanımı içerir, ancak daha önce belirtildiği gibi, çakışan bir kümenin sahipliğinin her iki kümenin tüm yöneticileri arasında paylaşıldığı anlamına gelir.
2. Küme 1 'deki bir kuyruk yöneticisini, geleneksel noktadan noktaya iletişim kanallarını kullanarak küme 2 'deki bir kuyruk yöneticisiyle eşleştirin.

Bu iki durumda da, trafiği uygun şekilde yönlendirmek için çeşitli araçlar kullanılabilir. Özellikle, kuyruk ya da kuyruk yöneticisi diğer adları diğer kümeye yönelmek için kullanılabilir ve boş **RQMNAME** özelliğine sahip bir kuyruk yöneticisi diğer adı, iş yükü dengelemesini istendiği yerde yeniden yönlendirir.

İlgili kavramlar

[“Küme adlandırma kuralları” sayfa 33](#)

Kuyruk yöneticisinin ait olduğu kümeyi tanıtan bir adlandırma kuralı kullanarak aynı kümedeki kuyruk yöneticilerini adlandırmayı düşünün. Kanal adları için benzer bir adlandırma kuralı kullanın ve kanal özelliklerini açıklayacak şekilde genişletin.

Kümeleme: Topoloji tasarımıyla ilgili dikkat edilecek noktalar

Bu konuda, IBM MQ kümelerini planlamaya ve yönetmeye ilişkin yönergeler sağlanır. Bu bilgiler, müşterilerden gelen testlere ve geribildirime dayalı bir kılavuzdur.

Kullanıcı uygulamalarının ve iç idari süreçlerin önceden nerede bulunacağı düşünülüp, birçok sorun önlenebilir veya daha sonraki bir tarihte en aza indirilebilir. Bu konu, performansı artırabilen tasarım kararlarıyla ilgili bilgileri içerir ve küme ölçeklendirdikçe bakım görevlerini basitleştirir.

- [“Kümeleme altyapısının performansı” sayfa 43](#)
- [“Tam havuzlar” sayfa 44](#)
- [“Uygulamalar tam havuzlardaki kuyrukları kullanmalı mı?” sayfa 45](#)
- [“Kanal tanımlarının yönetilmesi” sayfa 45](#)
- [“Birden çok kanal üzerinden iş yükü dengeleme” sayfa 45](#)

Kümeleme altyapısının performansı

Bir uygulama kümedeki bir kuyruk yöneticisinde bir kuyruk açmaya çalışıldığında, kuyruk yöneticisi, kuyruğun kümede nerede bulunduğunu öğrenmesi için ilgili kuyruğun tüm havuzlarına ilgisini kaydeder. Kuyruk konumu ya da yapılandırmasında yapılan güncellemeler, tüm havuzlar tarafından otomatik olarak ilgili kuyruk yöneticisine gönderilir. İlginizi çeken bu kayıt, dahili olarak abonelik olarak bilinir (bu abonelikler, IBM MQ içinde yayınlama/abone olma ileti sistemi için kullanılan IBM MQ abonelikleriyle aynı değildir)

Bir kümeyle ilgili tüm bilgiler her tam havuzdan geçer. Bu nedenle, tam havuzlar her zaman yönetimle ilgili ileti trafiği için bir kümede kullanılır. Bu abonelikleri yönetirken sistem kaynaklarının yüksek kullanımı ve bunların iletimi ve sonuçta ortaya çıkan yapılandırma iletileri, kümeleme altyapısında önemli bir yüke neden olabilir. Bu yükün anlaşılmasını ve mümkün olan yerlerde en aza indirilmesini sağlarken göz önünde bulundurulması gereken birçok şey vardır:

- Küme kuyruğunu kullanan tek tek kuyruk yöneticileri ne kadar çok olursa, sistemde o kadar çok abonelik olur ve bu nedenle değişiklikler oluştuğunda ve ilgili abonelere, özellikle de tam havuz kuyruğu yöneticilerine bildirilmesi gereken yönetim ek yükü o kadar büyük olur. Gereksiz trafiği ve tam havuz yükünü en aza indirmenin bir yolu, benzer uygulamaları (yani aynı kuyruklarla çalışan uygulamaları) daha az sayıda kuyruk yöneticilerine bağlamaktır.
- Sistemdeki abonelik sayısının başarımı etkilemesine ek olarak, kümelenmiş nesnelerin konfigürasyonundaki değişiklik hızı başarımı etkileyebilir; örneğin, kümelenmiş bir kuyruk konfigürasyonunun sık sık değiştirilmesi gibi.
- Bir kuyruk yöneticisi birden çok kümenin üyesiye (yani, çakışan bir küme sisteminin parçasıysa), aynı kuyruk yöneticileri birden çok kümenin tam havuzları olsa bile, bir kuyrukta yapılan herhangi bir ilgi, üyesi olduğu her küme için bir abonelikle sonuçlanır. Bu düzenleme, sistemdeki yükü artırır ve tek bir küme yerine birden çok çakışan kümenin gerekli olup olmadığını düşünmek için bir nedendir.

- Uygulama iletisi trafiđi (yani, IBM MQ uygulamaları tarafından küme kuyruklarına gönderilen iletler), hedef kuyruk yöneticilerine ulaşmak için tam havuzlar üzerinden gitmez. Bu ileti trafiđi, doğrudan iletinin kümeye girdiđi kuyruk yöneticisi ile küme kuyruđunun bulunduğu kuyruk yöneticisi arasında gönderilir. Bu nedenle, tam havuz kuyruđu yöneticileri söz konusu iki kuyruk yöneticisinden biri olmadıđı sürece, tam havuz kuyruđu yöneticilerine göre uygulama ileti trafiđinin yüksek oranlarına uyulması gerekli deđildir. Bu nedenle, kümeleme altyapısı yükünün önemli olduđu kümelerdeki uygulama iletisi trafiđi için tam havuz kuyruđu yöneticilerinin kullanılmaması önerilir.

Tam havuzlar

Havuz, bir kümenin üyesi olan kuyruk yöneticilerine ilişkin bilgilerin bir toplamıdır. Kümedeki her kuyruk yöneticisine ilişkin eksiksiz bir bilgi kümesini barındıran bir kuyruk yöneticisi tam bir havuza sahiptir. Tam havuzlar ve kısmi havuzlar hakkında daha fazla bilgi için bkz. [Küme havuzu](#).

Güvenilir ve yüksek düzeyde kullanılabilir olan sunucularda tam havuzlar tutulmalıdır ve tek hata noktalarından kaçınılmalıdır. Küme tasarımının her zaman iki tam havuzu olmalıdır. Tam havuzda bir hata oluşursa, küme çalışmaya devam edebilir.

Kümedeki bir kuyruk yöneticisi tarafından yapılan küme kaynaklarında yapılan güncellemelerin ayrıntıları; örneğin, kümelene miş kuyruklar, o kümedeki en çok iki tam havuza (ya da kümede yalnızca bir tam havuz kuyruk yöneticisi varsa) gönderilir. Bu tam havuzlar, bilgileri tutar ve kümede ilgili olan herhangi bir kuyruk yöneticisine geçirir (yani, ona abone olurlar). Kümenin her üyesinin küme kaynaklarının güncel bir görünümüne sahip olmasını sağlamak için, her kuyruk yöneticisinin bir kerede en az bir tam havuz kuyruk yöneticisiyle iletişim kurabilmesi gerekir.

Herhangi bir nedenle, bir kuyruk yöneticisi herhangi bir tam havuzla iletişim kuramıyorsa, bir dönem için önceden önbelleđe alınmış bilgi düzeyine dayalı olarak kümede çalışmaya devam edebilir, ancak daha önce kullanılmamış küme kaynaklarına yeni güncelleme ya da erişim sağlanmaz.

Bu nedenle, iki tam havuzu her zaman kullanılabilir tutmayı hedeflemelisiniz. Bununla birlikte, bu düzenleme, kümenin tam bir havuz olmadan kısa bir süre için yeterli şekilde işlev göstermesi nedeniyle aşırı önlemlerin alınması gerektiđi anlamına gelmez.

Bir kümenin, küme bilgilerinin kullanılabilirliđi dışında iki tam havuz kuyruđu yöneticisinin olması gerekmesinin başka bir nedeni vardır: Bu neden, kurtarma amacıyla tam havuz önbelleğinde tutulan küme bilgilerinin iki yerde bulunduğundan emin olmaktır. Yalnızca bir tam havuz varsa ve kümeyle ilgili bilgilerini kaybederse, kümenin yeniden çalışması için küme içindeki tüm kuyruk yöneticilerine el ile müdahale gerekir. Bununla birlikte, iki tam havuz varsa, bilgiler her zaman iki tam havuza yayınlanır ve bunlara abone olduğundan, başarısız olan tam havuz en az çabayla kurtarılabilir.

- Bu kümenin kullanıcılarını etkilemeden, iki tam havuz kümesi tasarımında tam havuz kuyruđu yöneticileri üzerinde bakım gerçekleştirilebilir: Küme, yalnızca bir havuzla çalışmaya devam eder; böylece, mümkün olan yerlerde havuzları aşağı indirir, bakım uygulamaları ve her defasında bir tane yedekler. İkinci tam havuzda bir kesinti olsa bile, çalışan uygulamalar en az üç gün boyunca etkilenmez.
- Coğrafi nedenlerle coğrafi olarak yerel bir tam havuz kullanmak gibi üçüncü bir havuzu kullanmak için iyi bir neden yoksa, iki havuz tasarımını kullanın. Üç tam havuz olması, şu anda kullanımda olan ikisinin hangilerinin kullanıldığını hiçbir zaman bilmediğiniz anlamına gelir ve birden çok iş yükü yönetimi parametresi arasındaki etkileşimlerin neden olduğu yönetimle ilgili sorunlar olabilir. İki tane fazla tam havuz olması önerilmez.
- Hala daha iyi kullanılabilirliđe gereksiniminiz varsa, tüm havuz kuyruđu yöneticilerini çok eşgörünümlü kuyruk yöneticileri olarak barındırmayı ya da kullanılabilirliđi artırmak için platforma özgü yüksek kullanılabilirlik desteđini kullanmayı düşünün.
- El ile tanımlanan küme gönderen kanallarıyla tüm havuz kuyruđu yöneticilerini tam olarak birbirine bağlamanız gerekir. Küme, haklı bir nedenle ikiden fazla tam havuza sahip olduğunda özellikle dikkatli olunmalıdır. Bu durumda genellikle bir veya daha fazla kanalı gözden kaçırmak ve hemen anlaşılması mümkün değildir. Tam ara bağlantı gerçekleşmediğinde, sorunların tanınması genellikle zordur. Bazı tam havuzlar tüm havuz verilerini tutmadığından ve bu nedenle bađlı oldukları tam havuzlara bađlı olarak kümedeki kuyruk yöneticilerinin farklı görünümlere sahip olmalarına neden olduğundan, bu havuzların tanınması zordur.

Uygulamalar tam havuzlardaki kuyrukları kullanmalı mı?

Tam havuz çoğu şekilde diğer kuyruk yöneticilerine benzer ve bu nedenle uygulama kuyruklarını tam havuzda barındırmak ve uygulamaları doğrudan bu kuyruk yöneticilerine bağlamak mümkündür. Uygulamalar tam havuzlardaki kuyrukları kullanmalı mı?

Yaygın olarak kabul edilen cevap "Hayır?" dir. Bu yapılandırma mümkün olsa da, birçok müşteri bu kuyruk yöneticilerini tam havuz kümesi önbelleğinin bakımı için özel olarak ayırmayı tercih eder. Her iki seçeneğe karar verirken dikkate alınması gereken noktalar burada açıklanmıştır, ancak sonuçta küme mimarisi ortamın belirli taleplerine uygun olmalıdır.

- Yükseltmeler: Genellikle, IBM MQ ' un yeni yayınlarında yeni küme özelliklerini kullanmak için öncelikle o kümenin tam havuz kuyruğu yöneticilerinin yükseltilmesi gerekir. Kümedeki bir uygulama yeni özellikler kullanmak istediğinde, tam havuzları (ve kısmi havuzların bazı alt kümelerini), birlikte bulunan bir dizi uygulamayı sınamadan güncelleyebilmek yararlı olabilir.
- Bakım: Benzer bir şekilde, tam havuzlara acil bakım uygulamanız gerekiyorsa, bunlar **REFRESH** komutuyla uygulamalara dokunmadan yeniden başlatılabilir ya da yenilenebilir.
- Performans: Kümeler büyüdükçe ve tam havuz kümesi önbelleği bakımına ilişkin talepler büyüdükçe, uygulamaların ayrı tutulması, sistem kaynakları için çekişme yoluyla uygulama performansını etkileyen riski azaltır.
- Donanım gereksinimleri: Tipik olarak, tam havuzların güçlü olması gerekmez; örneğin, iyi bir kullanılabilirlik beklentisi olan basit bir UNIX sunucusu yeterlidir. Alternatif olarak, çok büyük veya sürekli değişen kümeler için, tam havuz bilgisayarının performansı dikkate alınmalıdır.
- Yazılım gereksinimleri: Gereksinimler genellikle uygulama kuyruklarını tam bir havuzda barındırmak için ana nedendir. Küçük bir kümede, iş birliği (collocation), tüm kuyruk yöneticilerinin/sunucularının daha az olması için bir gereksinim anlamına gelebilir.

Kanal tanımlarının yönetilmesi

Tek bir küme içinde bile, iki kuyruk yöneticisi arasında birden çok rota sağlayan birden çok kanal tanımları olabilir.

Bazen tek bir küme içinde paralel kanallara sahip olmanın bir avantajı vardır, ancak bu tasarım kararının iyice düşünülmesi gerekir; karmaşıklık eklemenin yanı sıra, bu tasarım, performansı azaltan kanalların az kullanılmasıyla sonuçlanabilir. Bu durum, testlerin genellikle sabit bir hızda çok sayıda ileti göndermeyi içermesi nedeniyle ortaya çıkar, bu nedenle paralel kanallar tam olarak kullanılır. Ancak sürekli olmayan ileti akışının gerçek dünya koşullarıyla, iş yükü dengeleme algoritması, ileti akışı kanaldan kanala geçildikçe performansın düşmesine neden olur.

Bir kuyruk yöneticisi birden çok kümenin üyesiye, her küme için ayrı bir CLUSRCVR kanalı tanımlamak yerine küme ad listesi olan tek bir kanal tanımlama seçeneği vardır. Ancak, bu kurulum daha sonra yönetim sorunlarına neden olabilir; örneğin, TLS ' nin bir kümeye uygulanacağı, ancak bir saniye uygulanmadığı durumu göz önünde bulundurun. Bu nedenle ayrı tanımlar yaratılması tercih edilir ve "[Küme adlandırma kuralları](#)" sayfa 33 içinde önerilen adlandırma kuralı bunu destekler.

Birden çok kanal üzerinden iş yükü dengeleme

Bu bilgi, konunun gelişmiş bir şekilde anlaşılması amacıyla hazırlanmıştır. Bu konuya ilişkin temel açıklamalar için (buradaki bilgileri kullanmadan önce anlaşılması gereken) bkz. [İş yükü yönetimi için kümeleri kullanma](#), [Kümeler içinde iş yükü dengeleme](#) ve [Küme iş yükü yönetimi algoritması](#).

Küme iş yükü yönetimi algoritması büyük bir araç kümesi sağlar, ancak bunların tümü nasıl çalıştıklarını ve etkileşimde bulduklarını tam olarak anlamadan birbirleriyle birlikte kullanılmamalıdır. İş yükü dengeleme süreci için kanalların ne kadar önemli olduğu hemen belli olmayabilir: İş yükü yönetimi çevrimsel sıralı algoritması, kümelenebilir bir kuyruğu olan bir kuyruk yöneticisine birden çok küme kanalı gibi davranır ve bu kuyruğun birden çok eşgörünümü olarak kabul edilir. Bu işlem aşağıdaki örnekte daha ayrıntılı olarak açıklanmıştır:

1. Bir kümede kuyruk barındıran iki kuyruk yöneticisi vardır: QM1 ve QM2.

2. QM1 için beş küme alıcı kanalı vardır.
3. QM2 için yalnızca bir küme alıcı kanalı vardır.
4. QM3 üzerinde **MQPUT** ya da **MQOPEN** bir eşgörünüm seçtiğinde, algoritmanın iletiyi QM1 'e gönderme olasılığı QM2' e göre beş kat daha fazladır.
5. 4 numaralı adımdaki durum, algoritma beş kanaldan QM1 'e ve tek kanaldan QM2' e altı seçenekten seçim yapabileceği (5 + 1) ve yuvarlak robinler gördüğünden ortaya çıkar.

Diğer bir ince davranış, iletileri yerel kuyruk yöneticisinde tek bir yönetim ortamının yapılandırılmış olduğu kümelenmiş bir kuyruğa koyarken bile, IBM MQ iletilerin kuyruğun yerel eşgörünümüne mi, yoksa kuyruğun uzak eşgörünümüne mi konacağına karar vermek için yerel küme alıcı kanalının durumunu kullanmasıdır. Bu senaryoda:

1. İş yükü yönetimi algoritmasının tek tek küme kuyruklarına bakmadığı iletiler konurken, bu hedeflere ulaşabilen küme kanallarına bakar.
2. Yerel hedeflere ulaşmak için, yerel alıcı kanalları bu listeye eklenir (ileti göndermek için kullanılmamalarına rağmen).
3. Bir yerel alıcı kanal durdurulduğunda, iş yükü yönetimi algoritması, CLUSRCVR durdurulmamışsa varsayılan olarak alternatif bir örneği tercih eder. Hedef için birden çok yerel CLUSRCVR yönetim ortamı varsa ve en az biri durdurulmazsa, yerel yönetim ortamı uygun kalır.

Kümeleme: Birden çok küme iletim kuyruğu kullanan uygulama yalıtımı

Bir kümedeki kuyruk yöneticileri arasındaki ileti akışlarını yalıtabilirsiniz. Farklı küme gönderen kanalları tarafından iletilen iletileri farklı küme iletim kuyruklarına yerleştirebilirsiniz. Yaklaşımı tek bir kümede ya da çakışan kümelerle kullanabilirsiniz. Bu konuda, kullanmak üzere bir yaklaşım seçerken size yol gösterecek örnekler ve bazı en iyi uygulamalar sağlanmaktadır.

Bir uygulamayı konuşlandırdığınızda, IBM MQ ' in diğer uygulamalarla hangi kaynakları paylaştığını ve hangi kaynakları paylaşmadığını seçme seçeneğiniz vardır. Paylaşılacak kaynak tipleri vardır; ana kaynaklar sunucunun kendisi, kuyruk yöneticisi, kanallar ve kuyruklardır. Uygulamaları daha az paylaşılan kaynakla yapılandırmayı; ayrı kuyruklar, kanallar, kuyruk yöneticileri ya da hatta tek tek uygulamalara sunucu ayırmayı seçebilirsiniz. Bunu yaparsanız, genel sistem yapılandırması daha büyük ve daha karmaşık hale gelir. IBM MQ kümelerinin kullanılması, daha fazla sunucunun, kuyruk yöneticisinin, kuyruğun ve kanalın yönetilmesinin karmaşıklığını azaltır, ancak başka bir paylaşılan kaynak olan küme iletim kuyruğu SYSTEM . CLUSTER . TRANSMIT . QUEUE ' u tanıtır.

Şekil 12 sayfa 47 , paylaşımın önemini gösteren büyük bir IBM MQ dağıtımından bir dilimdir SYSTEM . CLUSTER . TRANSMIT . QUEUE . Çizgede, Client App uygulaması CL1 kümesindeki QM2 kuyruk yöneticisine bağlıdır. Client App ' den gelen bir ileti, Server App uygulaması tarafından işlenir. İleti, Server App tarafından CLUSTER2 içindeki QM3 kuyruk yöneticisindeki Q1 küme kuyruğundan alınır. İstemci ve sunucu uygulamaları aynı kümede olmadığı için, ileti QM1 ağ geçidi kuyruk yöneticisi tarafından aktarılır.

Küme ağ geçidini yapılandırmanın normal yolu, ağ geçidi kuyruk yöneticisini tüm kümelerin üyesi yapmaktır. Ağ geçidi kuyruk yöneticisinde, tüm kümelerdeki küme kuyrukları için tanımlanmış kümelenmiş diğer ad kuyrukları vardır. Kümelenmiş kuyruk diğer adları tüm kümelerde kullanılabilir. Küme kuyruğu diğer adlarına yerleştirilen iletiler, ağ geçidi kuyruk yöneticisi aracılığıyla doğru hedeflerine yönlendirilir. Ağ geçidi kuyruk yöneticisi, kümelenmiş diğer ad kuyruklarına gönderilen iletileri QM1 üzerindeki ortak SYSTEM . CLUSTER . TRANSMIT . QUEUE üzerine koyar.

Göbek ve konuşma mimarisi, kümeler arasındaki tüm iletilerin ağ geçidi kuyruk yöneticisinden geçmesini gerektirir. Sonuç, tüm iletilerin QM1, SYSTEM . CLUSTER . TRANSMIT . QUEUE üzerindeki tek küme iletim kuyruğundan akması olur.

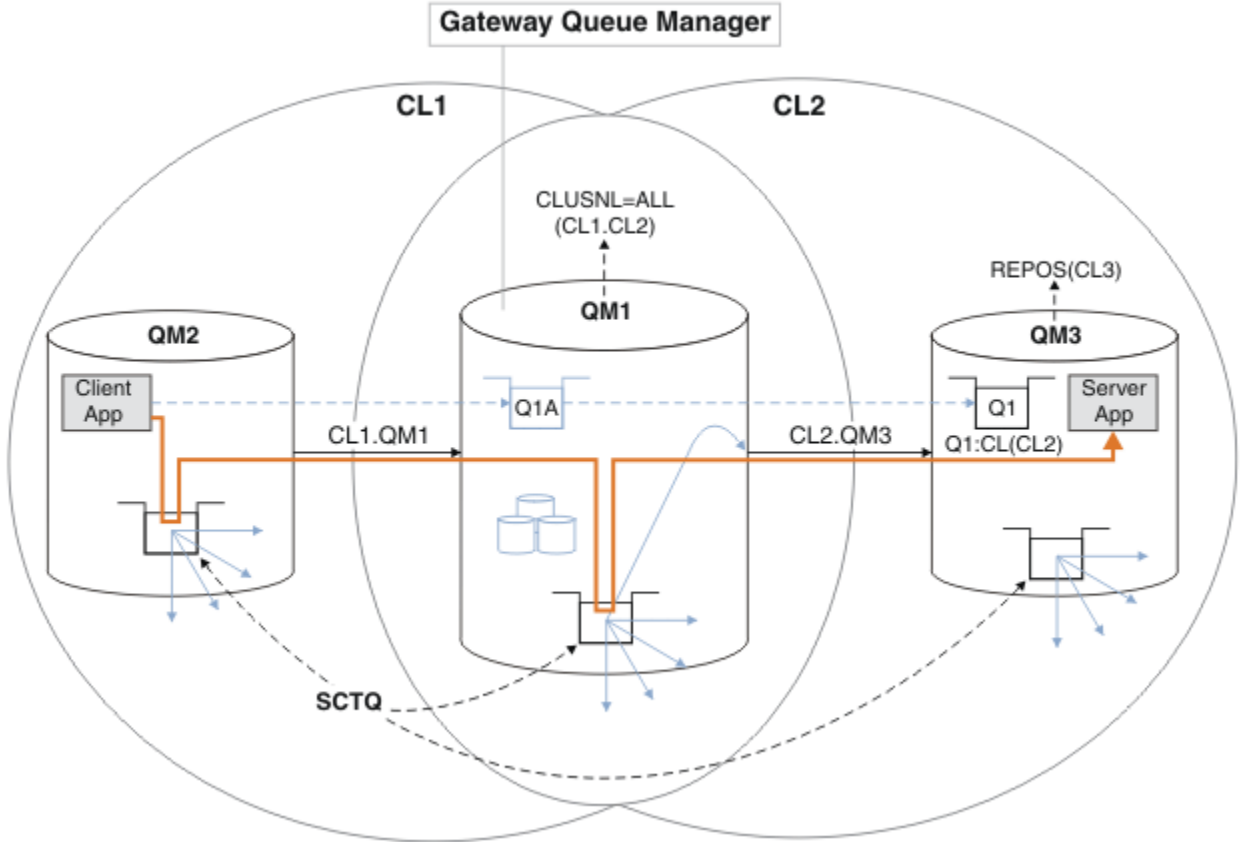
Başarım açısından bakıldığında, tek bir kuyruk sorun değildir. Genel bir iletim kuyruğu genellikle bir performans darboğazı göstermez. Ağ geçidindeki ileti çıkışı büyük ölçüde, ağ geçidine bağlanan kanalların performansına göre belirlenir. Üretilen iş genellikle kuyruk sayısından ya da kanalları kullanan kuyruklardaki ileti sayısından etkilenmez.

Diğer bazı bakış açılarından, birden çok uygulama için tek bir iletim kuyruğunun kullanılması bazı engellere sahiptir:

- Bir hedefteki ileti akışını, ileti akışından başka bir hedefe ayıramazsınız. Hedefler farklı kuyruk yöneticilerindeki farklı kümelerde olsa da, iletilerin iletilmeden önce depolanmasını ayıramazsınız.

Bir küme hedefi kullanılamaz duruma gelirse, tek bir iletim kuyruğunda hedef oluşturma için iletiler ve sonunda iletiler onu doldurur. İletim kuyruğu dolduğunda, iletilerin herhangi bir küme hedefi için iletim kuyruğuna yerleştirilmesine engel olur.

- İletilerin farklı küme hedeflerine aktarılmasını izlemek kolay değildir. Tüm iletiler tek bir iletim kuyruğunda. İletim kuyruğunun derinliğini görüntülemek, iletilerin tüm hedeflere aktarılıp aktarılmadığını gösteremez.



Not: Şekil 12 sayfa 47 ve aşağıdaki şekillerdeki oklar farklı türlerdedirler. Kesintisiz oklar ileti akışlarını gösterir. Kesiksiz oklardaki etiketler, ileti kanalı adlarıdır. Gri kesiksiz oklar, SYSTEM. CLUSTER. TRANSMIT. QUEUE ' den küme gönderen kanallara olası ileti akışlarıdır. Siyah kesikli çizgiler, etiketleri hedeflerine bağlar. Gri kesik çizgili oklar başvurulardır; örneğin, MQOPEN Client App tarafından küme diğer adı kuyruğu tanımına yapılan bir çağrıdan Q1A.

Şekil 12. IBM MQ kümelerini kullanarak merkeze konuşlandırılan ve mimariye konuşlandırılmış istemci-sunucu uygulaması

Şekil 12 sayfa 47'inde, Server App istemcileri kuyruğu açar Q1A. İletiler, QM2üzerinde SYSTEM. CLUSTER. TRANSMIT. QUEUE ögesine, QM1üzerinde SYSTEM. CLUSTER. TRANSMIT. QUEUE ögesine aktarılır ve daha sonra, Server App uygulaması tarafından alındıkları QM3üzerinde Q1 ' e aktarılır.

Client App iletisi, QM2 ve QM1üzerindeki sistem kümesi iletim kuyruklarından geçer. Şekil 12 sayfa 47' de amaç, ağ geçidi kuyruk yöneticisindeki ileti akışını istemci uygulamasından yalıtılmaktır; böylece, iletileri SYSTEM. CLUSTER. TRANSMIT. QUEUEüzerinde saklanmaz. Diğer kümelenmiş kuyruk yöneticilerinden herhangi birinde akışları yalıtabilirsiniz. Akışları diğer yönde, istemciye geri izole edebilirsiniz. Çözümlerin açıklamalarını kısa tutmak için, açıklamalar istemci uygulamasından yalnızca tek bir akışı dikkate alır.

Küme ağ geçidi kuyruk yöneticisinde küme iletisi trafiğini yalıtma çözümleri

Sorunu çözmenin bir yolu, kümeler arasında köprü kurmak için kuyruk yöneticisi diğer adlarını ya da uzak kuyruk tanımlarını kullanmaktır. Ağ geçidi kuyruk yöneticisindeki her ileti akışını ayırmak için kümelendiği bir uzak kuyruk tanımı, bir iletim kuyruğu ve bir kanal yaratın; Ağ geçidi kuyruk yöneticisinden gönderilen iletileri yalıtma için uzak kuyruk tanımlaması eklenmesibaşlıklı konuya bakın.

IBM WebSphere MQ 7.5 ' den başlayarak, küme kuyruğu yöneticileri tek bir küme iletim kuyruğuyla sınırlı değildir. İki seçeneğiniz vardır:

1. Ek küme iletim kuyruklarını el ile tanımlayın ve her iletim kuyruğundan hangi küme gönderen kanallarının aktarma iletilerini tanımlayın; Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtma için küme iletim kuyruğu eklenmesibaşlıklı konuya bakın.
2. Kuyruk yöneticisinin ek küme iletim kuyruklarını otomatik olarak yaratmasını ve yönetmesini sağla. Her küme gönderen kanalı için farklı bir küme iletim kuyruğu tanımlar; bkz. İleti trafiğini yalıtma için varsayılan küme iletim kuyruklarını ayırmak üzere değiştirme.

Bazı küme gönderen kanalları için el ile tanımlanan küme iletim kuyruklarını, gerisini yöneten kuyruk yöneticisiyle birleştirebilirsiniz. İletim kuyruklarının birleşimi, Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtma için küme iletim kuyruğu eklenmesibaşlıklı konudaki yaklaşımdır. Bu çözümde, kümeler arasındaki çoğu ileti ortak SYSTEM . CLUSTER . TRANSMIT . QUEUE kullanır. Bir uygulama kritik önem taşıyor ve tüm ileti akışları, el ile tanımlanan bir küme iletim kuyruğu kullanılarak diğer akışlardan yalıtılır.

Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtma için küme iletim kuyruğu eklenmesi içindeki yapılandırma sınırlıdır. Bu, başka bir küme kuyruğuyla aynı kümedeki aynı kuyruk yöneticisinde bulunan bir küme kuyruğuna giden ileti trafiğini ayırmaz. Dağıtılmış kuyruğun bir parçası olan uzak kuyruk tanımlamalarını kullanarak ileti trafiğini tek tek kuyruklara ayırabilirsiniz. Kümelerde, birden çok küme iletim kuyruğu kullanarak, farklı küme gönderen kanallarına giden ileti trafiğini ayırabilirsiniz. Aynı kümedeki birden çok küme kuyruğu, aynı kuyruk yöneticisinde, bir kümeyi gönderen kanalı paylaşır. Bu kuyruklara ilişkin iletiler, ağ geçidi kuyruk yöneticisinden iletilmeden önce aynı iletim kuyruğunda saklanır. Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtma için küme ve küme iletim kuyruğu eklenmesiyapılanışında, başka bir küme eklenerek ve kuyruk yöneticisi ve küme kuyruğu yeni kümenin bir üyesi olarak belirlenerek sınırlama yan adımdır. Yeni kuyruk yöneticisi, kümedeki tek kuyruk yöneticisi olabilir. Kümeye daha fazla kuyruk yöneticisi ekleyebilir ve bu kuyruk yöneticilerindeki küme kuyruklarını yalıtma için aynı kümeyi kullanabilirsiniz.

İlgili kavramlar

“Erişim denetimi ve birden çok küme iletim kuyruğu” sayfa 28

Bir uygulama iletileri uzak küme kuyruklarına koyduğunda üç denetleme kipi arasından seçim yapın. Kipler, küme kuyruğuna göre uzaktan denetleniyor, SYSTEM . CLUSTER . TRANSMIT . QUEUE ile yerel olarak denetleniyor ya da küme kuyruğuna ya da küme kuyruğu yöneticisine ilişkin yerel tanımlarla karşılaştırılıyor.

Küme iletim kuyrukları ve küme gönderen kanallarıyla çalışma

“Çakışan kümeler” sayfa 35

Çakışan kümeler, ek yönetim yetenekleri sağlar. Çakışan kümeleri yönetmek için gereken komut sayısını azaltma için ad listesi kullanın.

İlgili görevler

Uzak küme kuyruklarına ileti konmasına yetki verilmesi

Ağ geçidi kuyruk yöneticisinden gönderilen iletileri yalıtma için uzak kuyruk tanımlaması eklenmesi

Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtma için küme iletim kuyruğu eklenmesi

Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtma için küme ve küme iletim kuyruğu eklenmesi

İleti trafiğini yalıtma üzere küme iletim kuyruklarını ayırmak için varsayılan değerin değiştirilmesi

Ağ geçidi kuyruk yöneticisiyle çakışan iki küme oluşturma

Kümeler arasındaki ileti yollarının yapılandırılması

güvenlik

İlgili başvurular

setmqaut

Kümeleme: Küme iletim kuyruklarının nasıl yapılandırılacağını planlama

Küme iletim kuyruklarının seçiminde size yol gösterilir. Bir ortak varsayılan kuyruk, ayrı varsayılan kuyruklar ya da el ile tanımlanan kuyruklar yapılandırabilirsiniz.

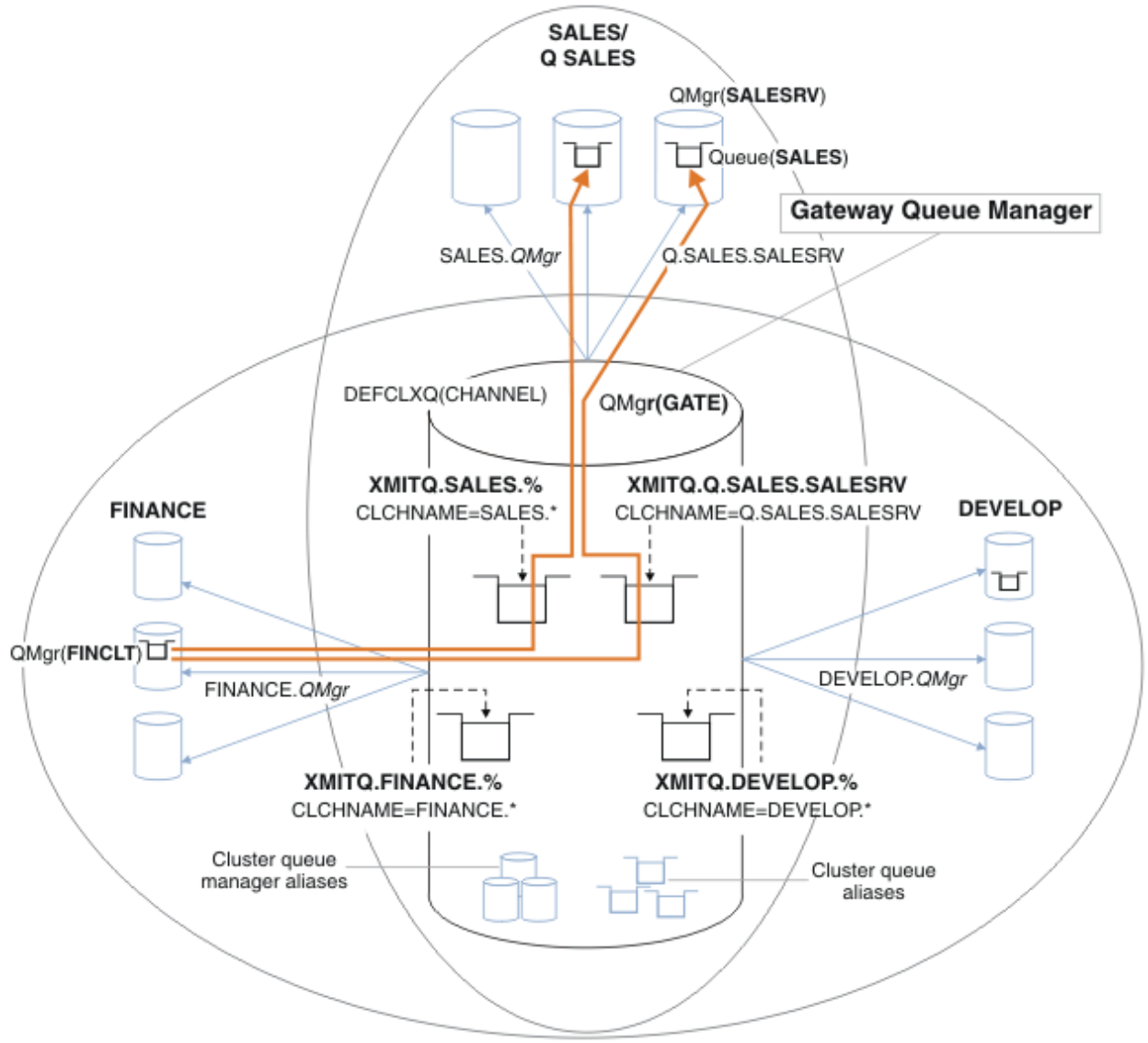
Başlamadan önce

[“Hangi küme iletim kuyruğunun kullanılacağını seçme” sayfa 51¹](#) i gözden geçirin.

Bu görev hakkında

Bir kuyruk yöneticisini küme iletim kuyruğu seçecek şekilde nasıl yapılandıracağınızı planlarken bazı seçenekler vardır.

1. Küme ileti aktarımları için varsayılan küme iletim kuyruğu nedir?
 - a. Ortak bir küme iletim kuyruğu, SYSTEM . CLUSTER . TRANSMIT . QUEUE.
 - b. Ayrı küme iletim kuyrukları. Kuyruk yöneticisi, ayrı küme iletim kuyruklarını yönetir. Bunları model kuyruğundan (SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE) kalıcı dinamik kuyruklar olarak oluşturur. Kullandığı her küme gönderen kanalı için bir küme iletim kuyruğu yaratır.
2. El ile yaratmaya karar verdiğiniz küme iletim kuyrukları için iki seçeneğiniz daha vardır:
 - a. Konfigürasyonunu el ile tanımlamaya karar verdiğiniz her kümeyi gönderen kanal için ayrı bir iletim kuyruğu tanımlayın. Bu durumda, iletim kuyruğunun **CLCHNAME** kuyruk özniteliğini bir küme gönderen kanalının adına ayarlayın. Bu iletim kuyruğundan ileti aktarmak için küme gönderen kanalını seçin.
 - b. Bir grup kümeyi gönderen kanal için ileti trafiğini aynı küme iletim kuyruğunda birleştirin; bkz. [Şekil 13 sayfa 50](#). Bu durumda, her bir ortak iletim kuyruğunun **CLCHNAME** kuyruk özniteliğini soysal bir küme gönderen kanal adına ayarlayın. Soysal küme gönderen kanal adı, küme gönderen kanal adlarını gruplamak için kullanılan bir süzgeçtir. Örneğin, SALES . * , SALES . ile başlayan adları olan tüm küme gönderen kanallarını gruplar. Süzgeç dizgisinin herhangi bir yerine birden çok genel arama karakteri yerleştirebilirsiniz. Genel arama karakteri bir yıldız işaretidir ("*"). Sıfır ile herhangi bir sayıda karakteri temsil eder.



Şekil 13. Farklı departman IBM MQ kümeleri için belirli iletim kuyrukları örneği

Yordam

1. Kullanılacak varsayılan küme iletim kuyruğunun tipini seçin.

- Tek bir küme iletim kuyruğu ya da her küme bağlantısı için ayrı kuyruklar seçin.

Varsayılan ayarı bırakın ya da **MQSC** komutunu çalıştırın:

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

2. Küme iletim kuyruğunu diğer akışlarla paylaşmaması gereken ileti akışlarını yalıt.

- Bkz. “Kümeleme: Birden çok küme iletim kuyruğunun konfigürasyonu örneği” sayfa 53. Yalıtılması gereken SALES kuyruğu, SALESRV üzerindeki SALES kümesinin bir üyesidir. SALES Kuyruğu yalıtılmak, yeni bir küme yaratmak Q.SALES, SALESRV kuyruk yöneticisini üye yapmak ve SALES kuyruğunu Q.SALES kuyruğuna ait olacak şekilde değiştirmek için.
- SALES ' e ileti gönderen kuyruk yöneticileri de yeni kümenin üyeleri olmalıdır. Örneğin, kümelmiş bir kuyruk diğer adını ve bir ağ geçidi kuyruk yöneticisini kullanırsanız, birçok durumda değişiklikleri ağ geçidi kuyruk yöneticisini yeni kümenin üyesi yapacak şekilde sınırlayabilirsiniz.

- Ancak, akışların ağ geçidinden hedefe ayrılması, akışları kaynak kuyruk yöneticisinden ağ geçidine ayırmaz. Ama bazen geçitten akan akışları ayırmak için yeterli olduğu ve geçit geçidine akmadığı ortaya çıkar. Yeterli değilse, kaynak kuyruk yöneticisini yeni kümeye ekleyin. İletilerin ağ geçidi üzerinden geçmesini istiyorsanız, küme diğer adını yeni kümeye taşıyın ve iletileri doğrudan hedef kuyruk yöneticisine değil, ağ geçidindeki küme diğer adına göndermeye devam edin.

İleti akışlarını yalıtım için aşağıdaki adımları izleyin:

- a) Akışların hedeflerini, her bir hedef kuyruğun belirli bir kümedeki tek kuyruk olacak şekilde yapılandırın (o kuyruk yöneticisinde).
 - b) Sistematik bir adlandırma kuralına göre oluşturduğunuz yeni kümeler için küme gönderen ve küme alıcı kanalları oluşturun.
 - Bkz. [“Kümeleme: çakışan kümeler için dikkat edilmesi gereken özel noktalar” sayfa 42.](#)
 - c) Hedef kuyruğa ileti gönderen her kuyruk yöneticisinde her yalıtılmış hedef için bir küme iletim kuyruğu tanımlayın.
 - Küme iletim kuyruklarına ilişkin adlandırma kuralı, önceki XMITQ . olan CLCHNAMEküme kanalı adı özniteliğinin değerini kullanmaktır.
3. Yönetişim ya da izleme gereksinimlerini karşılamak için küme iletim kuyrukları oluşturun.
- Tipik yönetim ve izleme gereksinimleri, her bir küme için bir iletim kuyruğuna ya da her bir kuyruk yöneticisi için bir iletim kuyruğuna neden olur. Küme kanalları için adlandırma kuralını izlerseniz, *ClusterName* . *QueueManagerName*, bir kuyruk yöneticisi kümesini ya da bir kuyruk yöneticisinin üyesi olduğu tüm kümeleri seçen soysal kanal adları oluşturmak kolaydır; bkz. [“Kümeleme: Birden çok küme iletim kuyruğunun konfigürasyonu örneği” sayfa 53.](#)
 - Yıldız işareti simgesini yüzde işaretiyle değiştirerek, küme iletim kuyruklarına ilişkin adlandırma kuralını soysal kanal adlarını kapsayacak şekilde genişletin. Örneğin,

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

İlgili kavramlar

Küme iletim kuyrukları ve küme gönderen kanallarıyla çalışma

[“Erişim denetimi ve birden çok küme iletim kuyruğu” sayfa 28](#)

Bir uygulama iletileri uzak küme kuyruklarına koyduğunda üç denetleme kipi arasından seçim yapın. Kipler, küme kuyruğuna göre uzaktan denetleniyor, SYSTEM . CLUSTER . TRANSMIT . QUEUE ile yerel olarak denetleniyor ya da küme kuyruğuna ya da küme kuyruğu yöneticisine ilişkin yerel tanımlarla karşılaştırılıyor.

[“Çakışan kümeler” sayfa 35](#)

Çakışan kümeler, ek yönetim yetenekleri sağlar. Çakışan kümeleri yönetmek için gereken komut sayısını azaltmak için ad listesi kullanın.

İlgili görevler

[Ağ geçidi kuyruk yöneticisinden gönderilen iletileri yalıtım için uzak kuyruk tanımlaması eklenmesi](#)

[Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtım için küme iletim kuyruğu eklenmesi](#)

[Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtım için küme ve küme iletim kuyruğu eklenmesi](#)

[İleti trafiğini yalıtım üzere küme iletim kuyruklarını ayırmak için varsayılan değerin değiştirilmesi](#)

[Ağ geçidi kuyruk yöneticisiyle çakışan iki küme oluşturma](#)

[Kümeler arasındaki ileti yollarının yapılandırılması](#)

Hangi küme iletim kuyruğunun kullanılacağını seçme

Farklı küme iletim kuyruğu konfigürasyon seçenekleri arasında seçim yapabilirsiniz.

Hangi küme iletim kuyruğunun bir küme gönderen kanalla ilişkilendirileceğini seçebilirsiniz.

1. Tek bir varsayılan küme iletim kuyruğuyla (SYSTEM . CLUSTER . TRANSMIT . QUEUE) ilişkilendirilmiş tüm küme gönderen kanalları olabilir; bu seçenek varsayılan değerdir.

2. Tüm küme gönderen kanallarını, ayrı bir küme iletim kuyruğuyla otomatik olarak ilişkilendirilecek şekilde ayarlayabilirsiniz. Kuyruklar, SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE model kuyruğundan SYSTEM . CLUSTER . TRANSMIT . *ChannelName* adlı kuyruk yöneticisi tarafından yaratılır. **DEFCLXQ** kuyruk yöneticisi özniteliği CHANNEL olarak ayarlanırsa, kanallar benzersiz olarak adlandırılmış küme iletim kuyruğunu kullanır.
3. Tek bir küme iletim kuyruğunun hizmet vereceği belirli küme gönderen kanalları ayarlayabilirsiniz. Bir iletim kuyruğu yaratarak ve **CLCHNAME** özniteliğini küme gönderen kanalının adına ayarlayarak bu seçeneği belirleyin.
4. Tek bir küme iletim kuyruğunun hizmet vereceği küme gönderen kanal gruplarını seçebilirsiniz. Bir iletim kuyruğu yaratıp **CLCHNAME** özniteliğini *ClusterName* . * gibi soysal bir kanal adına ayarlayarak bu seçeneği belirleyin. “Kümeleme: çakışan kümeler için dikkat edilmesi gereken özel noktalar” sayfa 42’deki adlandırma kurallarını izleyerek küme kanallarını adlandırırsanız, bu ad, kümedeki kuyruk yöneticilerine bağlı tüm küme kanallarını seçer *ClusterName*.

Bazı küme gönderen kanalları için varsayılan küme iletim kuyruğu seçeneklerinden birini belirli ve soysal küme iletim kuyruğu konfigürasyonlarıyla birleştirebilirsiniz.

En iyi uygulamalar

Çoğu durumda, var olan IBM MQ kuruluşları için varsayılan yapılandırma en iyi seçimdir. Küme kuyruğu yöneticisi, küme iletilerini tek bir küme iletim kuyruğunda (SYSTEM . CLUSTER . TRANSMIT . QUEUE) saklar. Farklı kuyruk yöneticilerine ve farklı kümelere ilişkin iletileri ayrı iletim kuyruklarında saklamak ya da kendi iletim kuyruklarını tanımlamak için varsayılan değeri değiştirme seçeneğiniz vardır.

Çoğu durumda, yeni IBM MQ kuruluşları için varsayılan yapılandırma da en iyi seçimdir. Varsayılan konfigürasyondan, her küme gönderen kanalı için bir iletim kuyruğuna sahip olma varsayılanına geçiş işlemi otomatik olarak gerçekleştirilir. Geri dönüş de otomatiktir. Birinin ya da diğersinin seçimi kritik değildir, tersine çevirebilirsiniz.

Farklı bir yapılandırma seçilmesinin nedeni, işlevsellik ya da performanstan çok yönetim ve yönetimle ilgili. Birkaç kural dışı durum söz konusu olduğunda, birden çok küme iletim kuyruğunun yapılandırılması, kuyruk yöneticisinin davranışına fayda sağlamaz. Daha fazla kuyruğa neden olur ve önceden ayarladığınız izleme ve yönetim yordamlarını tek bir iletim kuyruğuna başvurarak değiştirmenizi gerektirir. Bu nedenle, dengede, farklı bir seçim için güçlü yönetim ya da yönetim nedenleriniz yoksa, varsayılan yapılandırmayla birlikte kalmanız en iyi seçimdir.

Her iki kural dışı durum da, SYSTEM . CLUSTER . TRANSMIT . QUEUE üzerinde saklanan iletilerin sayısı artarsa ne olacağı ile ilgilidir. Bir hedefe ilişkin iletileri başka bir hedefe ilişkin iletilerden ayırmak için her adımı atarsanız, bir hedefe yönelik kanal ve teslim sorunları, başka bir hedefe teslim edilmesini etkilememeli. Ancak, iletilerin tek bir hedefe yeterince hızlı teslim edilmemesi nedeniyle SYSTEM . CLUSTER . TRANSMIT . QUEUE üzerinde depolanan ileti sayısı artabilir. Bir hedefe ilişkin SYSTEM . CLUSTER . TRANSMIT . QUEUE üzerindeki iletilerin sayısı, iletilerin diğer hedeflere gönderilmesini etkileyebilir.

Tek bir iletim kuyruğunun doldurulmasıyla sonuçlanan sorunları önlemek için yapılandırmanıza yeterli kapasite oluşturmayı hedefleyin. Daha sonra, bir hedef başarısız olursa ve bir ileti birikim listesi oluşmaya başlarsa, sorunu düzeltmek için zamanınız olur.

İletiler, küme ağ geçidi gibi bir göbek kuyruk yöneticisi aracılığıyla yönlendirilirse, ortak bir iletim kuyruğunu (SYSTEM . CLUSTER . TRANSMIT . QUEUE) paylaşırlar. Ağ geçidi kuyruk yöneticisinde SYSTEM . CLUSTER . TRANSMIT . QUEUE üzerinde saklanan ileti sayısı derinlik üst sınırına ulaşırsa, kuyruk yöneticisi, derinliği azalınca kadar iletim kuyruğuna ilişkin yeni iletileri reddetmeye başlar. Tıkanıklık, ağ geçidi üzerinden yönlendirilen tüm hedeflere ilişkin iletileri etkiler. İletiler, ağ geçidine ileti gönderen diğer kuyruk yöneticilerinin iletim kuyruklarını yedekler. Sorun, kuyruk yöneticisi hata günlüklerine yazılan iletilerde, ileti verimi düştüğünde ve ileti gönderilmesiyle iletinin hedefine ulaşma süresi arasında geçen daha uzun sürelerde kendini gösterir.

Tıkanıklığın tek bir iletim kuyruğu üzerindeki etkisi, dolmadan önce bile belirginleşebilir. Bazı büyük kalıcı olmayan iletiler ve bazı küçük iletilerle karışık ileti trafiğiniz varsa, iletim kuyruğu doldukça küçük iletileri teslim etme süresi artar. Gecikme, normalde diske yazılmayacak büyük kalıcı olmayan iletilerin diske

yazılmasından kaynaklanıyor. Bir küme iletim kuyruğunu diğer karma ileti akışlarıyla paylaşarak zaman açısından kritik ileti akışlarına sahipseniz, bu, diğer ileti akışlarından yalıtılmak için özel bir ileti yolu yapılandırmaya değer olabilir; Ağ geçidi kuyruk yöneticisinden gönderilen küme ileti trafiğini yalıtılmak için küme ve küme iletim kuyruğu eklenmesibaşlıklı konuya bakın.

Ayrı küme iletim kuyruklarının yapılandırılmasının diğer nedenleri yönetim gereksinimlerinin karşılanması ya da farklı küme hedeflerine gönderilen izleme iletilerinin basitleştirilmesidir. Örneğin, bir hedefe ilişkin iletilerin başka bir hedefe ilişkin iletilerle hiçbir zaman bir iletim kuyruğunu paylaşmadığını göstermeniz gerekebilir.

Varsayılan küme iletim kuyruğunu denetleyen **DEFCLXQ** kuyruk yöneticisi özniteliğini, her küme gönderen kanalı için farklı küme iletim kuyrukları yaratacak şekilde değiştirin. Birden çok hedef bir küme-gönderen kanalı paylaşabilir, bu nedenle kümelerinizi bu amacı tam olarak karşılayacak şekilde planlamanız gerekir. Ağ geçidi kuyruk yöneticisinden gönderilen küme ileti trafiğini yalıtılmak için küme ve küme iletim kuyruğu ekleme yöntemini sistematik olarak tüm küme kuyruklarınıza uygulayın. Hedeflediğiniz sonuç, hiçbir küme hedefinin bir küme gönderen kanalı başka bir küme hedefiyle paylaşmaması. Sonuç olarak, bir küme hedefine ilişkin hiçbir ileti, küme iletim kuyruğunu başka bir hedefe ilişkin bir iletiyle paylaşmaz.

Belirli bir ileti akışı için ayrı bir küme iletim kuyruğu yaratılması, o hedefe giden iletilerin akışını izlemeyi kolaylaştırır. Yeni bir küme iletim kuyruğu kullanmak için, kuyruğu tanımlayın, bir küme gönderen kanalla ilişkilendirin ve kanalı durdurun ve başlatın. Değişimin kalıcı olması gerekmiyor. Bir süre ileti akışını yalıtıp, iletim kuyruğunu izleyebilir ve varsayılan iletim kuyruğunu yeniden kullanmaya geri dönebilirsiniz.

İlgili görevler

Kümeleme: Birden çok küme iletim kuyruğunun konfigürasyonu örneği

Bu görevde, birden çok küme iletim kuyruğunun çakışan üç kümeye planlanması için gereken adımları uygulayabilirsiniz. Gereksinimler, ileti akışlarını bir küme kuyruğuna, diğer tüm ileti akışlarından ayırmayı ve farklı küme iletim kuyruklarındaki farklı kümelere ilişkin iletileri saklamayı içerir.

Kümeleme: Küme iletim kuyrukları arasında geçiş yapma

Var olan bir üretim kuyruğu yöneticisinin küme iletim kuyruklarındaki değişikliklerin nasıl yürürlüğe gireceğini planlayın.

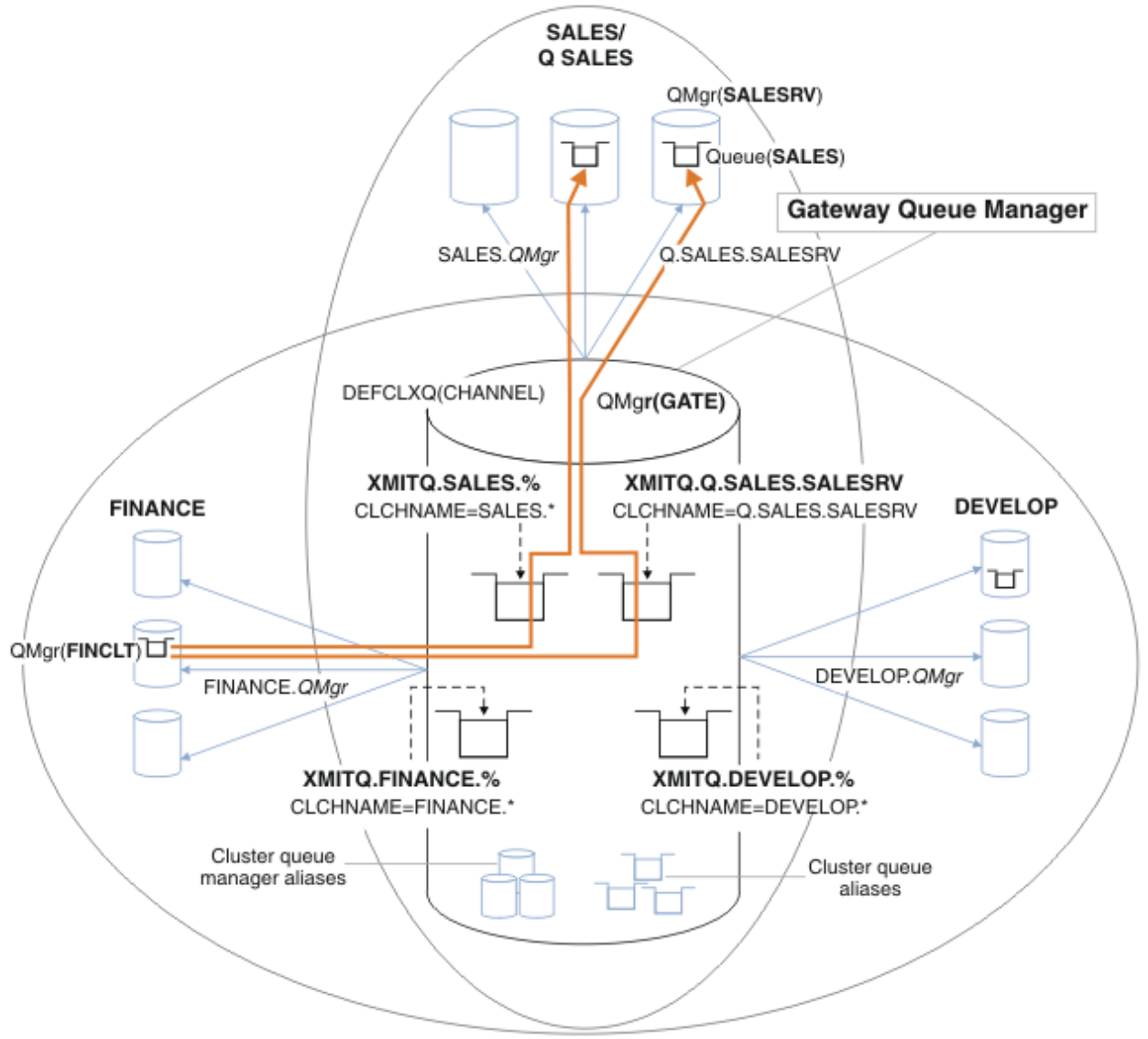
Kümeleme: Birden çok küme iletim kuyruğunun konfigürasyonu örneği

Bu görevde, birden çok küme iletim kuyruğunun çakışan üç kümeye planlanması için gereken adımları uygulayabilirsiniz. Gereksinimler, ileti akışlarını bir küme kuyruğuna, diğer tüm ileti akışlarından ayırmayı ve farklı küme iletim kuyruklarındaki farklı kümelere ilişkin iletileri saklamayı içerir.

Bu görev hakkında

Bu görevdeki adımlar, yordamın "Kümeleme: Küme iletim kuyruklarının nasıl yapılandırılacağını planlama" sayfa 49 içinde nasıl uygulanacağını ve Şekil 14 sayfa 54 içinde gösterilen yapılandırmaya nasıl ulaşacağını gösterir. Bu, ayrı küme iletim kuyruklarıyla yapılandırılan bir ağ geçidi kuyruk yöneticisiyle çakışan üç kümeye bir örnektir. Kümeleri tanımlamak için MQSC komutları "Örnek kümelerin yaratılması" sayfa 56 içinde açıklanmıştır.

Örneğin, iki gereksinim vardır. Bunlardan biri, ileti akışını ağ geçidi kuyruk yöneticisinden satışları günlüğe kaydeden satış uygulamasına ayırmaktır. İkincisi, herhangi bir zamanda farklı bölümlere gönderilmeyi bekleyen ileti sayısını sorgulamaktır. SALES, FINANCE ve DEVELOP kümeleri zaten tanımlı. Küme iletileri şu anda SYSTEM . CLUSTER . TRANSMIT . QUEUE' den iletiliyor.



Şekil 14. Farklı departman IBM MQ kümeleri için belirli iletim kuyrukları örneği

Kümeleri değiştirme adımları aşağıdaki gibidir. Tanımlar için Yeni bir kümede satış kuyruğunu yalıtmak ve ağ geçidi kümesi iletim kuyruklarını ayırmak için yapılan değişiklikler başlıklı konuya bakın.

Yordam

1. İlk yapılandırma adımı " Kullanılacak varsayılan küme iletim kuyruğunun tipini seçin " dir.

Bu karar, GATE kuyruk yöneticisinde aşağıdaki **MQSC** komutunu çalıştırarak ayrı varsayılan küme iletim kuyrukları yaratmaktır.

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

Bu varsayılanı seçmek için güçlü bir neden yoktur; amaç, küme iletim kuyruklarını el ile tanımlamaktır. Seçim zayıf bir tanımlama değerine sahip. El ile tanımlama yanlış yapılırsa ve ileti varsayılan bir küme iletim kuyruğundan aşağı akarsa, kalıcı dinamik küme iletim kuyruğunun yaratılışında bu ileti görüntülenir.

2. İkinci yapılandırma adımı " Küme iletim kuyruğunu diğer akışlarla paylaşmaması gereken ileti akışlarını yalıt " dir.

Bu durumda, SALESRV üzerindeki SALES kuyruğundan ileti alan satış uygulaması yalıtma gerektirir. Yalnızca ağ geçidi kuyruk yöneticisinden iletilerin yalıtılması gerekir. Üç alt adım şunlardır:

- a) "Akışların hedeflerini, her bir hedef kuyruğun belirli bir kümedeki tek kuyruk olacak şekilde yapılandırın (o kuyruk yöneticisinde)".

Bu örnek, SALESRV kuyruk yöneticisinin satış departmanındaki yeni bir kümeye eklenmesini gerektirir. Yalıtım gerektiren birkaç kuyruğunuz varsa, SALES kuyruğu için belirli bir küme yaratmaya karar verebilirsiniz. Küme adı için olası bir adlandırma kuralı, bu tür kümeleri Q . *QueueName* olarak adlandırmaktır; örneğin, Q . SALES. Yalıtılacak çok sayıda kuyruğunuz varsa daha pratik olabilecek alternatif bir yaklaşım, gerektiğinde ve yerde yalıtılmış kuyruk kümeleri oluşturmaktır. Küme adları QUEUES . *not* olabilir.

Örnekte, yeni kümeye Q . SALES adı verilir. Yeni kümeyi eklemek için Yeni bir kümede satış kuyruğunu yalıtma ve ağ geçidi kümesi iletim kuyruklarını ayırma değişiklikleri başlıklı konudaki tanımlara bakın. Tanımlama değişikliklerinin özeti aşağıdaki gibidir:

- i) Havuz kuyruğu yöneticilerindeki kümelerin ad listesine Q . SALES ekleyin. Ad listesi, **REPOSNL** kuyruk yöneticisi değiştirilmesinde belirtilir.
- ii) Ağ geçidi kuyruk yöneticisindeki kümelerin ad listesine Q . SALES ekleyin. Ad listesi, ağ geçidi kuyruk yöneticisindeki tüm küme kuyruğu diğer adı ve küme kuyruğu yöneticisi diğer adı tanımlamalarında kullanılır.
- iii) Hem üyesi olduğu kümeler için SALESRV kuyruk yöneticisinde bir ad listesi oluşturun, hem de SALES kuyruğunun küme üyeliğini değiştirin:

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

SALES kuyruğu, yalnızca geçiş için her iki kümenin de bir üyesidir. Yeni yapılandırma çalıştıktan sonra, SALES kuyruğunu SALES kümesinden kaldırılıyorsunuz; bkz. [Şekil 15 sayfa 59](#).

- b) "Sistematik bir adlandırma kuralına göre oluşturduğunuz yeni kümeler için küme gönderen ve küme alıcı kanalları oluşturun".

- i) Havuz kuyruğu yöneticilerinin her birine Q . SALES . *RepositoryQMgr* küme alıcı kanalını ekleyin
- ii) Diğer havuz yöneticisine bağlanmak için havuz kuyruğu yöneticilerinin her birine Q . SALES . *OtherRepositoryQMgr* küme gönderen kanalını ekleyin. Bu kanalları başlat.
- iii) Çalışmakta olan havuz kuyruğu yöneticilerinden birine Q . SALES . SALESRV ve Q . SALES . GATE küme alıcı kanallarını ekleyin.
- iv) Küme gönderen kanallarını Q . SALES . SALESRV ve Q . SALES . GATE SALESRV ve GATE kuyruk yöneticilerine ekleyin. Kümeyi gönderen kanalı, küme alıcı kanallarını yarattığınız havuz kuyruğu yöneticisine bağlayın.

- c) "Hedef kuyruğa ileti gönderen her kuyruk yöneticisinde her yalıtılmış hedef için bir küme iletim kuyruğu tanımlayın".

Ağ geçidi kuyruk yöneticisinde, Q . SALES . SALESRV küme gönderen kanalı için küme iletim kuyruğunu XMITQ . Q . SALES . SALESRV tanımlayın:

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

3. Üçüncü yapılandırma adımı "Yönetişim ya da izleme gereksinimlerini karşılamak için küme iletim kuyrukları oluşturun" dir.

Ağ geçidi kuyruk yöneticisinde küme iletim kuyruklarını tanımlayın:

```
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
```

Sonraki adım

Ağ geçidi kuyruk yöneticisinde yeni yapılandırmaya geçin.

Anahtar, yeni kanallar başlatılarak ve farklı iletim kuyruklarıyla ilişkili kanallar yeniden başlatılarak tetiklenir. Diğer bir seçenek olarak, ağ geçidi kuyruk yöneticisini durdurabilir ve başlatabilirsiniz.

1. Ağ geçidi kuyruk yöneticisinde aşağıdaki kanalları durdurun:

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

2. Ağ geçidi kuyruk yöneticisinde aşağıdaki kanalları başlatın:

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

Anahtar tamamlandığında, SALES kuyruğunu SALES kümesinden kaldırın; bkz. [Şekil 15 sayfa 59](#).

İlgili kavramlar

Hangi küme iletim kuyruğunun kullanılacağını seçme

Farklı küme iletim kuyruğu konfigürasyon seçenekleri arasında seçim yapabilirsiniz.

İlgili görevler

Kümeleme: Küme iletim kuyrukları arasında geçiş yapma

Var olan bir üretim kuyruğu yöneticisinin küme iletim kuyruklarındaki değişikliklerin nasıl yürürlüğe gireceğini planlayın.

Örnek kümelerin yaratılması

Örnek kümeyi yaratmak için tanımlar ve yönergeler; SALES kuyruğunu yalıtma ve ağ geçidi kuyruk yöneticisinde iletileri ayırmak için bu kümeyi değiştirin.

Bu görev hakkında

FINANCE, SALES ve Q.SALES kümelerini oluşturmak için tam **MQSC** komutları Temel kümeler için tanımlar, Yeni bir kümede satış kuyruğunu yalıtma ve ağ geçidi kümesi iletim kuyruklarını ayırmak için yapılan değişikliklerle kuyruk yöneticisi SALESRV üzerindeki satış kuyruğunu satış kümesinden kaldırın şeklinde sağlanır. Tanımlamaları kısa tutmak için DEVELOP kümesi tanımlamalardan çıkarılır.

Yordam

1. SALES ve FINANCE kümelerini ve ağ geçidi kuyruk yöneticisini oluşturun.

- a) Kuyruk yöneticilerini yaratın.

Çizelge 4 sayfa 56 içindeki kuyruk yöneticisi adlarının her biri için `crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QmgrName` komutunu çalıştırın.

Çizelge 4. Kuyruk yöneticisi adları ve kapı numaraları		
Açıklama	Kuyruk yöneticisi adı	Kapı numarası
Finans havuzu	FINR1	1414
Finans havuzu	FINR2	1415
Finans müşterisi	FINCLT	1418
Satış havuzu	SALER1	1416
Satış havuzu	SALER2	1417

Çizelge 4. Kuyruk yöneticisi adları ve kapı numaraları (devamı var)		
Açıklama	Kuyruk yöneticisi adı	Kapı numarası
Satış sunucusu	SALESRV	1419
Ağ Geçidi	GATE	1420

b) Tüm kuyruk yöneticilerini başlat

Çizelge 4 sayfa 56 içindeki kuyruk yöneticisi adlarının her biri için `startmqm QmgrName` komutunu çalıştırın.

c) Her kuyruk yöneticisi için tanımlama yaratılması

Şu komutu çalıştırın: `runmqsc QmgrName < filename` Burada dosyalar Temel kümeler için tanımları içinde listelenir ve dosya adı kuyruk yöneticisi adıyla eşleşir.

Temel kümeler için tanımlar

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. Örnek istek programını çalıştırarak yapılandırılmayı test edin.

a) Tetikleyici izleme programını SALESRV kuyruk yöneticisinde başlat

Windows' da bir komut penceresi açın ve `runmqtrm -m SALESRV` komutunu çalıştırın.

b) Örnek istek programını çalıştırın ve bir istek gönderin.

Windows' da bir komut penceresi açın ve `amqsreq A.SALES FINCLT` komutunu çalıştırın.

İstek iletişi geri yansıtılır ve 15 saniye sonra örnek program tamamlanır.

3. Q.SALES kümesindeki SALES kuyruğunu yalıtma ve ağ geçidi kuyruk yöneticisindeki SALES ve FINANCE kümesi için ayrı küme iletileri oluşturmak üzere tanımlamaları yaratın.

Şu komutu çalıştırın: `runmqsc QmgrName <filename>` Burada dosyalar aşağıdaki listede listelenir ve dosya adı neredeyse kuyruk yöneticisi adıyla eşleşir.

Yeni bir kümede satış kuyruğunu yalıtma ve ağ geçidi kümesi iletim kuyruklarını ayırma değişiklikleri

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)
```

chgate.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. SALES kuyruğunu SALES kümesinden kaldırın.

Şekil 15 sayfa 59’ünde **MQSC** komutunu çalıştırın:

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

Şekil 15. SALESRV kuyruk yöneticisindeki satış kuyruğunu satış kümesinden kaldır

5. Kanalları yeni iletim kuyruklarına çevirin.

Bu gereksinme, GATE kuyruk yöneticisinin kullandığı tüm kanalları durdurup başlatmaktır. Bunu en az sayıda komutla yapmak için kuyruk yöneticisini durdurun ve başlatın.

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

Sonraki adım

1. Yeni yapılandırmanın çalıştığını doğrulamak için örnek istek programını yeniden çalıştırın; bkz. adım [“2” sayfa 58](#)
2. GATE kuyruk yöneticisindeki tüm küme iletim kuyruklarından akan iletileri izleyin:
 - a. Kuyruk izlemesini açmak için küme iletim kuyruklarının her birinin tanımını değiştirin.

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

- b. Çıkışı simge durumuna küçültmek için kuyruk yöneticisi istatistiklerini izleme OFF olup olmadığını denetleyin ve birden çok sınıma gerçekleştirmek için izleme aralığını daha düşük bir değere ayarlayın.

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

- c. GATE kuyruk yöneticisini yeniden başlatın.
- d. SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV ve SYSTEM.CLUSTER.TRANSMIT.QUEUE’den eşit sayıda ileti aktığını doğrulamak için örnek istek programını birkaç kez çalıştırın. İstekler SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV üzerinden akar ve SYSTEM.CLUSTER.TRANSMIT.QUEUE aracılığıyla yanıt alır.

```
amqsmn -m GATE -t statistics
```

e. Birkaç aralıkla ilgili sonuçlar aşağıdaki gibidir:

```
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.
```

Birinci aralıkta bir, ikinci aralıkta iki istek ve yanıt iletisi gönderildi. İstek iletilerinin SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRVüzerine ve yanıt iletilerinin SYSTEM.CLUSTER.TRANSMIT.QUEUEüzerine yerleştirildiği çıkarımı yapabilirsiniz.

Kümeleme: Küme iletim kuyrukları arasında geçiş yapma

Var olan bir üretim kuyruğu yöneticisinin küme iletim kuyruklarındaki değişikliklerin nasıl yürürlüğe gireceğini planlayın.

Başlamadan önce

Geçiş işleminin yeni iletim kuyruğuna aktarması gereken ileti sayısını azaltırsanız, geçiş işlemi daha hızlı tamamlanır. Devam etmeden önce iletim kuyruğunu boşaltmaya çalışma nedenlerinden ötürü [Küme gönderen kanalı farklı bir iletim kuyruğuna aktarma işleminin nasıl çalıştığını](#) okuyun.

Bu görev hakkında

Küme iletim kuyruklarında yapılan değişikliklerin yürürlüğe girmesinin iki yolu vardır.

1. Kuyruk yöneticisinin değişiklikleri otomatik olarak yapmasına izin verin. Bu varsayılandır. Kuyruk yöneticisi, bir sonraki küme gönderen kanalı başlatıldığında, bekleyen iletim kuyruğu olan küme gönderen kanallarını değiştirir.
2. Değişiklikleri el ile yapın. Bir kümeyi gönderen kanal durdurulduğunda bu kanalda değişiklik yapabilirsiniz. Kümeyi gönderen kanal başlatılmadan önce bir küme iletim kuyruğundan diğerine geçebilirsiniz.

İki seçenektan hangisini seçeceğinize karar verirken hangi etkenleri göz önünde bulunduruyorsunuz ve anahtar nasıl yönetiyorsunuz?

Yordam

- Seçenek 1: Kuyruk yöneticisinin değişiklikleri otomatik olarak yapmasını sağlayabilirsiniz; bkz. [“Etkin küme-gönderen kanallarının başka bir küme iletim kuyrukları kümesine değiştirilmesi” sayfa 62.](#)

Kuyruk yöneticisinin sizin için geçiş yapmasını istiyorsanız bu seçeneği belirleyin.

Bu seçeneği tanımlamanın diğer bir yolu, kuyruk yöneticisinin kanalı durdurmaya zorlamadan bir küme gönderen kanalı değiştirmesini sağlamaktır. Kanalı durdurmaya ve daha sonra kanalı başlatmaya zorlama seçeneğiniz vardır. Anahtar, kanal başlatıldığında başlar ve kanal çalışırken çalışır, bu da seçenek 2 'den farklıdır. Seçenek 2 'de, kanal durdurulduğunda anahtar gerçekleşir.

Anahtarın otomatik olarak gerçekleşmesine izin vererek bu seçeneği belirlerseniz, bir küme gönderen kanalı başlatıldığında geçiş işlemi başlar. Kanal durdurulmazsa, işlenecek bir ileti varsa, devre dışı kaldıktan sonra başlar. Kanal durdurulduysa, START CHANNEL komutuyla başlatın. Kanal hizmet vermekte olduğu iletim kuyruğunda küme gönderen kanal için ileti kalmadığı anda anahtar işlemi tamamlanır. Bu durumda, kümeyi gönderen kanal için yeni gelen iletiler doğrudan

yeni iletim kuyruğunda saklanır. O zamana kadar, iletiler eski iletim kuyruğunda saklanır ve geçiş işlemi, iletileri eski iletim kuyruğundan yeni iletim kuyruğuna aktarır. Kümeyi gönderen kanal, tüm geçiş işlemi sırasında yeni küme iletim kuyruğundan iletileri iletir.

Anahtar işleminin ne zaman tamamlandığı sistemin durumuna bağlıdır. Değişiklikleri bir bakım penceresinde yapıyorsanız, geçiş sürecinin zamanında tamamlanıp tamamlanmayacağını önceden değerlendirin. Zaman içinde tamamlanıp tamamlanmayacağı, eski iletim kuyruğundan aktarım bekleyen iletilerin sayısının sifıra ulaşip ulaşmamasına bağlıdır.

İlk yöntemin avantajı otomatik olması. Bir dezavantajı, yapılandırma değişikliklerini yapma süresi bir bakım penceresiyle sınırlıysa, bakım penceresi içindeki anahtar işlemini tamamlamak için sistemi denetleyebileceğinizden emin olmanız gerekir. Emin olamıyorsanız, seçenek 2 daha iyi bir seçim olabilir.

- Seçenek 2: Değişiklikleri el ile yapın; bkz. [“Durdurulmuş bir küme gönderen kanalının başka bir küme iletim kuyruğuna değiştirilmesi” sayfa 63.](#)

Tüm geçiş işlemi el ile denetlemek ya da durdurulan ya da etkin olmayan bir kanal arasında geçiş yapmak istiyorsanız bu seçeneği belirleyin. Birkaç küme gönderen kanalını değiştiriyorsanız ve anahtarı bir bakım penceresi sırasında yapmak istiyorsanız, bu iyi bir seçimdir.

Bu seçeneğin diğer bir açıklaması, küme gönderen kanalı durdurulurken, kümeyi gönderen kanalı değiştirdiğinizi söylemektir.

Bu seçeneği belirlerseniz, anahtarın ne zaman gerçekleşeceğini tam olarak denetlemeniz gerekir. Değiştirme işlemi bir bakım zaman aralığı içinde sabit bir süre içinde tamamlama konusunda emin olun. Anahtarın süresi, bir iletim kuyruğundan diğerine kaç iletinin aktarılması gerektiğine bağlıdır. İletiler gelmeye devam ederse, işlemin tüm iletileri aktarması zaman alabilir. Eski iletim kuyruğundan ileti aktarmadan kanalı değiştirme seçeneğiniz vardır. Anahtar "anında". Kümeyi gönderen kanalı yeniden başlattığınızda, yeni atadığınız iletim kuyruğundaki iletileri işlemeye başlar.

İkinci yöntemin avantajı, geçiş süreci üzerinde denetime sahip olmaktır. Dezavantajı, değiştirilecek küme gönderen kanallarını tanımlamanız, gerekli komutları çalıştırmanız ve küme gönderen kanalının durmasını önleyebilecek belirsiz kanalları çözmeniz gerekmesi.

İlgili kavramlar

[Hangi küme iletim kuyruğunun kullanılacağını seçme](#)

Farklı küme iletim kuyruğu konfigürasyon seçenekleri arasında seçim yapabilirsiniz.

[Kümeyi gönderen kanalını farklı bir iletim kuyruğuna aktarma işlemi nasıl çalışır?](#)

İlgili görevler

[Kümeleme: Birden çok küme iletim kuyruğunun konfigürasyonu örneği](#)

Bu görevde, birden çok küme iletim kuyruğunun çakışan üç kümeye planlanması için gereken adımları uygulayabilirsiniz. Gereksinimler, ileti akışlarını bir küme kuyruğuna, diğer tüm ileti akışlarından ayırmayı ve farklı küme iletim kuyruklarındaki farklı kümelere ilişkin iletileri saklamayı içerir.

Etkin küme-gönderen kanallarının başka bir küme iletim kuyrukları kümesine değiştirilmesi

Bu görev, etkin küme gönderen kanallarını değiştirmek için size üç seçenek sunar. Bir seçenek, kuyruk yöneticisinin anahtarı otomatik olarak yapmasına izin vermektir; bu, çalışan uygulamaları etkilemez. Diğer seçenekler, kanalları el ile durdurup başlatmanızı ya da kuyruk yöneticisini yeniden başlatmanızı sağlar.

Başlamadan önce

Küme iletim kuyruğu konfigürasyonunu değiştirin. **DEFCLXQ** kuyruk yöneticisi özniteliğini değiştirebilir ya da iletim kuyruklarının **CLCHNAME** özniteliğini ekleyebilir ya da değiştirebilirsiniz.

Geçiş işleminin yeni iletim kuyruğuna aktarması gereken ileti sayısını azaltırsanız, geçiş işlemi daha hızlı tamamlanır. Devam etmeden önce iletim kuyruğunu boşaltmaya çalışma nedenlerinden ötürü [Küme gönderen kanalını farklı bir iletim kuyruğuna aktarma işleminin nasıl çalıştığını](#) okuyun.

Bu görev hakkında

Küme-iletim kuyruğu konfigürasyon değişiklikleri yapmak için kendi planınızı gerçekleştirmek üzere temel olarak görevdeki adımları kullanın.

Yordam

1. İsteğe bağlı: Geçerli kanal durumunu kaydet

Küme iletim kuyruklarına hizmet veren yürürlükteki ve saklanan kanalların durumunu kaydedin. Aşağıdaki komutlar, sistem kümesi iletim kuyruklarıyla ilişkili durumu görüntüler. Tanımladığınız küme iletim kuyruklarıyla ilişkili durumu görüntülemek için kendi komutlarınızı ekleyin. Bu iletim kuyruklarına ilişkin kanal durumunun görüntülenmesini kolaylaştırmak üzere tanımladığınız küme iletim kuyruklarını adlamak için XMITQ. *ChannelName* gibi bir kural kullanın.

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

2. İletim kuyruklarını değiştirin.

- Hiçbir şey yapma. Kuyruk yöneticisi, durdurulduktan ya da devre dışı bırakıldıktan sonra yeniden başlatıldığında küme gönderen kanallarını değiştirir.

Bir kuyruk yöneticisi yapılanışının değiştirilmesine ilişkin herhangi bir kural ya da endişeniz yoksa bu seçeneği belirleyin. Çalışan uygulamalar değişikliklerden etkilenmez.

- Kuyruk yöneticisini yeniden başlatın. Tüm küme-gönderen kanalları istek üzerine otomatik olarak durdurulur ve yeniden başlatılır.

Tüm değişiklikleri hemen başlatmak için bu seçeneği belirleyin. Çalışan uygulamalar, kapatılıp yeniden başlatılırken kuyruk yöneticisi tarafından kesilir.

- Tek tek küme gönderen kanallarını durdurun ve yeniden başlatın.

Birkaç kanalı hemen değiştirmek için bu seçeneği belirleyin. Uygulamaların çalıştırılması, ileti kanalınızın durdurulması ve yeniden başlatılması arasında ileti aktarımında kısa bir gecikmeyle karşılaşır. Kümeyi gönderen kanal, durdurmanız dışında çalışmaya devam eder. Geçiş işlemi sırasında iletiler eski iletim kuyruğuna teslim edilir, geçiş işlemi tarafından yeni iletim kuyruğuna aktarılır ve küme gönderen kanalı tarafından yeni iletim kuyruğundan iletilir.

3. İsteğe bağlı: Kanallar değişirken kanalları izle

Anahtar sırasında kanal durumunu ve iletim kuyruğu derinliğini görüntüler. Aşağıdaki örnek, sistem kümesi iletim kuyruklarının durumunu görüntüler.

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

4. İsteğe bağlı: AMQ7341 Kanal için iletim kuyruğu *ChannelName* kuyruktan *QueueName* kuyruğundan *QueueName* kuyruk yöneticisi hata günlüğüne yazılan iletilere bakın.

Durdurulmuş bir küme gönderen kanalının başka bir küme iletim kuyruğuna değiştirilmesi

El ile değişiklik yapmayı seçerseniz, küme gönderen kanalı durdurulduğunda kanalda değişiklik yapar ve küme gönderen kanalı başlatılmadan önce bir küme iletim kuyruğundan diğerine geçersiniz.

Başlamadan önce

Bazı yapılandırma değişiklikleri yapabilir ve şimdi etkilenen küme gönderen kanallarını başlatmadan bunları geçerli kılmak isteyebilirsiniz. Diğer bir seçenek olarak, gerek duyduğunuz yapılandırma değişikliklerini görevdeki adımlardan biri olarak yaparsınız.

Geçiş işleminin yeni iletim kuyruğuna aktarması gereken ileti sayısını azaltırsanız, geçiş işlemi daha hızlı tamamlanır. Devam etmeden önce iletim kuyruğunu boşaltmaya çalışma nedenlerinden ötürü [Küme gönderen kanalı](#) farklı bir iletim kuyruğuna aktarma işleminin nasıl çalıştığını okuyun.

Bu görev hakkında

Bu görev, durdurulan ya da etkin olmayan küme gönderen kanallar tarafından sunulan iletim kuyruklarını değiştirir. Bu görevi, bir küme gönderen kanalı durdurulduğu ve iletim kuyruğunu hemen değiştirmek istediğiniz için yapabilirsiniz. Örneğin, bir nedenden ötürü bir küme gönderen kanalı başlatılmıyor ya da başka bir yapılandırma sorunu var. Sorunu çözmek için bir küme gönderen kanalı yaratmaya ve eski küme gönderen kanalına ilişkin iletim kuyruğunu tanımladığınız yeni küme gönderen kanalla ilişkilendirmeye karar verdiniz.

Daha olası bir senaryo, küme iletim kuyruklarının yeniden konfigürasyonunun ne zaman gerçekleştirileceğini denetlemek istemeniz olabilir. Yeniden yapılandırmayı tam olarak denetlemek için, kanalları durdurur, yapılandırmayı değiştirir ve iletim kuyruklarını değiştirirsiniz.

Yordam

1. Değiştirmek istediğiniz kanalları durdurun

a) Değiştirmek istediğiniz çalışan ya da etkin olmayan kanalları durdurun. Etkin olmayan bir küme-gönderen kanalının durdurulması, siz yapılandırma değişiklikleri yaparken başlatılmasını önler.

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

2. İsteğe bağlı: Yapılandırma değişikliklerini yapın.

Örneğin, bkz. [“Kümeleme: Birden çok küme iletim kuyruğunun konfigürasyonu örneği” sayfa 53.](#)

3. Kümeyi gönderen kanalları yeni küme iletim kuyruklarına çevirin.

Multi Çoklu platformlar' da şu komutu verin:

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

z/OS z/OS' ta, iletileri değiştirmek ya da neler olduğunu izlemek için CSQUTIL komutunun SWITCH işlevini kullanın. Aşağıdaki komutu kullanın.

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

Daha fazla bilgi için bkz. [SWITCH işlevi](#).

runswchl ya da CSQUTIL SWITCH komutu, eski iletim kuyruğundaki iletileri yeni iletim kuyruğuna aktarır. Bu kanal için eski iletim kuyruğundaki ileti sayısı sifıra ulaştığında, anahtar tamamlanır. Komut zamanuyumludur. Komut, geçiş işlemi sırasında aşama iletilerini pencereye yazar.

Aktarım aşamasında, küme-gönderen kanalına gönderilen yeni iletiler yeni iletim kuyruğuna aktarılır.

Kümeyi gönderen kanal durdurulduğundan, iletiler yeni iletim kuyruğunda birikir. Durdurulan küme gönderen kanalı [“Etkin küme-gönderen kanallarının başka bir küme iletim kuyrukları kümesine değiştirilmesi” sayfa 62](#) içindeki [“2” sayfa 63](#) adımına karşılaştırın. Bu adımda, küme gönderen kanalı çalışır durumda olduğundan, iletiler yeni iletim kuyruğunda birikmez.

4. İsteğe bağlı: Kanallar değişirken kanalları izle

Farklı bir komut penceresinde, anahtar sırasında iletim kuyruğu derinliğini görüntüleyin. Aşağıdaki örnek, sistem kümesi iletim kuyruklarının durumunu görüntüler.

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


5. İsteğe bağlı: AMQ7341 Kanal için iletim kuyruğu *ChannelName* kuyruktan *QueueName* kuyruğundan *QueueName* kuyruk yöneticisi hata günlüğüne yazılan iletilere bakın.
6. Durdurmuş olduğunuz küme gönderen kanallarını yeniden başlatın.

Kanallar, siz onları durdurduğunuzda otomatik olarak başlamaz ve bunları DURDURULDU durumuna yerleştirir.

```
START CHANNEL(ChannelName)
```

İlgili başvurular

[Runswchl](#)

[KANAL ÇÖZÜMLE](#)

[BAĞLANTI DURDUR](#)

Kümeleme: Geçiş ve değişiklik ile ilgili en iyi uygulamalar

Bu konuda, IBM MQ kümelerini planlamaya ve yönetmeye ilişkin yönergeler sağlanır. Bu bilgiler, müşterilerden gelen testlere ve geribildirime dayalı bir kılavuздur.

1. “Kümedeki nesnelerin taşınması” sayfa 65 (Herhangi bir düzeltme paketi ya da yeni IBM MQ sürümü kurmadan, nesneleri bir küme içinde taşımaya ilişkin en iyi uygulamalar).
2. “Büyütmeler ve bakım kuruluşları” sayfa 66 (Bakım ya da büyüme uygularken ve yeni mimariyi test ederken bir çalışma kümesi mimarisinin çalışır durumda tutulmasına ilişkin en iyi uygulamalar).

Kümedeki nesnelerin taşınması

Uygulamalar ve kuyrukları

Bir kuyruk yöneticisinde bulunan bir kuyruk eşgörünümünü başka bir kuyruk yöneticisinde barındırılacak şekilde taşımanız gerekiyorsa, sorunsuz bir geçiş sağlamak için iş yükü dengeleme parametreleriyle çalışabilirsiniz.

Yeni barındırılacağı bir kuyruk eşgörünümü yaratın, ancak uygulamanız geçiş yapmaya hazır oluncaya kadar özgün yönetim ortamına ileti göndermeye devam etmek için küme iş yükü dengeleme ayarlarını kullanın. Bu, aşağıdaki adımlarla gerçekleştirilir:

1. Var olan kuyruğun **CLWLRANK** özelliğini yüksek bir değere (örneğin, beş) ayarlayın.
2. Kuyruğun yeni eşgörünümünü yaratın ve **CLWLRANK** özelliğini sıfır olarak ayarlayın.
3. Yeni sistemin daha fazla yapılandırmasını tamamlayın; örneğin, kuyruğun yeni örneğine ilişkin uygulamaları konuşlandırın ve tüketmeye başlayın.
4. Yeni kuyruk eşgörünümünün **CLWLRANK** özelliğini özgün eşgörünümünden (örneğin, dokuz) yüksek olacak şekilde ayarlayın.
5. Özgün kuyruk eşgörünümünün sistemdeki kuyruğa alınan iletileri işlemesine ve sonra kuyruğu silmesine izin verin.

Tüm kuyruk yöneticilerini taşıma

Kuyruk yöneticisi aynı anasistemde kalıyorsa, ancak IP adresi değişiyorsa, işlem aşağıdaki gibidir:

- DNS, doğru kullanıldığında, işlemin basitleştirilmesine yardımcı olabilir. [Connection name \(CONNNAME\)](#) kanal özniteliğini ayarlayarak DNS kullanımı hakkında bilgi için bkz. [ALTER CHANNEL](#).
- Bir tam havuzu taşıyorsanız, değişiklik yapmadan önce sorunsuz çalışan (örneğin, kanal durumuyla ilgili sorun yok) en az bir tam havuzunuz olduğundan emin olun.
- Trafik oluşmasını önlemek için [SUSPEND QMGR](#) komutunu kullanarak kuyruk yöneticisini askıya alın.
- Bilgisayarın IP adresini değiştirin. CLUSRCVR kanal tanımınız CONNAME alanında bir IP adresi kullanıyorsa, bu IP adresi girişini değiştirin. Güncellemelerin her yerde bulunduğundan emin olmak için DNS önbelleğinin temizlenmesi gerekebilir.
- Kuyruk yöneticisi tam havuzlara yeniden bağlandığında, kanal otomatik tanımlamaları otomatik olarak çözülür.

- Kuyruk yöneticisi tam bir havuz barındırıyorsa ve IP adresi değişiyorsa, bölümlerin el ile tanımlanan CLUSSDR kanallarını yeni konuma yönlendirmek için en kısa zamanda değiştirilmesini sağlamak önemlidir. Bu anahtar gerçekleştirilinceye kadar, bu kuyruk yöneticileri yalnızca geri kalan (değiştirilmemiş) tam havuzla iletişim kurabilir ve yanlış kanal tanımlamasıyla ilgili uyarı iletileri görülebilir.
- **RESUME QMGR** komutunu kullanarak kuyruk yöneticisini sürdürün.

Kuyruk yöneticisinin yeni bir anasisteme taşınması gerekiyorsa, kuyruk yöneticisi verileri kopyalanıp bir yedekten geri yüklenebilir. Ancak, başka seçenek yoksa, bu işlem önerilmez; yeni bir makinede kuyruk yöneticisi yaratılması ve kuyrukların ve uygulamaların önceki bölümde açıklandığı gibi eşlenmesi daha iyi olabilir. Bu durum, düzgün bir baştan başlama/geri alma mekanizması sağlar.

Tam bir kuyruk yöneticisini yedekleme kullanarak taşımaya karar verdiyseniz, aşağıdaki en iyi uygulamaları izleyin:

- Tüm işlemi, genellikle sistem kurtarma için kullanacağınız işlemleri işletim sistemi ortamınız için uygun şekilde uygulayarak yedekten bir kuyruk yöneticisi geri yüklemesi olarak kabul edin.
- Yerel olarak tutulan tüm küme bilgilerini atmak (şüpheli olan otomatik tanımlı kanallar da içinde olmak üzere) ve yeniden oluşturulmaya zorlamak için geçişten sonra **REFRESH CLUSTER** komutunu kullanın.

Not: Büyük kümeler için, **REFRESH CLUSTER** komutunun kullanılması, devam ederken kümede kesintiye neden olabilir ve bundan sonra küme nesnelere otomatik olarak ilgili tüm kuyruk yöneticilerine durum güncellemelerini gönderdiğinde, 27 günlük aralıklarla kümede kesintiye neden olabilir. Bkz. [Büyük bir kümede yenilenmesi kümenin performansını ve kullanılabilirliğini etkileyebilir.](#)

Bir kuyruk yöneticisi oluştururken ve kümede var olan bir kuyruk yöneticisinden kuruluşu eşlerken (bu konuda daha önce açıklandığı gibi), iki farklı kuyruk yöneticisini hiçbir zaman aynı olarak kabul etme. Özellikle, yeni bir kuyruk yöneticisine aynı kuyruk yöneticisi adını ve IP adresini vermeyin. Yeni bir kuyruk yöneticisinde 'bırakma' girişimi, IBM MQ kümelerindeki sorunların sık sık nedenidir. Önbellek, **QMID** özneliği de içinde olmak üzere güncellemeleri almayı bekler ve durum bozulabilir.

Yanlışlıkla aynı adla iki farklı kuyruk yöneticisi oluşturulursa, yanlış girdiyi kümeden çıkarmak için **RESET CLUSTER QMID** komutunu kullanmanız önerilir.

Büyütmeler ve bakım kuruluşları

Büyük patlama senaryosundan kaçının (örneğin, tüm küme ve kuyruk yöneticisi etkinliğini durdurma, tüm yükseltmeleri ve bakımı tüm kuyruk yöneticilerine uygulama, ardından her şeyi aynı anda başlatma). Kümeler, kuyruk yöneticisinin birden çok sürümüyle birlikte çalışacak şekilde tasarlanmıştır, bu nedenle iyi planlanmış, aşamalı bir bakım yaklaşımı önerilir.

Yedek bir planınız var:

- Yedek aldın mı?
- Yeni küme işlevini hemen kullanmaktan kaçının: Tüm kuyruk yöneticilerinin yeni düzeye yükseltildiğinden emin oluncaya kadar bekleyin ve bunların hiçbirini geri almayacağınızdan emin olun. Bazı kuyruk yöneticilerinin daha önceki bir düzeyde olduğu bir kümede yeni küme işlevinin kullanılması, tanımlanmamış bir davranışa neden olabilir.

Havuz, kendi sürümünde aldığı bir kaydı saklar. Aldığı kayıt daha sonraki bir sürümdeyse, kayıt saklandığında sonraki sürüm öznelikleri atılır. Bir IBM MQ 9.3 kuyruk yöneticisine ilişkin bilgi alan bir IBM MQ 9.2 kuyruk yöneticisi yalnızca IBM MQ 9.2 bilgilerini saklar. IBM MQ 9.2 kaydı alan bir IBM MQ 9.3 havuzu, sonraki sürümde tanıtılan öznelıklar için varsayılan değerleri saklar. Varsayılan değerler, aldığı kayda dahil edilmeyen öznelıklere ilişkin değerleri tanımlar.

Önce tam havuzları geçirin. Anlamadıkları bilgileri iletebilseler de, bunu sürdüremezler, bu nedenle kesinlikle gerekli olmadıkça önerilen yaklaşım bu değildir. Daha fazla bilgi için bkz. [Kuyruk yöneticisi kümesi geçişi](#).

Kümeleme: REFRESH CLUSTER en iyi uygulamalarını kullanma

Bir kümeyle ilgili yerel olarak tutulan tüm bilgileri atmak ve kümedeki tüm havuzlardan bu bilgileri yeniden oluşturmak için **REFRESH CLUSTER** komutunu kullanabilirsiniz. Kural dışı durumlar dışında bu komutu kullanmanıza gerek yoktur. Bunu kullanmanız gerekiyorsa, nasıl kullanacağınıza ilişkin özel konular vardır. Bu bilgiler, müşterilerden gelen testlere ve geribildirime dayalı bir kılavuzdur.

REFRESH CLUSTER komutunu yalnızca gerçekten yapmanız gerekiyorsa çalıştırın

IBM MQ küme teknolojisi, kümelenmiş bir kuyrukta değişiklik gibi küme yapılandırmasında herhangi bir değişikliğin, bilgileri bilmesi gereken kümenin herhangi bir üyesi tarafından otomatik olarak tanınmasını sağlar. Bu bilgi yayılımını gerçekleştirmek için daha fazla yönetim adımının atılması gerekmez.

Bu tür bilgiler, gerektiğinde kümedeki kuyruk yöneticilerine ulaşmazsa, örneğin, bir uygulama kümeyi ilk kez açmaya çalıştığında kümedeki başka bir kuyruk yöneticisi tarafından kümelenmiş bir kuyruk tanınmazsa, bu, küme altyapısında bir sorun olduğu anlamına gelir. Örneğin, bir kuyruk yöneticisi ile tam havuz kuyruk yöneticisi arasında bir kanal başlatılamıyor olabilir. Bu nedenle, tutarsızlıkların gözlemlendiği her durum araştırılmalıdır. Olanaklıysa, durumu **REFRESH CLUSTER** komutunu kullanmadan çözün.

Bu ürün belgelerinin başka yerlerinde belgelenen nadir durumlarda ya da IBM desteği tarafından istendiğinde, bir kümeyle ilişkin yerel olarak tutulan tüm bilgileri atmak ve kümedeki tüm havuzlardan bu bilgileri yeniden oluşturmak için **REFRESH CLUSTER** komutunu kullanabilirsiniz.

Büyük bir kümede yenileme, kümenin performansını ve kullanılabilirliğini etkileyebilir

REFRESH CLUSTER komutunun kullanımı, devam ederken kümede kesintiye neden olabilir; örneğin, kuyruk yöneticisi küme kaynaklarının yeniden yayılmasını işlerken tam havuzlar için ani bir artış yaratarak. Büyük bir kümede (yani yüzlerce kuyruk yöneticisinde) yenileme yapıyorsanız, mümkünse günlük işte komutu kullanmaktan kaçınmalı ve belirli tutarsızlıkları düzeltmek için alternatif yöntemler kullanmalısınız. Örneğin, bir küme kuyruğu kümede doğru şekilde yayılmıyorsa, küme tanımının değiştirilmesi gibi, kümelenmiş kuyruk tanımlamasının güncellenmesine ilişkin ilk araştırma tekniği, kuyruk yapılandırmasını kümede yeniden yayacaktır. Bu işlem, sorunun belirlenmesine ve geçici bir tutarsızlığın çözülmesine yardımcı olabilir.

Alternatif yöntemler kullanılmıyorsa ve **REFRESH CLUSTER** 'ı büyük bir kümede çalıştırmanız gerekirse, bunu kullanıcı iş yükleri üzerindeki etkiyi önlemek için yoğun olmayan zamanlarda ya da bakım aralığı sırasında yapmalısınız. Ayrıca, tek bir toplu işte büyük bir kümenin yenilenmesini önlemeli ve bunun yerine etkinliği "[Küme nesnelere otomatik güncellemeler gönderildiğinde performans ve kullanılabilirlik sorunlarını önleyin](#)" sayfa 67'inde açıkladığı gibi aşamalandırmalısınız.

Küme nesnelere otomatik güncellemeler gönderildiğinde performans ve kullanılabilirlik sorunlarını önleyin

Bir kuyruk yöneticisinde yeni bir küme nesnesi tanımlandıktan sonra, tanımlama zamanından itibaren her 27 günde bir bu nesne için bir güncelleme oluşturulur ve kümedeki her tam havuza gönderilir ve diğer ilgili kuyruk yöneticilerine gönderilir. **REFRESH CLUSTER** komutunu bir kuyruk yöneticisine verdiğinizde, belirtilen kümede yerel olarak tanımlanan tüm nesnelere bu otomatik güncelleme için saati sıfırlayın.

Büyük bir kümeyi (yani, yüzlerce kuyruk yöneticisi) tek bir kümede ya da bir sistemi yapılandırma yedeklemesinden yeniden yaratma gibi başka durumlarda, 27 gün sonra tüm bu kuyruk yöneticileri tüm nesne tanımlarını aynı anda tam havuzlara yeniden yayınlacak. Bu, tüm güncellemeler tamamlanıncaya kadar sistemin çok daha yavaş çalışmasına ya da kullanılamaz duruma gelmesine neden olabilir. Bu nedenle, büyük bir kümede birden çok kuyruk yöneticisi yenilenmez ya da yeniden oluşturmanız gerekirken, etkinliği birkaç saat ya da birkaç gün boyunca aşamalandırmanız gerekir; böylece, sonraki otomatik güncellemeler sistem performansını düzenli olarak etkilemez.

Sistem kümesi geçmiş kuyruğu

Bir **REFRESH CLUSTER** işlemi gerçekleştirildiğinde, kuyruk yöneticisi yenilemeden önce küme durumunun anlık görünümünü alır ve kuyruk yöneticisinde tanımlanmışsa, SYSTEM.CLUSTER.HISTORY.QUEUE (SCHQ) üzerinde saklar. Bu anlık görüntü, yalnızca sistemle ilgili daha sonra ortaya çıkabilecek sorunlar için IBM hizmet amaçlıdır.

SCHQ, başlangıçta dağıtılmış kuyruk yöneticilerine varsayılan olarak tanımlanır. z/OS geçişi için SCHQ el ile tanımlanmalıdır.

SCHQ 'daki iletilerin süresi üç ay sonra sona erer.

İlgili kavramlar

[“Yayınlama/abone olma kümeleri için REFRESH CLUSTER ile ilgili önemli noktalar” sayfa 102](#)

REFRESH CLUSTER komutunun verilmesi, kuyruk yöneticisinin küme konuları ve ilişkili yetkili sunucu abonelikleri de içinde olmak üzere, bir kümeyle ilgili yerel olarak tutulan bilgileri geçici olarak atmasına neden olur.

İlgili başvurular

[REFRESH CLUSTER çalıştırılırken uygulama sorunları görüldü](#)

[MQSC Komutları başvurusu: REFRESH CLUSTER](#)

Kümeleme: Kullanılabilirlik, çok eşgörünümlü ve olağanüstü durumdan kurtarma

Bu konuda, IBM MQ kümelerini planlamaya ve yönetmeye ilişkin yönergeler sağlanır. Bu bilgiler, müşterilerden gelen testlere ve geribildirime dayalı bir kılavuzdur.

IBM MQ Kümelemenin kendisi bir Yüksek Kullanılabilirlik çözümü değildir, ancak bazı durumlarda IBM MQ kullanılarak hizmetlerin kullanılabilirliğini artırmak için kullanılabilir; örneğin, farklı kuyruk yöneticilerindeki bir kuyruğun birden çok eşgörünümlü kullanılarak. Bu bölümde, IBM MQ altyapısının bu tür bir mimariye kullanılabilmesi için mümkün olan en yüksek düzeyde kullanılabilir olmasının sağlanmasına ilişkin yönergeler sağlanır.

Not: IBM MQ için diğer yüksek kullanılabilirlik ve olağanüstü durum kurtarma çözümleri kullanılabilir, bkz. [Yüksek kullanılabilirliğin yapılandırılması, kurtarma ve yeniden başlatma.](#)

Küme kaynaklarının kullanılabilirliği

İki tam havuzu korumak için olağan önerinin nedeni, bir tanesinin kaybının kümenin düzgün çalışması için önemli olmamasıdır. Her ikisi de kullanılamaz duruma gelse bile, kısmi havuzlar tarafından tutulan mevcut bilgi için 60 günlük bir ek süre vardır, ancak bu olayda yeni ya da önceden erişilen kaynaklar (örneğin, kuyruklar) yoktur.

Uygulama kullanılabilirliğini artırmak için kümeleri kullanma

Bir küme, kuyruğun ve uygulamanın birden çok eşgörünümlü kullanarak yüksek kullanılabilirlikli uygulamaların (örneğin, bir istek/yanıt tipi sunucu uygulaması) tasarlanmasına yardımcı olabilir. Gerekirse, kuyruk yöneticisi ya da kanal kullanılamıyorsa, öncelik öznitelikleri 'canlı' uygulamayı tercih edebilir. Bu, bir sorun ortaya çıktığında yeni iletileri işlemeye devam etmek için hızlı bir şekilde geçiş yapmak için güçlüdür.

Ancak, bir kümedeki belirli bir kuyruk yöneticisine teslim edilen iletiler yalnızca o kuyruk eşgörünümlüde tutulur ve bu kuyruk yöneticisi kurtarıncaya kadar işlenmek üzere kullanılamaz. Bu nedenle, gerçek verilerin yüksek düzeyde kullanılabilirliği için çok eşgörünümlü kuyruk yöneticileri gibi diğer teknolojileri de göz önünde bulundurmak isteyebilirsiniz.

Çok eşgörünümlü kuyruk yöneticileri


Yazılım Yüksek Düzeyde Kullanılabilirlik (çok eşgörünümlü), var olan iletilerinizi kullanılabilir tutmak için yerleşik bir olanaktır. Daha fazla bilgi için [IBM MQ ürününü yüksek kullanılabilirlik yapılandırmalarıyla kullanma](#), [Çok eşgörünümlü bir kuyruk yöneticisi yaratılması](#) ve aşağıdaki bölüme bakın. Kümedeki tüm kuyruk yöneticileri en az IBM WebSphere MQ 7.0.1 çalıştırıyorsa, kümedeki herhangi bir kuyruk yöneticisi bu teknik kullanılarak yüksek kullanılabilirlikli kılınabilir. Kümedeki herhangi bir kuyruk yöneticisi önceki düzeylerdeyse, ikincil bir IP 'ye geçiş yaparlarsa çok eşgörünümlü kuyruk yöneticileriyle bağlantıyı kaybedebilirler.

Bu konuda daha önce de tartışıldığı gibi, iki tam havuz yapılandırıldığı sürece, bunlar neredeyse doğaları gereği yüksek kullanılabilirliğe sahiptir. Gerekirse, IBM MQ yazılımı High Availability

(Yüksek Kullanılabilirlik)/çok eşgörünümlü kuyruk yöneticileri tam havuzlar için kullanılabilir. Bu yöntemleri kullanmak için güçlü bir neden yoktur ve aslında geçici kesintiler için bu yöntemler, yedek sisteme geçiş sırasında ek performans maliyetine neden olabilir. Örneğin, tek bir kanal kesintisi durumunda mutlaka yedek sisteme geçiş yapmayacağından, ancak kısmi havuzların küme kaynaklarını sorgulayamamasına neden olabileceğinden, iki tam havuz çalıştırmak yerine yazılım HA'nın kullanılması önerilmez.

Olağanüstü durum kurtarma

Örneğin, bir kuyruk yöneticisinin verilerini depolayan diskler bozulduğunda kurtarılması, iyi bir şekilde gerçekleştirilmesi zordur; IBM MQ yardımcı olabilir, ancak otomatik olarak gerçekleştirilemez. IBM MQ içindeki tek 'true' olağanüstü durumdan kurtarma seçeneği (herhangi bir işletim sistemi ya da diğer temel eşleme teknolojileri hariç) bir yedekten geri yüklemektir. Bu durumlarda göz önünde bulundurulması gereken bazı belirli noktalar vardır:

- Olağanüstü durumdan kurtarma senaryolarını test ederken dikkatli olun. Örneğin, yedekleme kuyruğu yöneticilerinin çalışmasını test ediyorsanız, bunları aynı anda çevrimiçi duruma getirirken, canlı kümeye yanlışlıkla katılma ve canlı küme kuyruk yöneticilerinininkiyle aynı adlandırılan kuyrukları barındırarak 'çalma' iletilerini başlatmaya dikkat edin.
- Olağanüstü durumdan kurtarma testi, çalışan bir canlı kümeye müdahale etmemelidir. Parazitten kaçınmak için teknikler şunlardır:
 - Güvenlik duvarı düzeyinde ağ ayırmayı ya da ayırmayı tamamlayın.
 -  Kanal başlatma ya da z/OS **chinit** adres alanı başlatılmıyor.
 - Gerçek bir olağanüstü durumdan kurtarma senaryosu ortaya çıkıncaya ya da çıkıncaya kadar olağanüstü durumdan kurtarma sistemine canlı TLS sertifikası verilmiyor.
- Kümedeki bir kuyruk yöneticisinin yedeği geri yüklenirken, yedek kümenin geri kalanıyla uyumsuz olabilir. **REFRESH CLUSTER** komutu güncellemeleri çözebilir ve kümeyle eşitleyebilir, ancak **REFRESH CLUSTER** komutu son çare olarak kullanılmalıdır. Bkz. “Kümeleme: REFRESH CLUSTER en iyi uygulamalarını kullanma” sayfa 67. Komutu kullanmaya başlamadan önce basit bir adımın gözden kaçırılıp kaçırılmadığını görmek için şirket içi süreç belgelerini ve IBM MQ belgelerini gözden geçirin.
- Herhangi bir kurtarma için, uygulamaların yeniden oynatmayla ve veri kaybıyla ilgilenmesi gerekir. Kuyrukların bilinen bir duruma mı, yoksa başka bir yerde yeniden gösterimleri yönetmek için yeterli bilgi olup olmadığına karar verilmelidir.

Dağıtılmış yayınlama/abone olma ağınızın planlanması

Bir kuyruk yöneticisinde yaratılan aboneliklerin, ağdaki başka bir kuyruk yöneticisine bağlı bir uygulama tarafından yayınlanan eşleşen iletileri alacağı bir kuyruk yöneticisi ağı yaratabilirsiniz. Uygun bir topoloji seçmek için el ile denetim, ağ boyutu, değişiklik sıklığı, kullanılabilirlik ve ölçeklenebilirlik gereksinimlerini göz önünde bulundurmanız gerekir.

Başlamadan önce

Bu görev, dağıtılmış yayınlama/abone olma ağlarının ne olduğunu ve bunların nasıl çalıştığını anladığınızı varsayar. Teknik bir genel bakış için bkz. [Dağıtılmış yayınlama/abone olma ağları](#).

Bu görev hakkında

Bir yayınlama/abone olma ağı için üç temel topoloji vardır:

- Doğrudan yöneltile küme
- Konu anasistem tarafından yönlendirilen küme
- Sıradüzen

İlk iki topoloji için başlangıç noktası bir IBM MQ küme yapılandırmasıdır. Üçüncü topoloji bir küme ile ya da küme olmadan oluşturulabilir. Temel kuyruk yöneticisi ağının planlanmasına ilişkin bilgi için bkz. [“Dağıtılmış kuyruklarınızın ve kümelerinizin planlanması” sayfa 19](#).

Doğrudan yöneltilen küme , bir küme zaten varken yapılandırılacak en basit topolojidir. Herhangi bir kuyruk yöneticisinde tanımladığınız herhangi bir konu, kümedeki her kuyruk yöneticisinde otomatik olarak kullanılır kılınır ve yayınlar, yayınlama uygulamasının bağlandığı herhangi bir kuyruk yöneticisinden, eşleşen aboneliklerin bulunduğu her kuyruk yöneticisine doğrudan yönlendirilir. Yapılandırmanın bu basitliği, IBM MQ ' in kümedeki her kuyruk yöneticisi arasında yüksek düzeyde bilgi paylaşımı ve bağlantırlık sağlamasına dayanır. Küçük ve basit ağlar (yani, az sayıda kuyruk yöneticisi ve oldukça statik bir yayıncı ve abone kümesi) için bu kabul edilebilir. Bununla birlikte, daha büyük veya daha dinamik ortamlarda kullanıldığında, ek yük engelleyici olabilir. Bkz. “Yayınlama/abone olma kümelerinde doğrudan yönlendirme” sayfa 75.

Konu anasistemi tarafından yönlendirilen küme , kümedeki herhangi bir kuyruk yöneticisinde tanımladığınız herhangi bir konuyu otomatik olarak kümedeki her kuyruk yöneticisinde kullanılabilir kılarak, doğrudan yöneltilen bir kümeyle aynı avantajı sağlar. Ancak, konu anasistemi tarafından yönlendirilen kümeler her konuyu barındıran kuyruk yöneticilerini dikkatli bir şekilde seçmenizi gerektirir; bu konuya ilişkin tüm bilgiler ve yayınlar bu konu anasistem kuyruk yöneticilerinden geçer. Bu, sistemin tüm kuyruk yöneticileri arasında kanalları ve bilgi akışlarını korumasına gerek olmadığı anlamına gelir. Ancak bu, yayınların artık abonelere doğrudan gönderilmeyebileceği, ancak bir konu anasistem kuyruk yöneticisi aracılığıyla yönlendirilebileceği anlamına da gelir. Bu nedenle sisteme, özellikle de konuları barındıran kuyruk yöneticilerine ek yük yüklenebilir, bu nedenle topolojinin dikkatli planlanması gerekir. Bu topoloji, özellikle birçok kuyruk yöneticisi içeren ağlar için ya da dinamik bir yayıncı ve abone kümesini barındıran ağlar (yani, sık eklenen ya da kaldırılan yayıncılar ya da aboneler) için etkilidir. Rotaların kullanılabilirliğini artırmak ve yayın iş yükünü yatay olarak ölçeklemek için ek konu anasistemleri tanımlanabilir. Bkz. “Konu Yayınlama/abone olma kümelerinde anasistem yönlendirmesi” sayfa 79.

Sıradüzeni , en el ile yapılandırmanın kurulmasını gerektirir ve değiştirilmesi en zor topolojidir. Sıradüzendeki her kuyruk yöneticisi ile doğrudan ilişkileri arasındaki ilişkileri el ile yapılandırmanız gerekir. İlişkiler yapılandırıldıktan sonra, yayınlar (önceki iki topoloji için olduğu gibi) sıradüzendeki diğer kuyruk yöneticilerindeki aboneliklere yönlendirilir. Yayınlar, sıradüzen ilişkileri kullanılarak yönlendirilir. Bu, çok özel topolojilerin farklı gereksinimlere uyacak şekilde yapılandırılmasına olanak sağlar, ancak ara kuyruk yöneticileri aracılığıyla birçok "sekme" gerektiren yayınların aboneliklere erişmesiyle de sonuçlanabilir. Her zaman bir yayın için hiyerarşi içinde yalnızca bir rota vardır, bu nedenle her kuyruk yöneticisinin kullanılabilirliği önemlidir. Sıradüzenler genellikle yalnızca tek bir kümenin yapılandırılmadığı durumlarda tercih edilir; örneğin, birden çok kuruluş yayılırken. Bkz. “Yayınlama/abone olma sıradüzenlerinde yönlendirme” sayfa 103.

Gerektiğinde, yukarıdaki üç topoloji belirli topografik gereksinimleri çözmek için birleştirilebilir. Örneğin, Birden çok kümenin konu alanlarını birleştirme başlıklı konuya bakın.

Dağıtılmış yayınlama/abone olma ağınız için uygun bir topoloji seçmek üzere aşağıdaki geniş soruları göz önünde bulundurmanız gerekir:

- Ağınız ne kadar büyük olacak?
- Yapılandırmasında ne kadar manuel denetime ihtiyacınız var?
- Hem konu ve abonelik açısından hem de kuyruk yöneticileri açısından sistem ne kadar dinamik olacak?
- Kullanılabilirlik ve ölçeklenebilirlik gereksinimleriniz nelerdir?
- Tüm kuyruk yöneticileri doğrudan birbirine bağlanabilir mi?

Yordam

- Ağınızın ne kadar büyük olması gerektiğini tahmin edin.
 - a) Kaç konuya ihtiyacınız olduğunu tahmin edin.
 - b) Kaç tane yayıncı ve abone olmasını beklediğinizi tahmin edin.
 - c) Yayınlama/abone olma etkinliklerine kaç kuyruk yöneticisinin katılacağını tahmin edin.

Ayrıca bkz. “Yayınlama/abone olma kümelemesi: En iyi uygulamalar” sayfa 88, özellikle aşağıdaki bölümler:

- Sisteminizin boyutunun nasıl boyutlandırılır

- Yayınlama/abone olma etkinliğine dahil olan küme kuyruğu yöneticilerinin sayısını sınırlama nedenleri
- Hangi konuların kümeleyeceğine karar verme

Ağınızda çok sayıda kuyruk yöneticisi bulunuyorsa ve birçok yayıncı ve aboneyi işlerse, büyük olasılıkla bir konu anasistemi yönlendirmeli kümesi ya da sıradüzeni kullanmanız gerekir. Doğrudan yönlendirilmiş kümeler neredeyse el ile yapılandırma gerektirmez ve küçük ya da statik ağlar için iyi bir çözüm olabilir.

- Her konuyu, yayıncıyı ya da aboneyi hangi kuyruk yöneticisinde bulduracağınıza ilişkin el ile denetime ne kadar gereksinim duyacağınızı göz önünde bulundurun.
 - a) Kuyruk yöneticilerinizden bazılarının diğerlerinden daha az yetenekli olup olmadığını göz önünde bulundurun.
 - b) Bazı kuyruk yöneticilerinize yönelik iletişim bağlantılarının diğerlerine göre daha kırılabilir olup olmadığını göz önünde bulundurun.
 - c) Bir konunun birçok yayınına ve birkaç abonesine sahip olmasını beklediğiniz durumları belirleyin.
 - d) Bir konunun birçok aboneye ve birkaç yayınına sahip olmasını beklediğiniz durumları belirleyin.

Tüm topolojilerde, yayınlar diğer kuyruk yöneticilerindeki aboneliklere teslim edilir. Doğrudan yönlendirilmiş bir kümede bu yayınlar aboneliklerin en kısa yolunu alır. Konu anasistemi tarafından yönlendirilen bir kümede ya da bir sıradüzende, yayınların izlediği rotayı denetleyebilirsiniz. Kuyruk yöneticileriniz yeteneklerinden farklıysa ya da farklı kullanılabilirlik ve bağlanabilirlik düzeylerine sahipse, büyük olasılıkla belirli iş yüklerini belirli kuyruk yöneticilerine atamak istersiniz. Bunu, konu anasistemi tarafından yönlendirilen bir küme ya da bir sıradüzen kullanarak yapabilirsiniz.

Tüm topolojilerde, mümkün olduğunda aboneliklerle aynı kuyruk yöneticisinde yayınlama uygulamalarının birlikte bulunması, genel giderleri en aza indirir ve performansı en üst düzeye çıkarır. Konu anasistem tarafından yönlendirilen kümeler için, konuyu barındıran kuyruk yöneticilerine yayıncılar ya da aboneler koymayı düşünebilirsiniz. Bu, bir yayını aboneye iletmek için kuyruk yöneticileri arasında fazladan "sekme" işlemini kaldırır. Bu yaklaşım, özellikle bir konunun çok sayıda yayıncısı ve az sayıda abonesi veya çok sayıda abonesi ve az sayıda yayıncısı olduğu durumlarda etkilidir. Örneğin, Merkezleştirilmiş yayıncılar ya da aboneler kullanılarak anasistem yönlendirmesibaşlıklı konuya bakın.

Ayrıca bkz. "Yayınlama/abone olma kümelemesi: En iyi uygulamalar" sayfa 88, özellikle aşağıdaki bölümler:

- Hangi konuların kümeleyeceğine karar verme
- Yayıncı ve abonelik yeri
- Ağ etkinliğinin ne kadar dinamik olacağını göz önünde bulundurun.
 - a) Farklı konularda abonelerin ne sıklıkta ekleneceğini ve kaldırılacağını tahmin edin.

Bir abonelik bir kuyruk yöneticisine eklendiğinde ya da kaldırıldığında ve bu belirli bir konu dizisine ilişkin ilk ya da son abonelik olduğunda, bu bilgiler topolojideki diğer kuyruk yöneticilerine iletilir. Doğrudan yöneltilen bir kümede ve bir sıradüzende, bu abonelik bilgileri, topolojideki her kuyruk yöneticisine konu üzerinde yayıncıları olsun ya da olmasın yayılır. Topoloji birçok kuyruk yöneticisinden oluşuyorsa, bu önemli bir performans ek yükü olabilir. Konu anasistemi tarafından yönlendirilen bir kümede, bu bilgiler yalnızca, aboneliğin konu dizisiyle eşlenen kümelenebilir bir konuyu barındıran kuyruk yöneticilerine yayılır.

"Yayınlama/abone olma kümelemesi: En iyi uygulamalar" sayfa 88 içinde Abonelik değişikliği ve dinamik konu dizileri bölümüne de bakın.

Not: Birçok benzersiz konu dizisi kümesinin hızla ve sürekli olarak değiştirildiği çok dinamik sistemlerde, modeli "her yerde yayınla" kipine geçirmek en iyisi olabilir. Bkz. Yayınlama/abone olma ağlarında abonelik performansı.

- b) Kuyruk yöneticilerinin topolojide ne kadar dinamik olduğunu göz önünde bulundurun.

Bir sıradüzen, topolojideki kuyruk yöneticisinde yapılan her bir değişikliğin el ile eklenmesini ya da sıradüzenden kaldırılmasını gerektirir; sıradüzendeki daha yüksek düzeylerde kuyruk

yöneticileri değiştirilirken dikkatli olunmalıdır. Bir sıradüzendeki kuyruk yöneticileri genellikle el ile yapılandırılan kanal bağlantılarını da kullanır. Kuyruk yöneticileri eklendikçe ve sıradüzenden kaldırıldığında kanal ekleyerek ve kaldırarak bu bağlantıları korumalısınız.

Bir yayınlama/abone olma kümesinde, kuyruk yöneticileri kümeye ilk katıldıklarında gerekli olan diğer kuyruk yöneticilerine otomatik olarak bağlanır ve otomatik olarak konuları ve abonelikleri anlar.

- Rota kullanılabilirliği ve yayın trafiği ölçeklenebilirlik gereksinimlerinizi göz önünde bulundurun.
 - a) Bir kuyruk yöneticisi kullanılmadığında bile, yayınlama kuyruğu yöneticisinden abone olan bir kuyruk yöneticisine her zaman kullanılabilir bir rotanın olması gerekip gerekmediğine karar verin.
 - b) Ağın ne kadar ölçeklenebilir olması gerektiğini göz önünde bulundurun. Yayın trafiği düzeyinin tek bir kuyruk yöneticisi ya da kanal üzerinden yöneltilemeyecek kadar yüksek olup olmadığına ve bu yayın trafiği düzeyinin tek bir konu dalı tarafından mı işleneceğine, yoksa birden çok konu dalına mı dağıtılacağına karar verin.
 - c) İleti sıralamasını korumanız gerekip gerekmediğini göz önünde bulundurun.

Doğrudan yöneltilecek bir küme, iletileri doğrudan yayınlama kuyruğu yöneticilerinden abone olan kuyruk yöneticilerine gönderdiğinden, rota boyunca ara kuyruk yöneticilerinin kullanılabilirliğini göz önünde bulundurmanıza gerek yoktur. Benzer şekilde, ara kuyruk yöneticilerine ölçekleme de dikkate alınmaz. Bununla birlikte, daha önce de belirtildiği gibi, kümedeki tüm kuyruk yöneticileri arasında kanalların ve bilgi akışlarının otomatik olarak korunmasının yükü, özellikle büyük veya dinamik bir ortamda performansı önemli ölçüde etkileyebilir.

Konu anasistemi tarafından yönlendirilen bir küme, tek tek konular için ayarlanabilir. Önemli bir yayın iş yükü olan konu ağacının her dalının farklı bir kuyruk yöneticisinde tanımlandığından ve her kuyruk yöneticisinin yeterince yüksek performans gösterdiğinden ve konu ağacının o dalı için beklenen iş yükü için kullanılabilir olduğundan emin olun. Ayrıca, birden çok kuyruk yöneticisindeki her konuyu tanımlayarak kullanılabilirliği ve yatay ölçeklemeyi daha da geliştirebilirsiniz. Bu, sistemin kullanılmayan konu anasistem kuyruk yöneticilerini yönlendirmesini ve bunlar arasında yayın trafiğini dengelemesini sağlar. Ancak, birden çok kuyruk yöneticisiyle ilgili bir konu tanımladığınızda aşağıdaki kısıtlamaları da tanıtırsınız:

- Yayınlarda ileti sıralamasını kaybedersiniz.
- Alıkonan yayınları kullanamazsınız. Bkz. “[Yayınlama/abone olma kümelerinde tutulan yayınlar için tasarımla ilgili dikkat edilecek noktalar](#)” sayfa 101.

Birden çok rota aracılığıyla bir sıradüzendeki yönlendirmenin yüksek kullanılabilirliğini ya da ölçeklenebilirliğini yapılandıramazsınız.

Ayrıca bkz. “[Yayınlama/abone olma kümelemesi: En iyi uygulamalar](#)” sayfa 88 [Yayın trafiği](#) bölümü.

- Bu hesaplamalara dayalı olarak, bir konu anasistemi tarafından yönlendirilen kümeyi mi, doğrudan yöneltilecek kümeyi mi, sıradüzenini mi, yoksa bu topolojilerin bir karışımını mı kullanacağınıza karar vermenize yardımcı olmak için sağlanan bağlantıları kullanın.

Sonraki adım

Artık dağıtılmış yayınlama/abone olma ağınızı yapılandırmaya hazırsınız.

İlgili görevler

[Kuyruk yöneticisi kümesinin yapılandırılması](#)

[Dağıtılmış kuyruğa alma yapılandırılıyor](#)

[Yayınlama/abone olma kümesinin yapılandırılması](#)

[Kuyruk yöneticisinin yayınlama/abone olma sıradüzenine bağlanması](#)

Yayınlama/abone olma kümeleri tasarlanması

İki temel yayınlama/abone olma küme topolojisi vardır: *doğrudan yöneltme* ve *konu anasistem yöneltmesi*. Her birinin farklı yararları vardır. Yayınlama/abone olma kümenizi tasarlariken, beklenen ağ gereksinimlerinize en uygun topolojiyi seçin.

İki yayınlama/abone olma küme topolojisine genel bakış için bkz. [Küme yayınlama/abone olma](#). Ağ gereksinimlerinizi değerlendirmenize yardımcı olması için bkz. [“Dağıtılmış yayınlama/abone olma ağınızın planlanması” sayfa 69](#) ve [“Yayınlama/abone olma kümelemesi: En iyi uygulamalar” sayfa 88](#).

Genel olarak, her iki küme topolojisi de aşağıdaki avantajları sağlar:

- Noktadan noktaya küme topolojisinin üstündeki basit konfigürasyon.
- Kümeye katılan ve kümeden çıkan kuyruk yöneticilerinin otomatik olarak işlenmesi.
- Fazladan kuyruk yöneticileri ekleyerek ve bunlara ek abonelikler ve yayıncılar dağıtarak ek abonelikler ve yayıncılar için ölçeklendirme kolaylığı.

Bununla birlikte, gereksinimler daha spesifik hale geldiğinden, iki topolojinin farklı avantajları vardır.

Doğrudan yönltilen yayınlama/abone olma kümeleri

Doğrudan yöneltmeyle, kümedeki herhangi bir kuyruk yöneticisi, bağlı uygulamalardan gelen yayınları, eşleşen bir aboneliğe sahip kümedeki diğer herhangi bir kuyruk yöneticisine doğrudan gönderir.

Doğrudan yönlendirilen bir yayınlama/abone olma kümesi aşağıdaki avantajları sağlar:

- Aynı kümedeki belirli bir kuyruk yöneticisinde abonelik için yönlendirilen iletiler doğrudan bu kuyruk yöneticisine iletilir ve bir ara kuyruk yöneticisinden geçmesine gerek yoktur. Bu, anasistem tarafından yönlendirilen topolojiyle ya da sıradüzensel topolojiyle karşılaştırıldığında performansı artırabilir.
- Tüm kuyruk yöneticileri birbirine doğrudan bağlı olduğundan, bu topolojinin yöneltme altyapısında tek bir hata noktası yoktur. Bir kuyruk yöneticisi yoksa, kümedeki diğer kuyruk yöneticilerine ilişkin abonelikler, kullanılabilir kuyruk yöneticilerindeki yayıncılardan ileti almaya devam edebilir.
- Özellikle var olan bir kümede, yapılandırılması çok basittir.

Doğrudan yönlendirilmiş bir yayınlama/abone olma kümesi kullanırken dikkate alınacak konular:

- Kümedeki tüm kuyruk yöneticileri, kümedeki diğer tüm kuyruk yöneticilerini tanırlar.
- Kümelenmiş bir konuya ilişkin bir ya da daha çok abonelik barındıran bir kümedeki kuyruk yöneticileri, bu kuyruk yöneticileri kümelenmiş konularda ileti yayınlamasalar da, kümedeki diğer tüm kuyruk yöneticilerine otomatik olarak küme gönderen kanalları oluşturur.
- Kümelenmiş bir konu altındaki bir konu yöneticisine yapılan ilk abonelik, kümedeki diğer her kuyruk yöneticisine bir ileti gönderilmesiyle sonuçlanır. Benzer şekilde, silinecek konu dizgisine ilişkin son abonelik de bir iletiyle sonuçlanır. Kümelenmiş bir konu altında tek tek konu dizgileri ne kadar çok kullanılırsa, aboneliklerin değiştirilme oranı da o kadar yüksek olursa, kuyruklar arası yönetici iletişimi de o kadar artar.
- Kümedeki her kuyruk yöneticisi, kuyruk yöneticisi bu konuları yayınlamadığında ya da bu konulara abone olmadığında bile, kendisine bildirilen abone olunan konu dizgileri bilgisini korur.

Yukarıdaki nedenlerden ötürü, doğrudan yönltilen bir konuya sahip bir kümedeki tüm kuyruk yöneticileri ek bir ek yük ile karşı karşıya kalacaktır. Kümede kuyruk yöneticileri ne kadar çok olursa, ek yük o kadar büyük olur. Benzer şekilde, abone olunan konu dizgileri ne kadar çok olursa, değişiklik oranı da o kadar yüksek olur. Bu, büyük ya da dinamik doğrudan yönltilen yayınlama/abone olma kümesinde küçük sistemlerde çalışan kuyruk yöneticilerine çok fazla yük yüklenmesine neden olabilir. Daha fazla bilgi için bkz. [Doğrudan yönlendirilmiş yayınlama/abone olma performansı](#) .

Bir kümenin doğrudan yönlendirilmiş kümelenmiş yayınlama/abone olma genel giderlerini karşılamayacağını bildiğinizde, bunun yerine konu anasistemi yönlendirilmiş yayınlama/abone olma özelliğini kullanabilirsiniz. Diğer bir seçenek olarak, kümedeki her kuyruk yöneticisinde **PSCLUS** kuyruk yöneticisi özneteliğini DEVRE Dışı olarak ayarlayarak, kümelenmiş yayınlama/abone olma işlevlerini tümüyle geçersiz kılabilirsiniz. Bkz. [“Kümelenmiş yayınlama/abone olmayı engelleme” sayfa 99](#). Bu, kümelenmiş konuların oluşturulmasını önler ve bu nedenle ağınızın, kümelenmiş yayınlama/abone olma ile ilişkili herhangi bir genel gidere neden olmamasını sağlar.

Konu anasistem tarafından yönlendirilen yayınlama/abone olma kümeleri

Konu anasistem yönlendirmesiyle, kümelenmiş konuların denetimci olarak tanımlandığı kuyruk yöneticileri yayınlar için yönlendirici olur. Kümedeki barındırmayan kuyruk yöneticilerinden gelen yayınlar, barındırma kuyruk yöneticisi aracılığıyla kümedeki herhangi bir kuyruk yöneticisine, eşleşen bir abonelikte yönlendirilir.

Konu anasistem tarafından yönlendirilen yayınlama/abone olma kümesi, doğrudan yönlendirilen bir yayınlama/abone olma kümesi için aşağıdaki ek yararları sağlar:

- Yalnızca anasistem tarafından yönetilen konuların tanımlandığı kuyruk yöneticileri, kümedeki diğer tüm kuyruk yöneticilerini tanır.
- Yalnızca konu anasistem kuyruk yöneticilerinin kümedeki diğer tüm kuyruk yöneticilerine bağlanabilmeleri gerekir ve genellikle yalnızca aboneliklerin var olduğu yerlere bağlanır. Bu nedenle kuyruk yöneticileri arasında çalışan kanallar önemli ölçüde azdır.
- Kümelenmiş bir konuya ilişkin bir ya da daha fazla aboneliği barındıran küme kuyruğu yöneticileri, yalnızca aboneliğin konu dizgisiyle eşlenen bir küme konusunu barındıran kuyruk yöneticilerine otomatik olarak küme gönderen kanallar oluşturur.
- Kümelenmiş bir konu altındaki bir konu yöneticisine yapılan ilk abonelik, kümelenmiş konuyu barındıran kümedeki bir kuyruk yöneticisine gönderilen bir iletiyle sonuçlanır. Benzer şekilde, silinecek konu dizgisine ilişkin son abonelik de bir iletiyle sonuçlanır. Kümelenmiş bir konu altında tek tek konu dizgileri ne kadar çok kullanılırsa ve aboneliklerin değiştirilme hızı ne kadar yüksekse, kuyruklar arası yönetici iletişimi o kadar artar, ancak yalnızca abonelik anasistemleri ve konu anasistemleri arasında.
- Fiziksel yapılandırma üzerinde daha fazla denetim. Doğrudan yönlendirmeye, tüm kuyruk yöneticileri yayınlama/abone olma kümesine katılarak genel giderlerini artırmaları gerekir. Konu anasistem yönlendirmesiyle, yalnızca konu anasistem kuyruk yöneticileri diğer kuyruk yöneticilerini ve bunların aboneliklerini tanır. Konu anasistem kuyruk yöneticilerini belirttik olarak seçtiğinizden, bu kuyruk yöneticilerinin yeterli ekipmanda çalıştığından emin olabilir ve diğer kuyruk yöneticileri için daha az güçlü sistemler kullanabilirsiniz.

Anasistem tarafından yönlendirilen yayınlama/abone olma kümesini kullanırken dikkate alınması gerekenler:

- Yayıncı ya da abone, bir konu barındırma kuyruk yöneticisinde bulunmadığında, yayınlama kuyruğu yöneticisi ile abone olan kuyruk yöneticisi arasında fazladan bir "sekme" kullanılır. Ek "sekmenin" neden olduğu gecikme süresi, konu anasistemi yönlendirmenin doğrudan yönetiminin daha az verimli olduğu anlamına gelebilir.
- Büyük kümelerde, konu anasistemi yönlendirmesi, doğrudan yönlendirmeye elde edebileceğiniz önemli performans ve ölçekleme sorunlarını azaltır.
- Tüm konularınızı tek bir kuyruk yöneticisinde ya da çok az sayıda kuyruk yöneticisinde tanımlamayı seçebilirsiniz. Bunu yaparsanız, konu anasistem kuyruk yöneticilerinin iyi bağlantılı olan güçlü sistemlerde bulunduğundan emin olun.
- Aynı konuyu birden çok kuyruk yöneticisinde tanımlayabilirsiniz. IBM MQ iş yükü, ilgili konuya ilişkin tüm anasistemlerdeki yayınları dengelediği için, bu, konunun kullanılabilirliğini artırır ve ölçeklenebilirliği artırır. Ancak, aynı konuyu birden çok kuyruk yöneticisinde tanımlamanın o konuya ilişkin ileti sırasını kaybettiği unutulmamalıdır.
- Farklı konuları farklı kuyruk yöneticilerine barındırarak, ileti sırasını kaybetmeden ölçeklenebilirliği geliştirebilirsiniz.

İlgili görevler

Senaryo: Yayınlama/abone olma kümesi yaratılması

Yayınlama/abone olma kümesinin yapılandırılması

Dağıtılmış yayınlama/abone olma ağlarını ayarlama

Dağıtılmış yayınlama/abone olma sorunlarının giderilmesi

Yayınlama/abone olma kümelerinde doğrudan yönlendirme

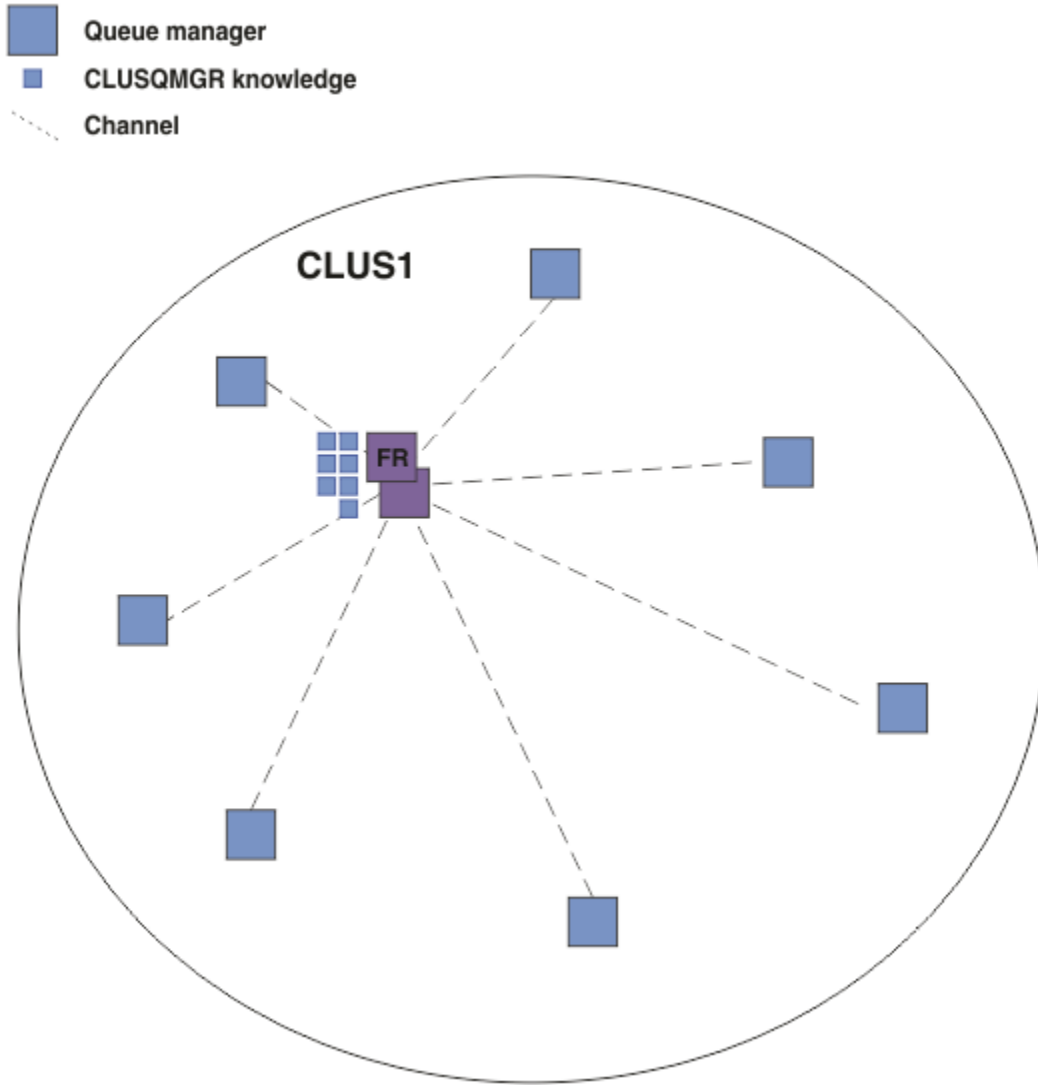
Herhangi bir yayınlama kuyruğu yöneticisinden gelen yayınlar, eşleşen bir aboneliğe sahip kümedeki diğer herhangi bir kuyruk yöneticisine doğrudan yönlendirilir.

Yayınlama/abone olma sıradüzenlerindeki ve kümelerdeki kuyruk yöneticileri arasında iletilerin nasıl yönlendirildiğine ilişkin bir giriş için bkz. [Dağıtılmış yayınlama/abone olma ağları](#).

Doğrudan yöneltile bir yayınlama/abone olma kümesi aşağıdaki gibi davranır:

- Tüm kuyruk yöneticileri diğer tüm kuyruk yöneticilerini otomatik olarak bilir.
- Kümelenmiş konulara abonelikleri olan tüm kuyruk yöneticileri, kümedeki diğer tüm kuyruk yöneticilerine kanallar oluşturur ve onlara abonelikleri hakkında bilgi verir.
- Bir uygulama tarafından yayınlanan iletiler, bağlı olduğu kuyruk yöneticisinden, doğrudan eşleşen bir aboneliğin bulunduğu her kuyruk yöneticisine yöneltilir.

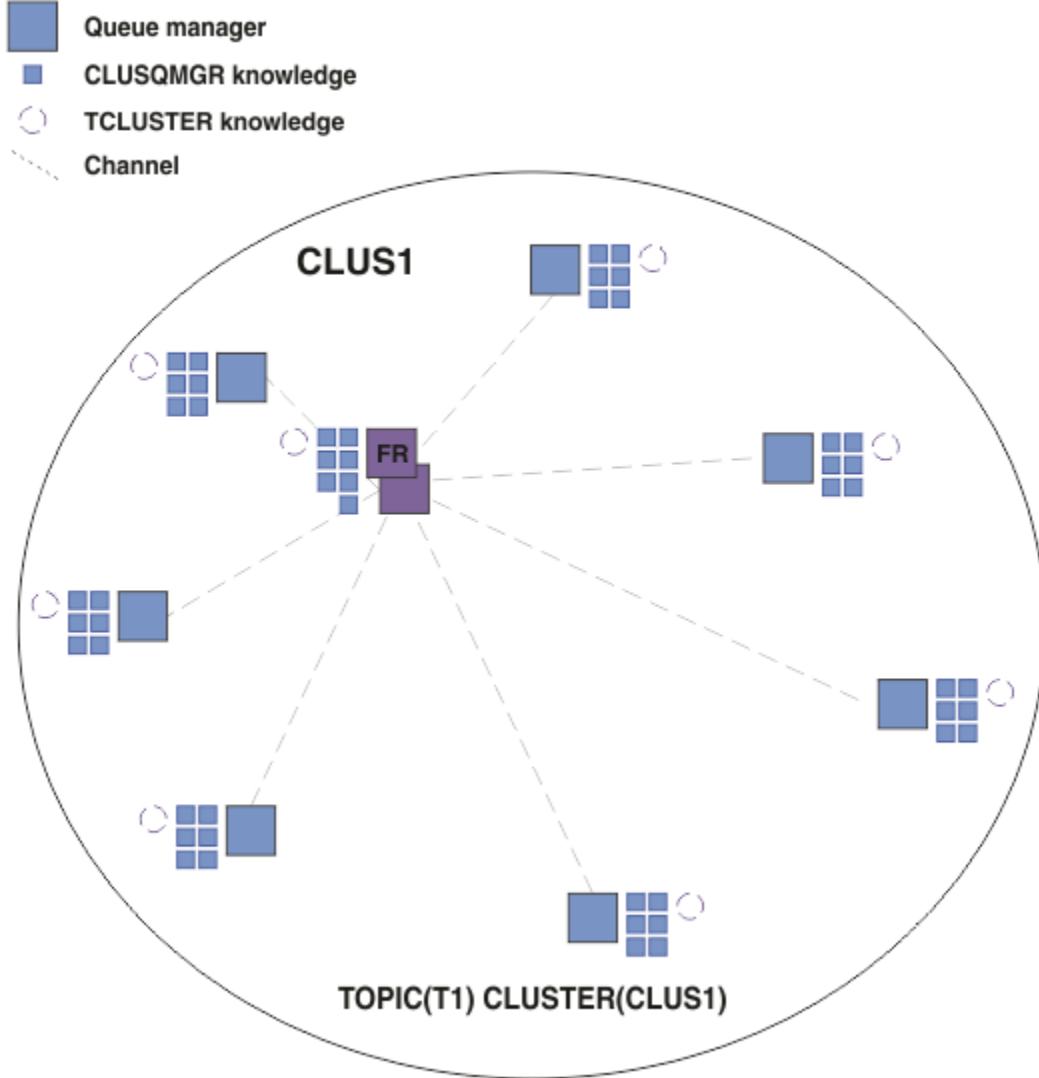
Aşağıdaki çizge, yayınlama/abone olma ya da noktadan noktaya iletişim etkinlikleri için kullanılmayan bir kuyruk yöneticisi kümesini göstermektedir. Kümedeki her kuyruk yöneticisinin yalnızca tam havuz kuyruğu yöneticilerine/yöneticilerinden bağlandığını unutmayın.



Şekil 16. Bir kuyruk yöneticisi kümesi

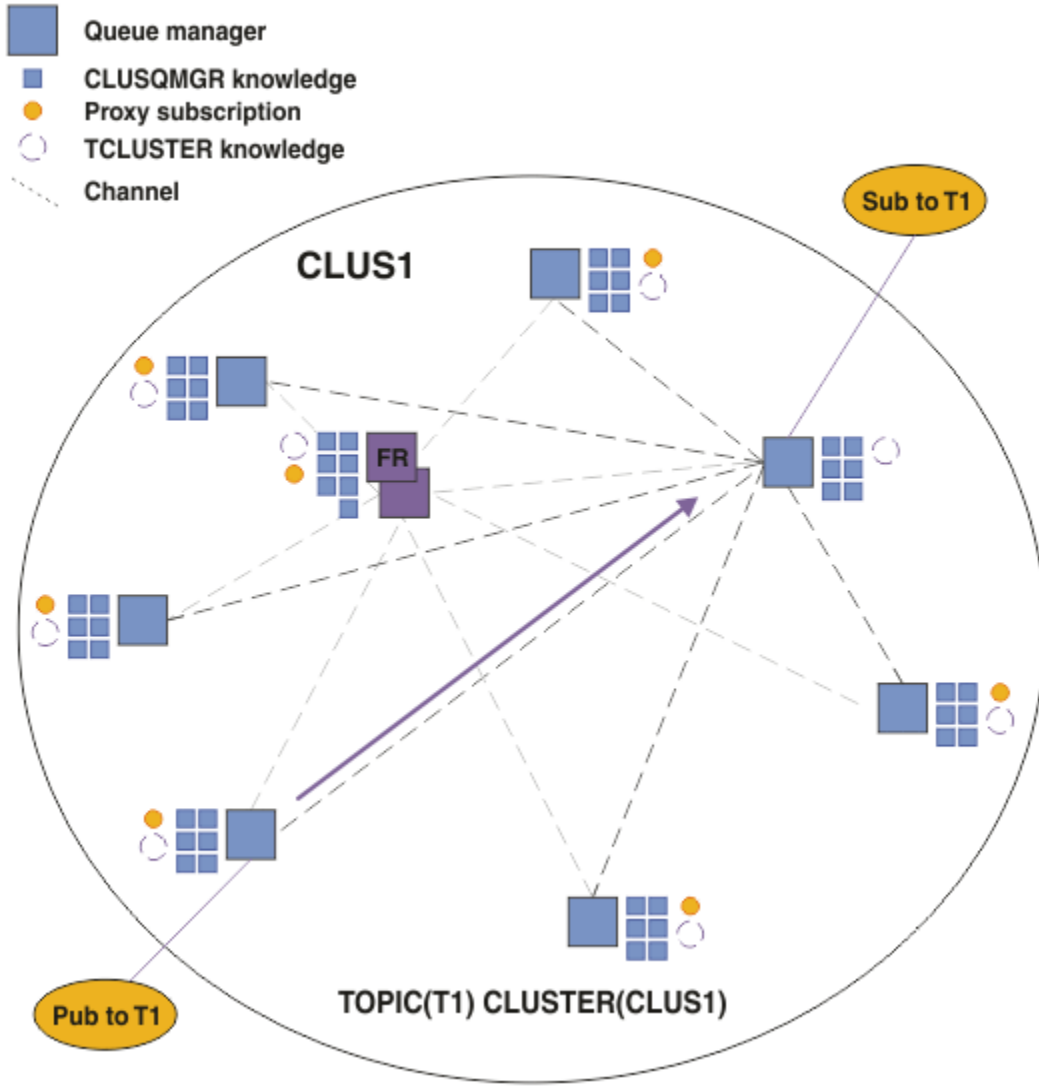
Doğrudan yöneltile bir kümedeki kuyruk yöneticileri arasında yayınların akması için, konu ağacının bir dalını [Bir yayınlama/abone olma kümesinin yapılandırılması](#) konusunda açıkladığı gibi kümellersiniz ve *doğrudan yöneltme* seçeneğini (varsayılan) belirtirsiniz.

Doğrudan yöneltilen bir yayınlama/abone olma kümesinde, konu nesnesini kümedeki herhangi bir kuyruk yöneticisinde tanımlarsınız. Bunu yaptığınızda, nesnenin bilgisi ve kümedeki diğer tüm kuyruk yöneticilerinin bilgisi, otomatik olarak tam havuz kuyruk yöneticileri tarafından kümedeki tüm kuyruk yöneticilerine gönderilir. Bu, herhangi bir kuyruk yöneticisi konuya başvurmadan önce gerçekleşir:



Şekil 17. Doğrudan yöneltilen yayınlama/abone olma kümesi

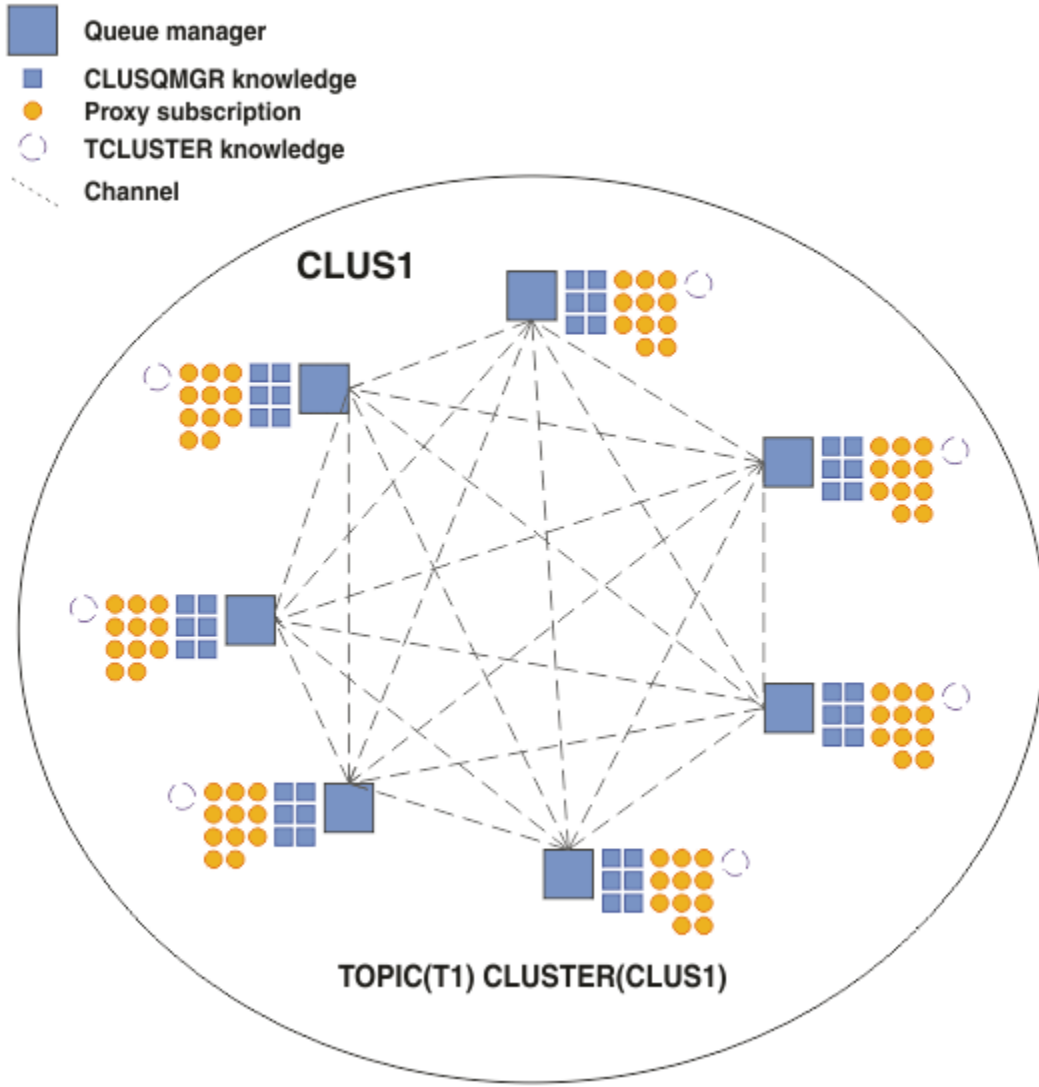
Abonelik yaratıldığında, aboneliği barındıran kuyruk yöneticisi, kümedeki her kuyruk yöneticisine bir kanal oluşturur ve aboneliğin ayrıntılarını gönderir. Bu dağıtılmış abonelik bilgisi, her kuyruk yöneticisinde bir yetkili sunucu aboneliği ile temsil edilir. Kümedeki yetkili aboneliğin konu dizgisiyle eşleşen herhangi bir kuyruk yöneticisinde bir yayın üretildiğinde, yayınlayıcı kuyruk yöneticisinden abonelik barındıran her kuyruk yöneticisine bir küme kanalı oluşturulur ve ileti her birine gönderilir.



Şekil 18. Bir yayıncı ve kümelenmiş bir konuya abone olan doğrudan yönlendirilmiş yayınlama/abone olma kümesi

Yayınlara abonelik barındırma kuyruk yöneticilerine doğrudan yönlendirilmesi, yapılandırmayı basitleştirir ve yayınlara aboneliklere teslim edilmesinde gecikme süresini en aza indirir.

Ancak, aboneliklerin ve yayıncıların konumuna bağlı olarak, kümeniz hızlı bir şekilde tam olarak birbirine bağlanabilir ve her kuyruk yöneticisinin diğer kuyruk yöneticileriyle doğrudan bağlantısı olabilir. Bu, ortamınızda kabul edilebilir ya da edilmeyebilir. Benzer şekilde, abone olunan konu dizgileri kümesi sık sık değişiyorsa, bu bilgilerin tüm kuyruk yöneticileri arasında yayılmasının ek yükü de önemli olabilir. Doğrudan yönlendirilen bir yayınlama/abone olma kümesindeki tüm kuyruk yöneticileri bu ek giderlerle başa çıkabilmelidir.



Şekil 19. Tam olarak birbirine bağlı, doğrudan yönltilen bir yayınlama/abone olma kümesi

Özet ve diğer önemli noktalar

Doğrudan yönlendirilmiş bir yayınlama/abone olma kümesi, oluşturmak veya yönetmek için çok az manuel müdahaleye ihtiyaç duyar ve yayıncılar ile aboneler arasında doğrudan yönlendirme sağlar. Belirli yapılandırmalar için bu genellikle en uygun topoloji, özellikle birkaç kuyruk yöneticisine sahip kümeler ya da yüksek kuyruk yöneticisi bağlantılılığının kabul edilebilir olduğu ve aboneliklerin seyrek olarak değiştiği yerdir. Ancak, sisteminize bazı kısıtlamalar da getirir:

- Her kuyruk yöneticisindeki yük, kümedeki toplam kuyruk yöneticisi sayısı ile orantılıdır. Bu nedenle, daha büyük kümelerde, tek tek kuyruk yöneticileri ve bir bütün olarak sistem performans sorunlarıyla karşılaşabilir.
- Varsayılan olarak, abone olunan tüm kümelenebilir konu dizgileri küme boyunca yayılır ve yayınlar yalnızca ilişkili konuya aboneliği olan uzak kuyruk yöneticilerine yayılır. Bu nedenle abonelik kümesinde yapılan hızlı değişiklikler sınırlayıcı bir faktör olabilir. Bu varsayılan davranışı değiştirebilir ve bunun yerine tüm yayınları tüm kuyruk yöneticilerine yayarak yetkili sunucu abonelikleri gereksinimini ortadan kaldırebilirsiniz. Bu, abonelik bilgi trafiğini azaltır, ancak büyük olasılıkla yayın trafiğini ve her kuyruk yöneticisinin oluşturduğu kanal sayısını artırır. Bkz. [Yayınlama/abone olma ağlarında abonelik performansı](#).

Not: Benzer bir kısıtlama sıradüzenler için de geçerlidir.

- Yayınlama/abone olma kuyruk yöneticilerinin birbiriyle bağlantılı olması nedeniyle, yetkili sunucu aboneliklerinin ağdaki tüm düğümlere yayılması zaman alır. Uzak yayınların hemen abone olmaya başlaması gerekmez; bu nedenle, yeni bir konu dizesi aboneliğinden sonra erken yayınlar gönderilmeyebilir. Tüm yayınları tüm kuyruk yöneticilerine yayararak abonelik gecikmesinin neden olduğu sorunları kaldırabilirsiniz; bu, yetkili sunucu abonelikleri gereksinimini ortadan kaldırır. Bkz. [Yayınlama/abone olma ağlarında abonelik performansı](#).

Not: Bu kısıtlama sıradüzenler için de geçerlidir.

Doğrudan yönlendirmeyi kullanmadan önce, [“Konu Yayınlama/abone olma kümelerinde anasistem yönlendirmesi” sayfa 79](#)ve [“Yayınlama/abone olma sıradüzenlerinde yönlendirme” sayfa 103](#) içinde ayrıntılı olarak açıklanan alternatif yaklaşımları keşfedin.

Konu Yayınlama/abone olma kümelerinde anasistem yönlendirmesi

Kümedeki barındırmayan kuyruk yöneticilerinden gelen yayınlar, barındırma kuyruk yöneticisi aracılığıyla kümedeki herhangi bir kuyruk yöneticisine, eşleşen bir abonelikle yönlendirilir.

Yayınlama/abone olma sıradüzenlerindeki ve kümelerdeki kuyruk yöneticileri arasında iletilerin nasıl yönlendirildiğine ilişkin bir giriş için bkz. [Dağıtılmış yayınlama/abone olma ağları](#).

Konu anasistemi yönlendirmesinin davranışını ve yararlarını anlamak için öncelikle [“Yayınlama/abone olma kümelerinde doğrudan yönlendirme” sayfa 75](#) 'i anlamanız gerekir.

Anasistem tarafından yönlendirilen yayınlama/abone olma kümesi aşağıdaki gibi davranır:

- Kümeleneş denetlenen konu nesnelere, kümedeki tek tek kuyruk yöneticilerine el ile tanımlanır. Bunlara *konu anasistem kuyruk yöneticileridir*.
- Bir küme kuyruęu yöneticisinde abonelik yapıldığında, abonelik anasistem kuyruk yöneticisinden konu anasistem kuyruk yöneticilerine kanallar yaratılır ve yetkili sunucu abonelikleri yalnızca konuyu barındıran kuyruk yöneticisinde yaratılır.
- Bir uygulama bir konuda bilgi yayınladığında, baęlı kuyruk yöneticisi yayını her zaman konuyu barındıran bir kuyruk yöneticisine iletir ve bu yöneticiyi, konu ile eşleşen abonelikleri olan kümedeki tüm kuyruk yöneticilerine iletir.

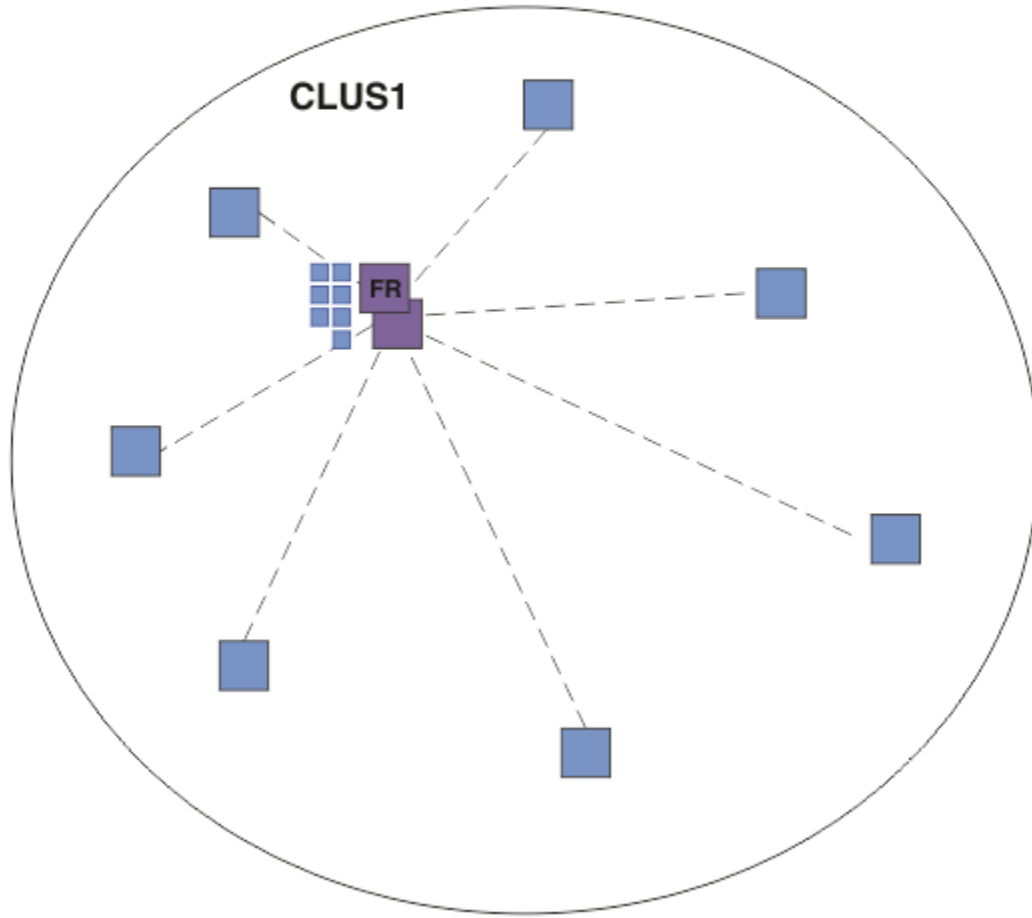
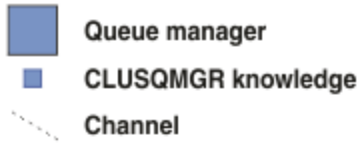
Bu süreç aşağıdaki örneklerde daha ayrıntılı olarak açıklanmıştır.

Konu Tek bir konu anasistemi kullanılarak anasistem yönlendirmesi

Bir konu anasistem tarafından yönlendirilen kümede kuyruk yöneticileri arasında yayınların akması için, konu ağacının bir dalını [Yayınlama/abone olma kümesinin yapılandırılması](#) konusunda açıkladığı gibi kümelersiniz ve *konu anasistem yönlendirmesi* seçeneğini belirtirsiniz.

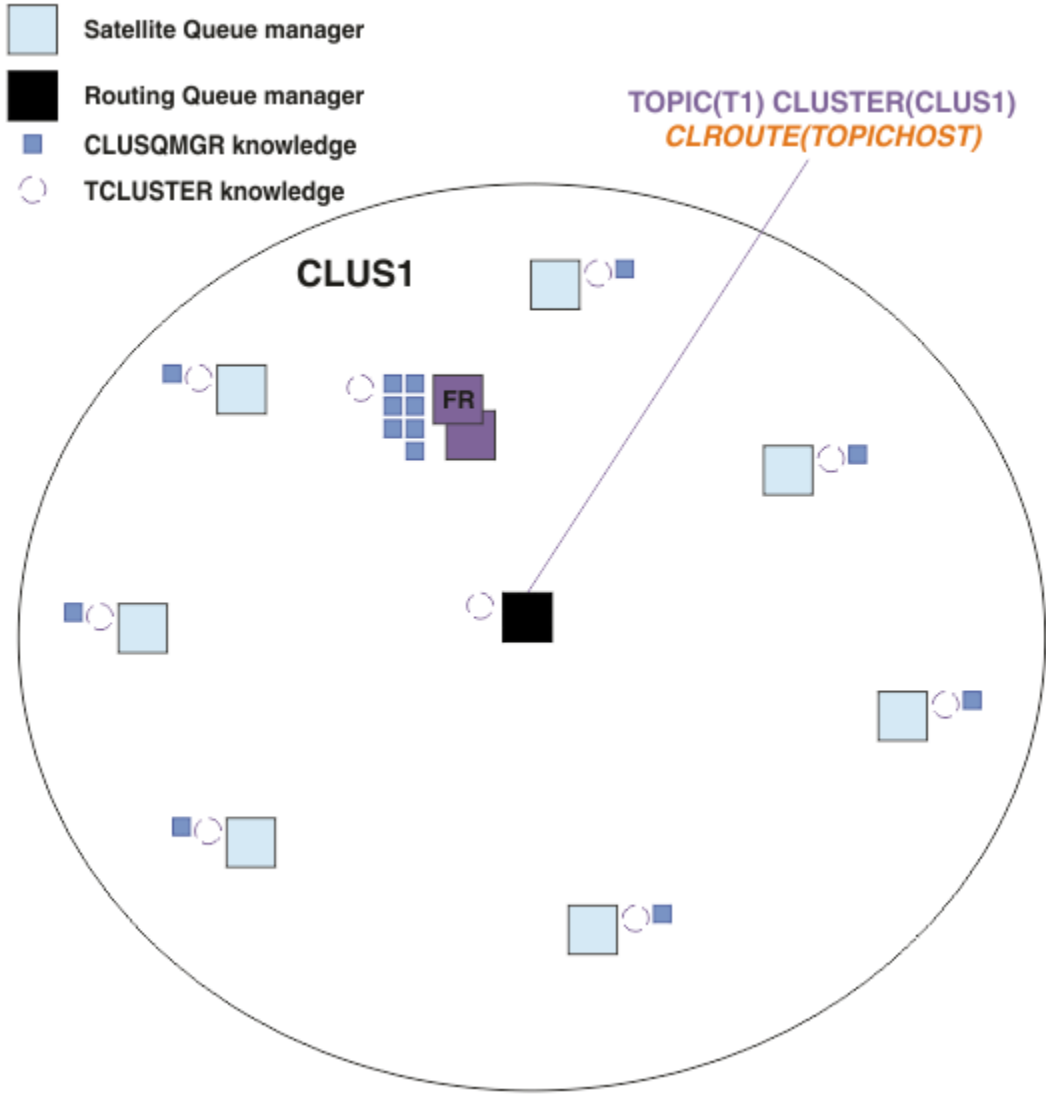
Bir kümedeki birden çok kuyruk yöneticisinde konu anasistemi tarafından yönlendirilen konu nesnesi tanımlamak için çeşitli nedenler vardır. Ancak, basitlik için tek bir konu ana makinesiyle başlıyoruz.

Aşağıdaki çizge, yayınlama/abone olma ya da noktadan noktaya iletişim etkinlikleri için kullanılmayan bir kuyruk yöneticisi kümesini göstermektedir. Kümedeki her kuyruk yöneticisinin yalnızca tam havuz kuyruęu yöneticilerine/yöneticilerinden baęlandığını unutmayın.



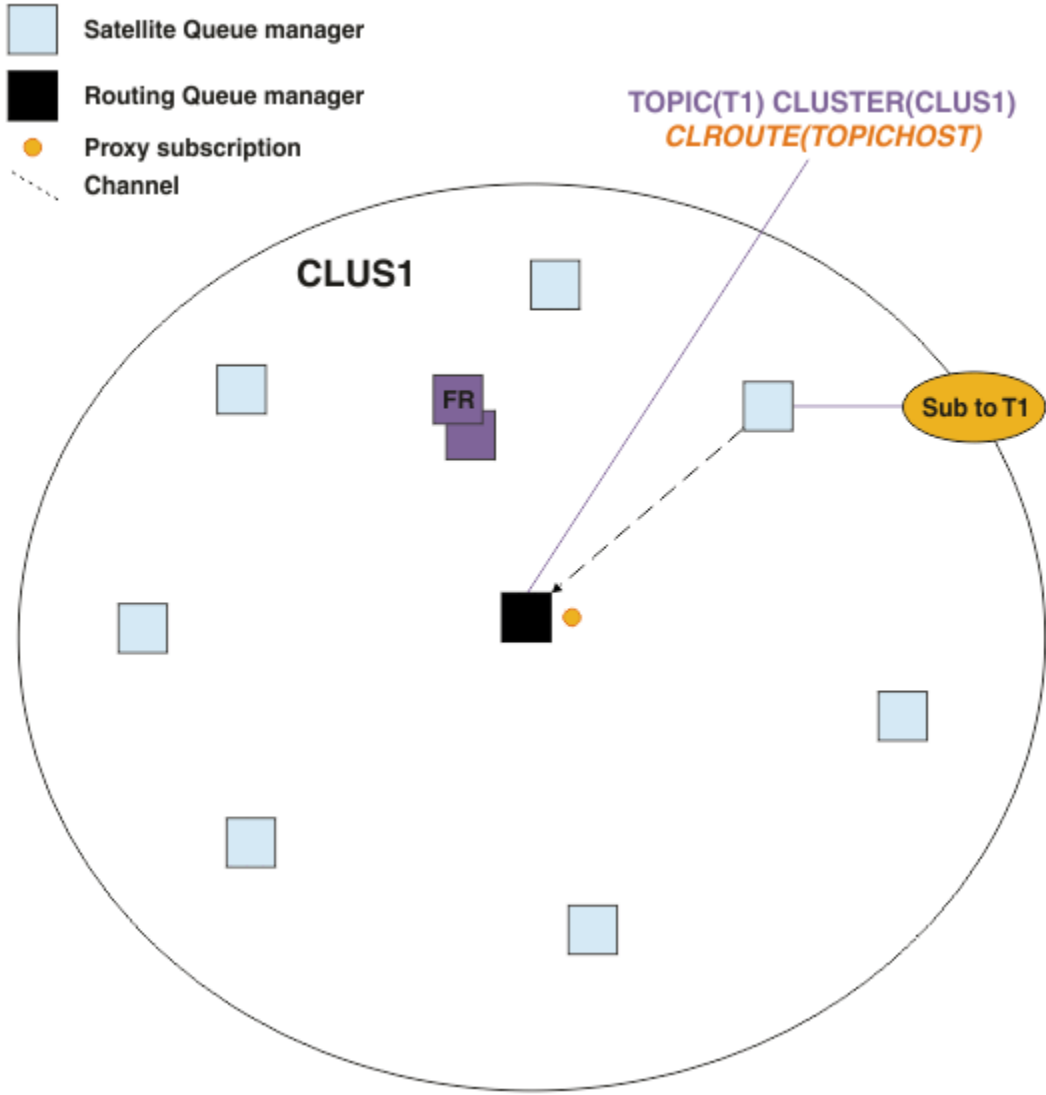
Şekil 20. Bir kuyruk yöneticisi kümesi

Anasistem tarafından yönlendirilen bir yayınlama/abone olma kümesinde, konu nesnesini kümedeki belirli bir kuyruk yöneticisinde tanımlarsınız. Trafiği yayınlayın/abone olun, daha sonra bu kuyruk yöneticisinden geçer, bu da onu kümede kritik bir kuyruk yöneticisi yapar ve iş yükünü artırır. Bu nedenle, tam bir havuz kuyruğu yöneticisi kullanılması değil, kümede başka bir kuyruk yöneticisi kullanılması önerilir. Konu nesnesini anasistem kuyruk yöneticisinde tanımladığınızda, nesnenin ve anasisteminin bilgisi, tam havuz kuyruğu yöneticileri tarafından kümedeki diğer tüm kuyruk yöneticilerine otomatik olarak gönderilir. *Doğrudan yöneltme*' den farklı olarak, her kuyruk yöneticisine kümedeki diğer her kuyruk yöneticisi hakkında bilgi vermediğini unutmayın.



Şekil 21. Bir konu anasisteminde tek bir konu tanımlı olarak anasistem tarafından yönlendirilen yayınlama/abone olma kümesi

Bir kuyruk yöneticisinde abonelik yaratıldığında, abone olan kuyruk yöneticisi ile konu anasistem kuyruk yöneticisi arasında bir kanal yaratılır. Abone olan kuyruk yöneticisi yalnızca konu anasistem kuyruk yöneticisine bağlanır ve aboneliğin ayrıntılarını (*yetkili sunucu aboneliği* biçiminde) gönderir. Konu anasistem kuyruk yöneticisi, bu abonelik bilgilerini kümedeki diğer kuyruk yöneticilerine iletmez.

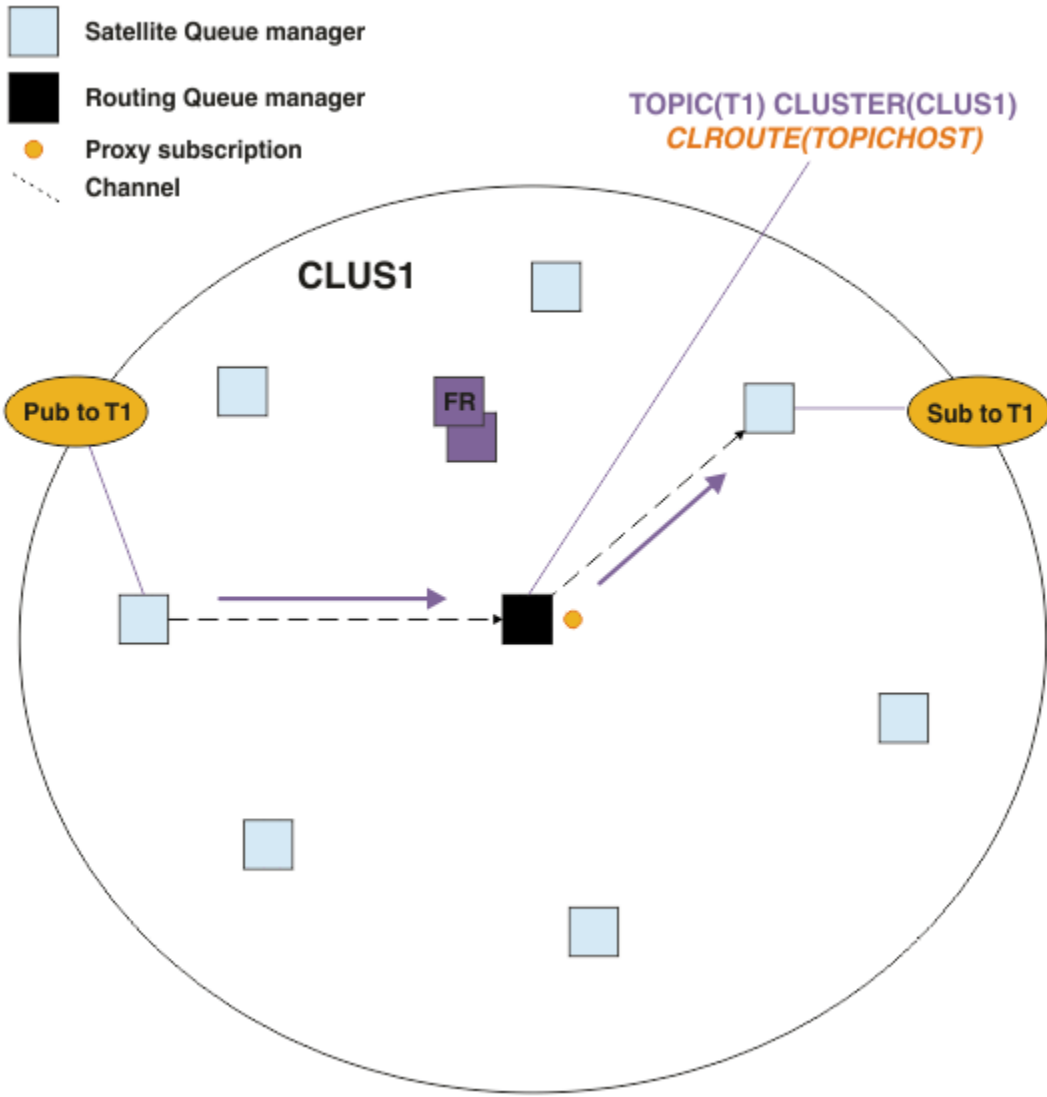


Şekil 22. Bir konu anasisteminde tanımlanmış bir konu ve bir abone içeren anasistem tarafından yönlendirilen yayınlama/abone olma kümesi

Bir yayınlama uygulaması başka bir kuyruk yöneticisine bağlandığında ve bir ileti yayınlandığında, yayınlama kuyruğu yöneticisi ile konu anasistem kuyruk yöneticisi arasında bir kanal yaratılır ve ileti o kuyruk yöneticisine iletilir. Yayınlama kuyruğu yöneticisinin kümedeki diğer kuyruk yöneticilerine ilişkin herhangi bir bilgi sahibi olmadığı için, ileti kümede o konuya abone olmasa da konu anasistem kuyruk yöneticisine iletilir. Yayınlama kuyruğu yöneticisi yalnızca konu anasistem kuyruk yöneticisine bağlanır. Varsa, yayınlar konu anasistemi aracılığıyla abone olan kuyruk yöneticilerine yönlendirilir.

Yayınlayıcıyla aynı kuyruk yöneticisine ilişkin abonelikler, öncelikle iletileri bir konu anasistem kuyruk yöneticisine göndermeden doğrudan karşılanır.

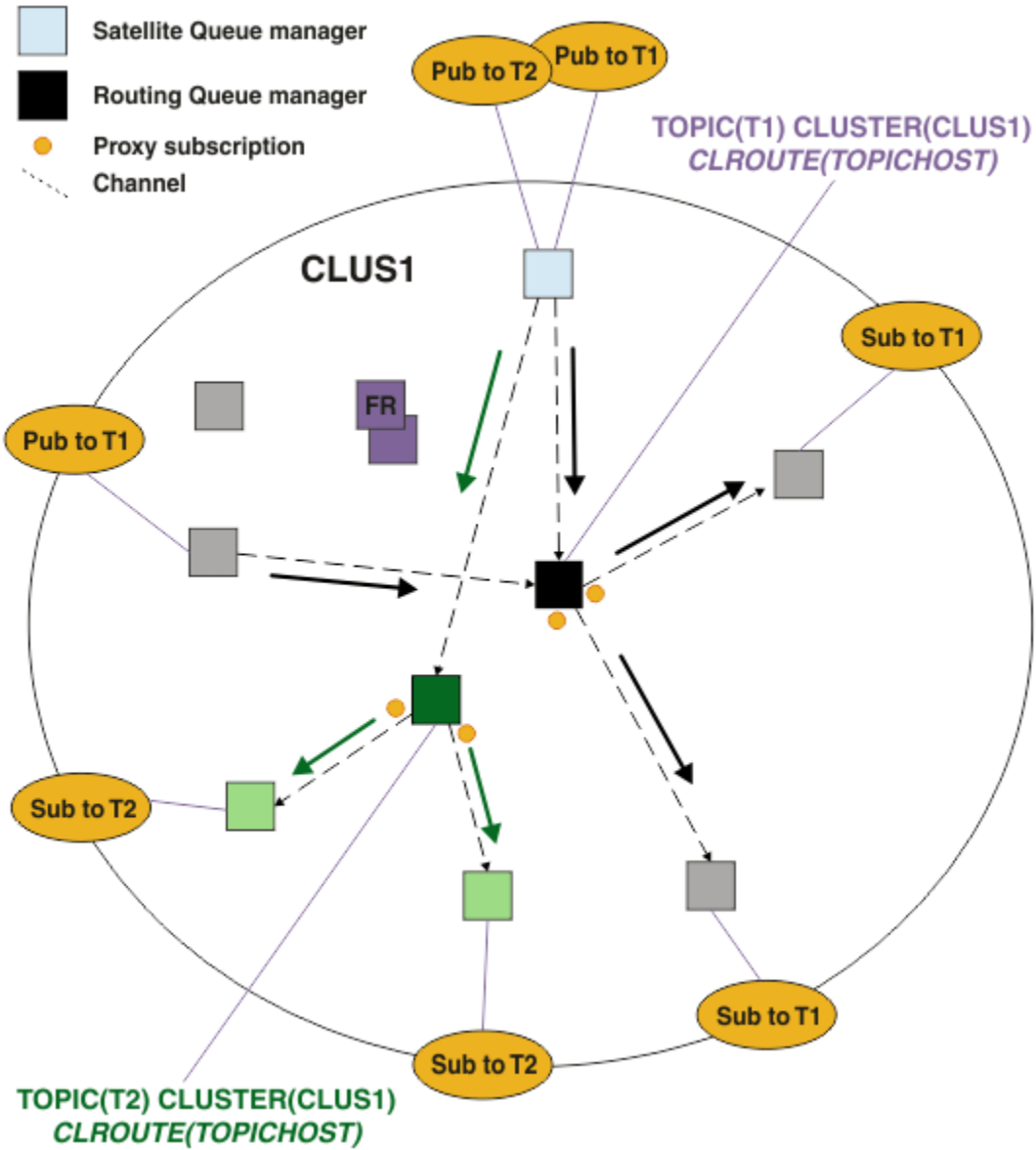
Her konu anasistem kuyruk yöneticisinin oynadığı kritik rol nedeniyle, konu barındırmanın yükleme, kullanılabilirlik ve bağlantı gereksinimlerini işleyebilecek kuyruk yöneticilerini seçmeniz gerektiğini unutmayın.



Şekil 23. Bir konu, bir konu, bir abone ve bir yayınlayıcı içeren anasistem tarafından yönlendirilen yayınlama/abone olma kümesi

Konu ağacının birden çok kuyruk yöneticilerine bölünmesi

Yönetilen bir konu barındırma kuyruk yöneticisi, yalnızca yönetilen konu nesnesinin yapılandırıldığı konu ağacının dalına ilişkin abonelik bilgisi ve yayın iletilerinden sorumludur. Kümedeki farklı yayınlama/abone olma uygulamaları tarafından farklı konular kullanılıyorsa, konu ağacının farklı kümelenmiş dallarını barındıracak farklı kuyruk yöneticileri yapılandırabilirsiniz. Bu, kümedeki her konu anasistem kuyruk yöneticisinde yayın trafiğini, abonelik bilgilerini ve kanalları azaltarak ölçeklemeye olanak sağlar. Bu yöntemi, konu ağacının ayrı yüksek birim dalları için kullanmalısınız:



Şekil 24. Her biri bir konu anasisteminde tanımlanmış iki konu içeren anasistem tarafından yönlendirilen yayınlama/abone olma kümesi

Örneğin, Konu ağaçlarında açıklanan konular kullanılarak, T1 konusu /USA/Alabamakonu dizgisiyle yapılandırıldıysa ve T2 konusu /USA/Alaskakonu dizgisiyle yapılandırıldıysa, /USA/Alabama/Mobile içinde yayınlanan bir ileti, T1kuyruk yöneticisi aracılığıyla yönlendirilir. ve /USA/Alaska/Juneau 'e yayınlanan bir ileti, T2' yi barındıran kuyruk yöneticisi aracılığıyla yönlendirilir.

Not: Tek bir abonelik, konu ağacında kümelenmiş noktalardan daha yüksek bir joker karakter kullanarak konu ağacının birden çok kümeli dalına yayılamaz. Bkz. Genel arama kartı abonelikleri.

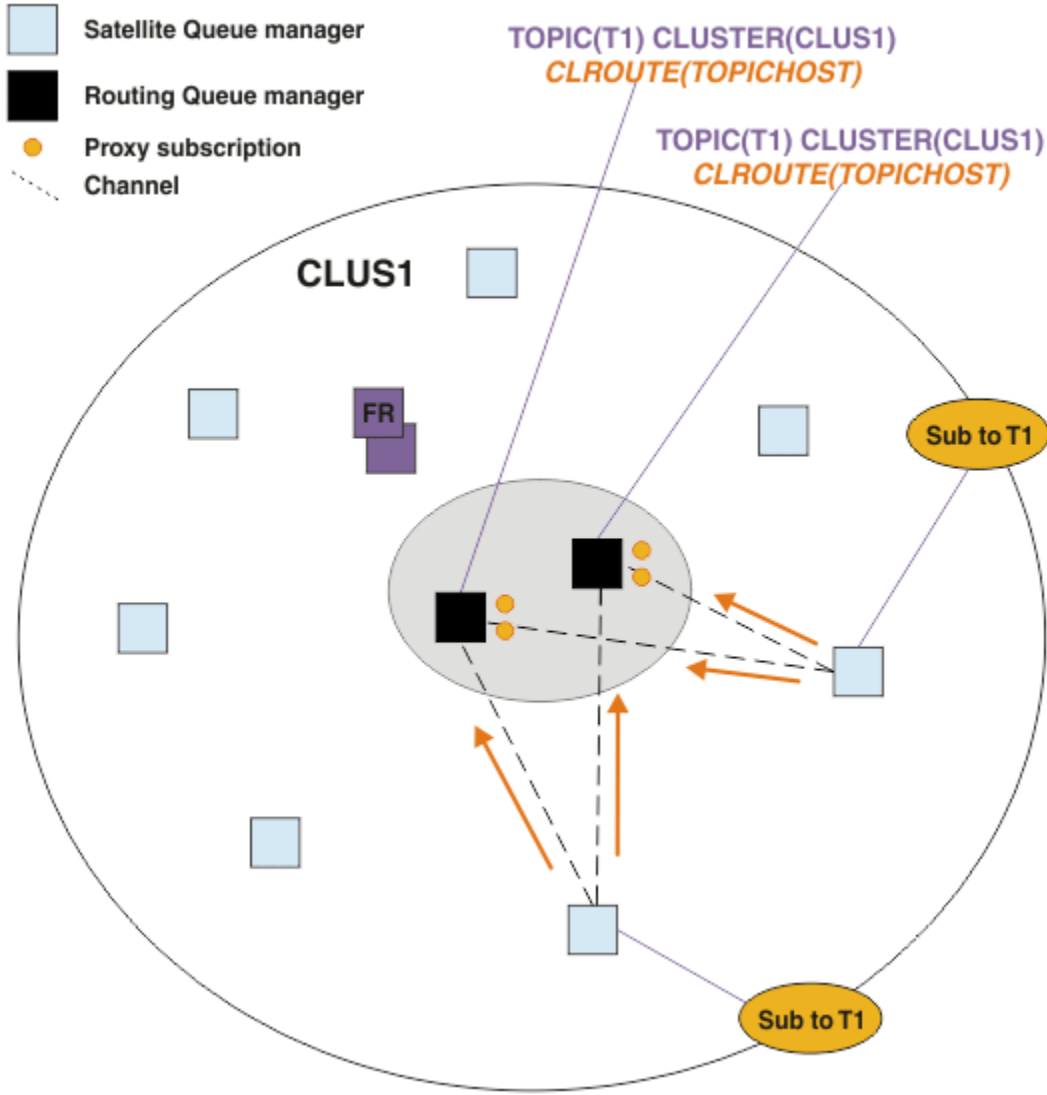
Konu Tek bir konu için birden çok konu anasistemi kullanarak anasistem yöneltmesi

Tek bir kuyruk yöneticisinin bir konunun yönlendirilmesiyle ilgili sorumluluğu varsa ve bu kuyruk yöneticisi kullanılamaz duruma gelirse ya da iş yükünü işleyemezse, yayınlar aboneliklere hemen akmaz.

Yalnızca bir kuyruk yöneticisinde bir konu tanımladığınızda elde ettiğinizden daha fazla esneklik, ölçeklenebilirlik ve iş yükü dengelemeye gereksinim duyarsanız, birden fazla kuyruk yöneticisinde bir konu tanımlayabilirsiniz. Yayınlanan her bir ileti, tek bir konu anasistemi aracılığıyla yönlendirilir. Eşleşen birden çok konu anasistemi tanımlaması varsa, konu anasistemlerinden biri seçilir. Seçim, kümelenmiş kuyruklar için olduğu gibi yapılır. Bu, iletilerin kullanılabilir konu anasistemlerine yöneltilmesini, kullanılmayan

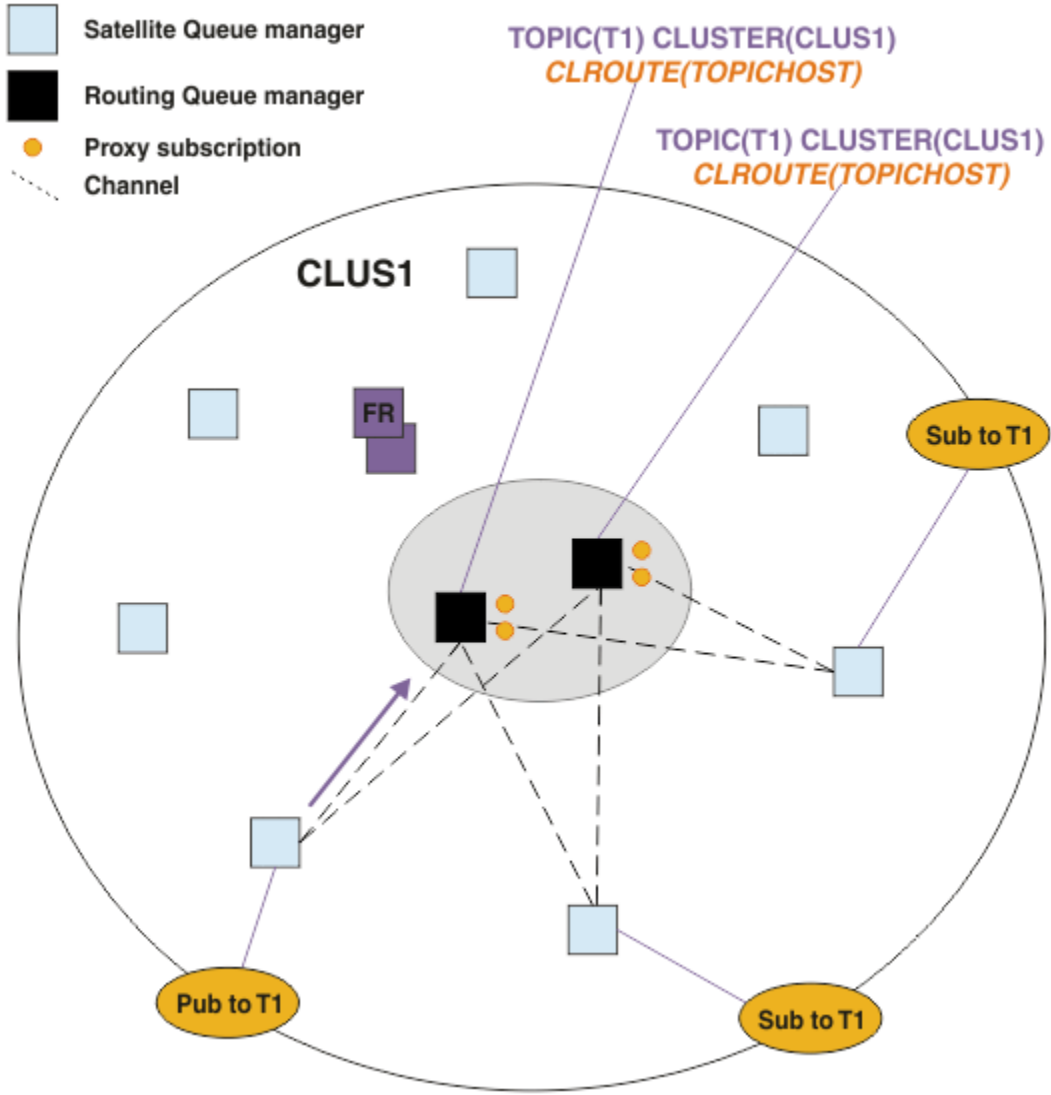
iletilerin önlenmesine ve ileti yükünün birden çok konu anasistem kuyruk yöneticisi ve kanalında dengelenmesine olanak sağlar. Ancak, kümede aynı konu için birden çok konu anasistemi kullandığınızda, birden çok ileti arasında sıralama korunmaz.

Aşağıdaki çizge, aynı konunun iki kuyruk yöneticisinde tanımlandığı bir konu anasistem tarafından yönlendirilen kümeyi göstermektedir. Bu örnekte, abone olan kuyruk yöneticileri abone olunan konuyla ilgili bilgileri her iki konu anasistem kuyruk yöneticisine de yetkili sunucu aboneliği biçiminde gönderir:



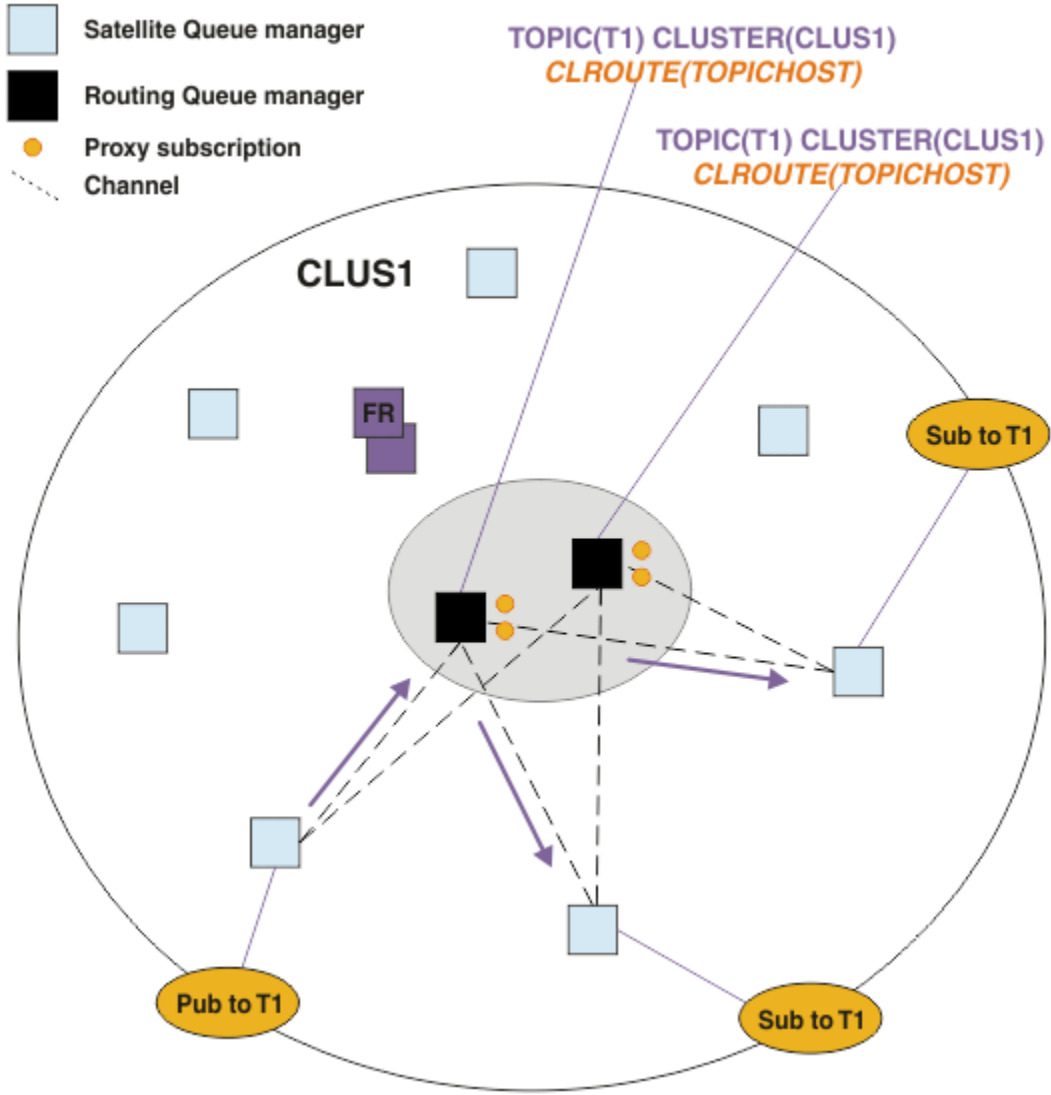
Şekil 25. Birden çok konu içeren bir anasistem yayınlama/abone olma kümesinde yetkili sunucu abonelikleri yaratılması

Bir yayın, barındırmayan bir kuyruk yöneticisinden yapıldığında, kuyruk yöneticisi yayının bir kopyasını o konuya ilişkin konu anasistem kuyruk yöneticilerinden *birine* gönderir. Sistem, küme iş yükü yönetimi algoritması varsayılan davranışına dayalı olarak anasistemi seçer. Tipik bir sistemde, bu, her konu anasistem kuyruk yöneticisinde bir sıralı zincir dağılımına benzer. Aynı yayınlama uygulamasındaki iletiler arasında benzerlik yoktur; bu, NOTFIXEDküme bağ tanımlama tipinin kullanılmasına eşittir.



Şekil 26. Birden çok konu içeren bir anasistem yayınlama/abone olma kümesinde yayınların alınması

Seçilen konu anasistem kuyruk yöneticisine ilişkin gelen yayınlar, eşleşen bir yetkili sunucu aboneliği kaydettirmiş olan tüm kuyruk yöneticilerine iletilir:



Şekil 27. Yayınların birden çok konu içeren bir anasistem yayınlama/abone olma kümesindeki abonelere yönlendirilmesi

Aboneliklerin ve yayıncıların bir konu anasistem kuyruk yöneticisinde yerel olarak yapılması

Yukarıdaki örnekler, yönetilen konu nesnelere barındırmak istemeyen kuyruk yöneticilerindeki yayıncılar ve aboneler arasındaki yönlendirmeyi gösterir. Bu topolojilerde, iletiler aboneliklere erişmek için birden çok sekme gerektirir.

Ek sekmenin istenmemesi durumunda, anahtar yayıncılarının konu barındırma kuyruk yöneticilerine bağlanması uygun olabilir. Ancak, bir konu için birden çok konu anasistemi ve yalnızca bir yayıncı varsa, tüm yayın trafiği, yayıncının bağlı olduğu konu anasistem kuyruk yöneticisi aracılığıyla yönlendirilir.

Benzer şekilde, anahtar abonelikleri varsa, bunlar bir konu anasistem kuyruk yöneticisinde bulunabilir. Ancak, yöneltilen konunun birden çok anasistemi varsa, yalnızca yayınların bir kısmı ek sekme girmekten kaçınacaktır; geri kalanlar önce diğer konu anasistem kuyruk yöneticileri aracılığıyla yönlendirilecektir.

Bu gibi topolojiler burada daha ayrıntılı olarak açıklanmıştır: [Merkezleştirilmiş yayıncılar ya da aboneler kullanılarak anasistem yönlendirmesi](#).

Not: Yöneltilen konu anasistemleriyle yayıncılar ya da abonelikler birlikte bulunurken yönlendirilen konu yapılandırması değiştirildiğinde özel planlama gerekir. Örneğin, Konu anasistemi tarafından yönlendirilen kümeye ek konu anasistemleri eklemebaşlıklı konuya bakın.

Özet ve diğer önemli noktalar

Konu anasistem tarafından yönlendirilen yayınlama/abone olma kümesi, size her konuyu hangi kuyruk yöneticilerinin bulundurduğu konusunda kesin denetim sağlar ve bu kuyruk yöneticileri, konu ağacının o dalı için *yöneltilme* kuyruk yöneticileri olur. Ayrıca, abonelikleri ya da yayıncıları olmayan kuyruk yöneticilerinin konu anasistem kuyruk yöneticileriyle bağlantı kurmalarına gerek yoktur ve abonelikleri olan kuyruk yöneticilerinin bir konuyu barındırmak istemeyen kuyruk yöneticilerine bağlanmalarına gerek yoktur. Bu yapılandırma, kümedeki kuyruk yöneticileri arasındaki bağlantı sayısını ve kuyruk yöneticileri arasında geçirilen bilgi miktarını önemli ölçüde azaltabilir. Bu özellikle, kuyruk yöneticilerinin yalnızca bir alt kümesinin yayınlama/abone olma işi gerçekleştirdiği büyük kümeler için geçerlidir. Bu yapılandırma, kümedeki tek tek kuyruk yöneticilerindeki yük üzerinde de bir denetim sağlar; böylece (örneğin) daha güçlü ve daha esnek sistemlerde yüksek etkin konuları barındırmayı seçebilirsiniz. Belirli yapılandırmalar için-özellikle daha büyük kümeler-genellikle *doğrudan yönlendirmede* daha uygun bir topolojidir.

Ancak, konu anasistemi yönlendirmesi sisteminize bazı kısıtlamalar da getirir:

- Sistem yapılandırması ve bakımı, doğrudan yönlendirmeye göre daha fazla planlama gerektirir. Konu ağacında hangi kümeyi göstereceğine ve kümedeki konu tanımlamalarının konumuna karar vermeniz gerekir.
- Yöneltilen doğrudan konular için olduğu gibi, yeni bir konu anasistem tarafından yönlendirilen konu tanımlandığında, bilgiler tam havuz kuyruğu yöneticilerine ve oradan doğrudan kümenin tüm üyelerine gönderilir. Bu olay, başlatılmadıysa, kümenin her üyesi için tam havuzlardan kanalların başlatılmasına neden olur.
- Yayınları, kümede abonelik olmasa da, anasistem kuyruk yöneticisinden her zaman anasistem kuyruk yöneticisine gönderilir. Bu nedenle, aboneliklerin genellikle var olması beklenirken ya da genel bağlantı ve bilgi yükü ek yayın trafiği riskinden daha fazlayken yönlendirilen konuları kullanmanız gerekir.

Not: Daha önce açıklandığı gibi, yayıncıların bir konu anasisteminde yerel hale getirilmesi bu riski azaltabilir.

- Anasistem dışı kuyruk yöneticisinde yayınlanan iletiler, aboneliği barındıran kuyruk yöneticisine doğrudan gitmez, her zaman bir konu anasistem kuyruk yöneticisi aracılığıyla yönlendirilir. Bu yaklaşım, kümenin toplam ek yükünü artırabilir ve ileti gecikme süresini artırabilir ve performansı azaltabilir.

Not: Daha önce açıklandığı gibi, aboneliklerin ya da yayıncıların bir konu anasisteminde yerel kılınması bu riski azaltabilir.

- Tek bir konu anasistem kuyruk yöneticisinin kullanılması, bir konuda yayınlanan tüm iletiler için tek bir hata noktası sağlar. Birden çok konu anasistemi tanımlayarak bu tek hata noktasını kaldırabilirsiniz. Ancak, birden çok anasisteme sahip olmak, abonelikler tarafından alınan yayınlanan iletilerin sırasını etkiler.
- Birden çok kuyruk yöneticisinden gelen yayınlama trafiğinin bu yöneticiler tarafından işlenmesi gerektiğinden, konu anasistem kuyruk yöneticileri fazladan ileti yüklemesini gerçekleştirmektedir. Bu yük azaltılabilir: Tek bir konu için birden çok konu anasistemi kullanın (bu durumda ileti sıralaması korunmaz) ya da konu ağacının farklı dalları için yönlendirilen konuları barındırmak üzere farklı kuyruk yöneticileri kullanın.

Konu anasistemi yönlendirmesini kullanmadan önce, “Yayınlama/abone olma kümelerinde doğrudan yönlendirme” sayfa 75 ve “Yayınlama/abone olma sıradüzenlerinde yönlendirme” sayfa 103 içinde ayrıntılı olarak açıklanan alternatif yaklaşımları keşfedin.

Yayınlama/abone olma kümelemesi: En iyi uygulamalar

Kümelenmiş konuların kullanılması, kuyruk yöneticileri arasında yayınlama/abone olma etki alanının genişletilmesini kolaylaştırır, ancak mekanik ve sonuçlar tam olarak anlaşılmazsa sorunlara yol açabilir.

Bilgi paylaşımı ve yayın yönlendirmesi için iki model vardır. Bireysel iş gereksinimlerinizi en iyi şekilde karşılayan ve seçtiğiniz kümede en iyi performansı sağlayan modeli uygulayın.

Aşağıdaki bölümlerdeki en iyi uygulama bilgileri tüm çözümlere tek bir boyut sağlamaz, bunun yerine ortak sorunları çözmek için ortak yaklaşımları paylaşır. IBM MQ kümeleri ve yayınlama/abone olma ileti sistemi hakkında temel bilgilere sahip olduğunuzu ve [Dağıtılmış yayınlama/abone olma ağları](#) ve [“Yayınlama/abone olma kümeleri tasarlanması” sayfa 72](#) içindeki bilgilere aşina olduğunuzu varsayar.

Bir kümeyi noktadan noktaya ileti sistemi için kullandığınızda, kümedeki her kuyruk yöneticisi bilgi gereksinimi temelinde çalışır. Yani, yalnızca kümedeki diğer kuyruk yöneticileri ve kümelenmiş kuyruklardaki diğer kuyruk yöneticileri gibi diğer küme kaynaklarını, bunlara bağlanan uygulamalar bunları kullanmak için istekte bulunduğu bulur. Bir kümeye yayınlama/abone olma ileti sistemi eklediğinizde, küme kuyruğu yöneticileri arasında bilgi paylaşımı ve bağlantılık düzeyi artar. Yayınlama/abone olma kümelerine ilişkin en iyi uygulamaları izleyebilmek için, bu değişikliğin davranıştaki etkilerini tam olarak anlamanız gerekir.

En iyi mimariyi, kesin gereksinimlerinize göre oluşturmanızı sağlamak için, yayınlama/abone olma kümelerinde bilgi paylaşımı ve yayın yönlendirmesi için iki model vardır: *doğrudan yönlendirme* ve *konu anasistem yönlendirmesi*. Doğru seçimi yapmak için her iki modeli ve her bir modelin karşıladığı farklı gereksinimleri anlamanız gerekir. Bu gereksinimler, [“Dağıtılmış yayınlama/abone olma ağının planlanması” sayfa 69](#) ile birlikte aşağıdaki bölümlerde ele alınmıştır:

- [“Yayınlama/abone olma etkinliğine dahil olan küme kuyruğu yöneticilerinin sayısını sınırlama nedenleri” sayfa 89](#)
- [“Hangi konuların kümeleyeceğine karar verme” sayfa 90](#)
- [“Sisteminizin boyutunun nasıl ölçüleceğini” sayfa 90](#)
- [“Yayınlayıcı ve abonelik konumu” sayfa 91](#)
- [“Yayın trafiği” sayfa 92](#)
- [“Abonelik değişikliği ve dinamik konu dizgileri” sayfa 92](#)

Yayınlama/abone olma etkinliğine dahil olan küme kuyruğu yöneticilerinin sayısını sınırlama nedenleri

Bir kümede yayınlama/abone olma ileti sistemini kullandığınızda kapasite ve performans konuları vardır. Bu nedenle, en iyi uygulama, kuyruk yöneticileri arasında yayınlama/abone olma gereksinmesini dikkatle göz önünde bulundurmak ve etkinliği yalnızca bunu gerektiren kuyruk yöneticilerinin sayısı ile sınırlamaktır. Konuları yayınlaması ve bunlara abone olması gereken kuyruk yöneticisi kümesi alt sınırı belirlendikten sonra, bunlar yalnızca bunları içeren ve başka kuyruk yöneticisi içermeyen bir kümenin üyeleri yapılabilir.

Bu yaklaşım özellikle, noktadan noktaya ileti sistemi için düzgün çalışan bir küme oluşturmuşsanız kullanışlıdır. Var olan bir büyük kümeyi bir yayınlama/abone olma kümesine dönüştürmeye çalıştığınızda, yayınlama/abone olma işi için, yürürlükteki kümeyi kullanmak yerine, uygulamaların deneneceği ayrı bir küme yaratmak daha iyi bir uygulamadır. Zaten bir ya da daha fazla noktadan noktaya kümede olan var olan kuyruk yöneticilerinin bir alt kümesini kullanabilir ve bu alt kümeyi yeni yayınlama/abone olma kümesinin üyeleri yapabilirsiniz. Ancak, yeni kümenin tam havuz kuyruğu yöneticileri başka bir kümenin üyesi olmamalıdır; bu, var olan kümenin tam havuzlarından ek yükü yalıtır.

Yeni bir küme oluşturamazsanız ve var olan bir büyük kümeyi yayınlama/abone olma kümesine dönüştürmeniz gerekirse, doğrudan yöneltilen bir model kullanmayın. Konu anasistem tarafından yönlendirilen model genellikle daha büyük kümelerde daha iyi çalışır, çünkü genellikle yayınlama/abone olma bilgi paylaşımını ve yayınlama/abone olma işini etkin olarak gerçekleştiren kuyruk yöneticileri kümesiyle bağlantılıdır. Kısıtlar, konuları barındıran kuyruk yöneticilerine odaklanır. Bu kural dışı durum, konu tanımlamasını bulduran bir kuyruk yöneticisinde abonelik bilgilerinin el ile yenilenmesi çağrılırsa, bu durumda konu anasistem kuyruk yöneticisi kümedeki her kuyruk yöneticisine bağlanır. Bkz. [Yetkili sunucu aboneliklerinin yeniden eşzamanlanması](#).

Bir kümenin boyutu ya da yürürlükteki yükü nedeniyle yayınlama/abone olma için kullanılamayacağı belirtilerseniz, bu kümenin beklenmedik bir şekilde bir yayınlama/abone olma kümesine dönüşmesini

önlemek iyi bir uygulamadır. Kümedeki herhangi bir kuyruk yöneticisine kümeli konu eklemeyi durdurmak için **PSCLUS** kuyruk yöneticisi özelliğini kullanın. Bkz. "Kümelenmiş yayınlama/abone olmayı engelleme" sayfa 99.

Hangi konuların kümeleyeceğine karar verme

Kümeye hangi konuların ekleneceğini dikkatli bir şekilde seçmek önemlidir: Bu konular konu ağacının yukarısı ne kadar yüksekse, kullanımları o kadar yaygınlaşır. Bu, abonelik bilgilerinin ve yayınların gerekenden daha fazla yayılmasına neden olabilir. Konu ağacının, bazılarının kümelenmesi gerekirken bazılarının kümelenmesi gereken birden çok ayrı dalı varsa, kümelenmesi gereken her dalın kökünde denetlenen konu nesnelere yaratın ve bunları kümeye ekleyin. Örneğin, /A, /B ve /C dallarının kümelenmesi gerekiyorsa, her dal için ayrı bir kümelenmiş konu nesnelere tanımlayın.

Not: Sistem, konu ağacında kümelenmiş konu tanımlamalarını içiçe yerleştirmenizi önerir. Konuları, her alt dal için konu ağacında yalnızca bir noktada kümeleme izniniz vardır. Örneğin, /A ve /A/B için kümelenmiş konu nesnelere tanımlayamazsınız. Kümelenmiş konuların içiçe yerleştirilmesi, özellikle abonelikler genel arama karakterleri kullandığında hangi kümelenmiş nesnenin hangi abonelik için geçerli olduğu konusunda karışıklığa neden olabilir. Bu, yönlendirme kararlarının konu anasistemlerini ayırmanızla kesin olarak tanımlandığı konu anasistemi yönlendirmesi kullanılırken daha da önemlidir.

Kümelenmiş konuların konu ağacına eklenmesi gerekiyorsa, ancak ağacın kümelenmiş noktanın altındaki bazı dalları kümelenmiş davranışı gerektirmiyorsa, abonelik ve yayın kapsamı özneliklerini kullanarak diğer konular için abonelik ve yayın paylaşımı düzeyini azaltabilirsiniz.

Görülen davranışı dikkate almadan, konu kök düğümünü kümeye koymamalısınız. Örneğin, konu dizisinde üst düzey bir niteleyici kullanarak, genel konuları mümkün olan yerlerde açık hale getirebilirsiniz: /global ya da /cluster.

Kök konu düğümünü kümelenmiş yapmak istememenin başka bir nedeni vardır. Bunun nedeni, her kuyruk yöneticisinin kök düğüm (SYSTEM . BASE . TOPIC konu nesnesi) için yerel bir tanımlaması olmasıdır. Bu nesne kümedeki bir kuyruk yöneticisinde kümelenildiğinde, diğer tüm kuyruk yöneticilerinin bundan haberi olur. Ancak, aynı nesnenin yerel tanımlaması varsa, özellikleri küme nesnesini geçersiz kılar. Bu, kuyruk yöneticilerinin konu kümelenmemiş gibi davranmalarına neden olur. Bu sorunu çözmek için, SYSTEM . BASE . TOPIC' in her tanımını kümelemeniz gerekir. Bunu, doğrudan yöneltilen tanımlamalar için, ancak konu anasistemi tarafından yöneltilen tanımlamalar için değil, her kuyruk yöneticisinin konu anasistemi olmasına neden olduğu için yaparsınız.

Sisteminizin boyutunun nasıl ölçüleceğini

Yayınlama/abone olma kümeleri genellikle bir kümedeki noktadan noktaya ileti sistemine farklı bir küme kanalları kalıbıyla sonuçlanır. Noktadan noktaya iletişim modeli bir 'opt in' modelidir, ancak yayınlama/abone olma kümeleri, özellikle doğrudan yönlendirilmiş konular kullanıldığında, abonelik dağılımıyla daha ayırım gözetmeyen bir yapıya sahip. Bu nedenle, bir yayınlama/abone olma kümesindeki hangi kuyruk yöneticilerinin diğer kuyruk yöneticilerine bağlanmak için küme kanallarını kullanacağını ve hangi koşullar altında kullanacağını belirlemek önemlidir.

Aşağıdaki çizelge, yayınlama/abone olma kümesindeki kuyruk yöneticisi rolüne bağlı olarak, olağan çalışma altındaki bir yayınlama/abone olma kümesinde her kuyruk yöneticisi için beklenen tipik küme gönderen ve alıcı kanalları kümesini listeler.

Çizelge 5. Her yönlendirme yöntemi için küme gönderen ve alıcı kanalları.				
Kuyruk yöneticisi rolü	Doğrudan küme alıcıları	Doğrudan küme gönderenleri	Konu küme günlük nesnelere	Konu küme gönderenleri
Tam havuz	AllQmgrs	AllQmgrs	AllQmgrs	AllQMGRS
Konu tanımlamasının anasistemi	geçerli değil	geçerli değil	AllSubs+AllPubs (1)	AllSubs (1)

Çizelge 5. Her yönlendirme yöntemi için küme gönderen ve alıcı kanalları. (devamı var)

Kuyruk yöneticisi rolü	Doğrudan küme alıcıları	Doğrudan küme gönderenleri	Konu küme günlük nesnelere	Konu küme gönderenleri
Abonelikler oluşturuldu	AllPubs (1)	AllQMgrs	AllHosts	AllHosts
Yayıncılar bağlandı	AllSubs (1)	AllSubs (1)	AllHosts	AllHosts
Yayıncı ya da abone yok	AllSubs (1)	Yok (1)	Yok (2)	Yok (2)

Anahtar:

AllQMgrs

Kümedeki her kuyruk yöneticisine/yöneticisine bir kanal.

AllSubs

Aboneliğin yaratıldığı her kuyruk yöneticisine/yöneticisinden bir kanal.

AllPubs

Yayınlama uygulamasının bağlı olduğu her kuyruk yöneticisine/yöneticisinden bir kanal.

AllHosts

Kümelenmiş konu nesnesinin tanımının yapılandırıldığı her kuyruk yöneticisine bir kanal.

Yok

Yalnızca ileti alışverişi yayınlamak/abone olmak amacıyla kümedeki diğer kuyruk yöneticilerine yönelik ya da bu yöneticilerden gelen herhangi bir kanal yoktur.

Notlar:

1. Bu kuyruk yöneticisinden yetkili sunucu abonelikleri için bir kuyruk yöneticisi yenilemesi yapılırsa, kümedeki diğer tüm kuyruk yöneticilerine bir kanal otomatik olarak yaratılabilir.
2. Bu kuyruk yöneticisinden yetkili sunucu aboneliklerine ilişkin bir kuyruk yöneticisi yenilemesi yapıldıysa, kümelenmiş bir konunun tanımlamasını barındıran kümedeki diğer kuyruk yöneticilerine ilişkin bir kanal otomatik olarak yaratılabilir.

Önceki çizelge, konu anasistemi yöneltmenin genellikle doğrudan yöneltmeye göre çok daha az küme gönderen ve alıcı kanalı kullandığını gösterir. Kanal bağlantılılığı, bir kümedeki belirli kuyruk yöneticileri için bir sorunsu, kapasite veya belirli kanallar oluşturma yeteneği (örneğin, güvenlik duvarları aracılığıyla) nedeniyle, konu anasistem yönlendirmesi tercih edilen bir çözümdür.

Yayıncı ve abonelik konumu

Kümelenmiş yayınlama/abone olma, bir kuyruk yöneticisinde yayınlanan iletilerin kümedeki diğer herhangi bir kuyruk yöneticisindeki aboneliklere teslim edilmesini sağlar. Noktadan noktaya ileti sistemine gelince, kuyruk yöneticileri arasında ileti iletme maliyeti performansa zarar verebilir. Bu nedenle, iletilerin yayınlandığı aynı kuyruk yöneticilerindeki konulara abonelik yaratmayı düşünmeniz gerekir.

Bir küme içinde konu anasistemi yönlendirmesini kullanırken, kuyruk yöneticilerini barındıran konu ile ilgili olarak aboneliklerin ve yayıncıların konumunu göz önünde bulundurmanız da önemlidir. Yayıncı, kümelenmiş konunun anasistemi olan bir kuyruk yöneticisine bağlanmadığında, yayınlanan iletiler her zaman kuyruk yöneticisini barındıran bir konuya gönderilir. Benzer şekilde, kümelenmiş bir konu için konu anasistemi olmayan bir kuyruk yöneticisinde bir abonelik yaratıldığında, kümedeki diğer kuyruk yöneticilerinden yayınlanan iletiler her zaman önce bir konu barındırma kuyruk yöneticisine gönderilir. Daha spesifik olarak, abonelik konuyu barındıran bir kuyruk yöneticisinde bulunuyorsa, ancak aynı konuyu barındıran bir ya da daha fazla kuyruk yöneticisi daha varsa, diğer kuyruk yöneticilerinden gelen yayınların bir kısmı diğer konu barındırma kuyruk yöneticileri aracılığıyla yönlendirilir. Yayıncılar ve abonelikler arasındaki mesafeyi en aza indirmek üzere konu anasistemi yönlendirmeli yayınlama/abone olma kümesi tasarlamaya ilişkin ek bilgi için [Merkezleştirilmiş yayıncıları ya da aboneleri kullanarak konu anasistem yönlendirmesi](#) başlıklı konuya bakın.

Yayın trafiği

Bir kümedeki bir kuyruk yöneticisine bağlı bir uygulama tarafından yayınlanan iletiler, küme gönderen kanalları kullanılarak diğer kuyruk yöneticilerindeki aboneliklere iletilir.

Doğrudan yöneltme kullandığınızda, yayınlanan iletiler kuyruk yöneticileri arasındaki en kısa yolu alır. Yani, doğrudan yayınlama kuyruğu yöneticisinden abonelikleri olan her kuyruk yöneticisine gider. İletiler, konu için abonelikleri olmayan kuyruk yöneticilerine iletilmez. Bkz. [Bir yayınlama/abone olma ağındaki yetkili sunucu abonelikleri](#).

Kümedeki herhangi bir kuyruk yöneticisi ile başka bir kuyruk yöneticisi arasındaki yayın iletisi oranı yüksekse, bu iki nokta arasındaki küme kanalı altyapısının hızı koruyabilmesi gerekir. Bu, kullanılmakta olan kanalların ve iletim kuyruğunun ayarlanmasını içerebilir.

Konu anasistemi yöneltme özelliğini kullandığınızda, konu anasistemi olmayan bir kuyruk yöneticisinde yayınlanan her ileti bir konu anasistem kuyruk yöneticisine iletilir. Bu, kümede başka bir yerde bir ya da daha fazla aboneliğe sahip olup olmadığından bağımsızdır. Bu, planlamada göz önünde bulundurulması gereken diğer etkenleri de ortaya çıkarır:

- Her yayını bir konu anasistem kuyruk yöneticisine göndermenin ek gecikme süresi kabul edilebilir mi?
- Her konu anasistem kuyruk yöneticisi gelen ve giden yayın hızını sürdürebilir mi? Birçok farklı kuyruk yöneticisindeki yayıncıları olan bir sistemi düşünün. Hepsini mesajlarını çok küçük bir konu barındırma kuyruk yöneticilerine gönderirse, bu konu anasistemleri bu iletilerin işlenmesinde ve abone olunan kuyruk yöneticilerine yönlendirilmesinde bir darboğaza dönüşebilir.
- Yayınlanan iletilerin önemli bir bölümünün eşleşen bir aboneye sahip olmaması bekleniyor mu? Bu durumda ve bu tür iletileri yayınlama hızı yüksekse, yayıncının kuyruk yöneticisini konu anasistemi yapmak en iyisi olabilir. Bu durumda, kümede herhangi bir abonelik olmayan yayınlanmış iletiler başka bir kuyruk yöneticilerine iletilmez.

Bu sorunlar, yayın yükünü bunlara dağıtmak için birden çok konu anasistemi tanıtılarak da hafifletilebilir:

- Her biri yayın trafiğinin oranına sahip birden çok farklı konu olduğunda, bunları farklı kuyruk yöneticilerine barındırmayı düşünün.
- Konular farklı konu anasistemlerine ayrılmıyorsa, aynı konu nesnesini birden çok kuyruk yöneticisine tanımlamayı düşünebilirsiniz. Bu, yayınların yönlendirme için her birinde iş yükünün dengelenmesine neden olur. Ancak, bu yalnızca yayın iletisi sıralaması gerekli olmadığında uygundur.

Abonelik değişikliği ve dinamik konu dizgileri

Diğer bir dikkat edilmesi gereken nokta, yedek sunucu aboneliklerinin yayılması için sistemin başarımı üzerindeki etkidir. Genellikle bir kuyruk yöneticisi, belirli bir kümelenmiş konu dizesi için ilk abonelik (yalnızca yapılandırılmış bir konu nesnesi değil) o kuyruk yöneticisinde yaratıldığında kümedeki diğer bazı kuyruk yöneticilerine bir yetkili sunucu aboneliği iletisi gönderir. Benzer şekilde, belirli bir kümelenmiş konu dizisine ilişkin son abonelik silindiğinde bir yetkili sunucu aboneliği silme iletisi gönderilir.

Doğrudan yöneltme için, abonelikleri olan her kuyruk yöneticisi bu yetkili sunucu aboneliklerini kümedeki diğer her kuyruk yöneticisine gönderir. Konu anasistem yönlendirmesi için, abonelikleri olan her kuyruk yöneticisi, yalnızca bu kümelenmiş konu için bir tanımlama barındıran her kuyruk yöneticisine yetkili sunucu abonelikleri gönderir. Bu nedenle, doğrudan yönlendirmeye, kümede ne kadar çok kuyruk yöneticisi varsa, bu yöneticiler arasında yetkili sunucu aboneliklerini koruma yükü o kadar yüksektir. Bununla birlikte, konu anasistemi yönlendirmesiyle, kümedeki kuyruk yöneticilerinin sayısı bir faktör değildir.

Her iki yöneltme modelinde de, bir yayınlama/abone olma çözümü abone olunan birçok benzersiz konu dizisinden oluşuyorsa ya da kümedeki bir kuyruk yöneticisindeki konular sık sık abone olunup abonelikten çıkarılırsa, söz konusu kuyruk yöneticisinde sürekli olarak yetkili sunucu aboneliklerini dağıtan ve silen iletiler oluşturulmasından kaynaklanan önemli bir ek yük görülecektir. Bu, doğrudan yöneltmeyle, bu iletilerin kümedeki her kuyruk yöneticisine gönderilmesi gereksinimiyle birleştirilir.

Anasistem tarafından yönlendirilen bir sistem içinde bile aboneliklerin değiştirilme oranı çok yüksekse, yetkili sunucu aboneliği ek yükünü azaltmanın yolları hakkında bilgi için [Yayınlama/abone olma ağlarında abonelik performansı](#) başlıklı konuya bakın.

Küme konularının tanımlanması

Küme konuları, **cluster** özneliği tanımlanmış yönetimle ilgili konulardır. Küme konularına ilişkin bilgiler bir kümenin tüm üyelerine gönderilir ve birden çok kuyruk yöneticisiyle ilgili bir konu alanının bölümlerini yaratmak için yerel konularla birleştirilir. Bu, bir kuyruk yöneticisinde bir konuda yayınlanan iletilerin kümedeki diğer kuyruk yöneticilerinin aboneliklerine teslim edilmesini sağlar.

Bir kuyruk yöneticisinde küme konusu tanımladığınızda, küme konusu tanımlaması tam havuz kuyruğu yöneticilerine gönderilir. Daha sonra tüm havuzlar küme konusu tanımlamasını küme içindeki tüm kuyruk yöneticilerine geçirerek, aynı küme konusunu kümedeki herhangi bir kuyruk yöneticisindeki yayıncılar ve aboneler için kullanılabilir hale getirir. Küme konusu yarattığınız kuyruk yöneticisi, küme konusu anasistemi olarak bilinir. Küme konusu kümedeki herhangi bir kuyruk yöneticisi tarafından kullanılabilir, ancak bir küme konusunda yapılan değişiklikler, bu konunun tanımlandığı (anasistem) kuyruk yöneticisinde yapılmalıdır; bu noktada değişiklik, tam havuzlar aracılığıyla kümenin tüm üyelerine yayılır.

Doğrudan yöneltme kullandığınızda, kümelenmiş konu tanımlamasının yeri sistemin davranışını doğrudan etkilemez; kümedeki tüm kuyruk yöneticileri konu tanımlamasını aynı şekilde kullanır. Bu nedenle, konu gerekli olduğu sürece kümenin üyesi olacak herhangi bir kuyruk yöneticisinde konuyu tanımlamanız gerekir ve bu, tam havuz kuyruğu yöneticileriyle düzenli olarak iletişim kuracak kadar güvenilir bir sistemde yer almalıdır.

Konu anasistemi yönlendirmesini kullandığınızda, kümedeki diğer kuyruk yöneticileri bu kuyruk yöneticisine kanallar oluşturduğundan ve bu kuyruk yöneticisine abonelik bilgileri ve yayınlar gönderdiğinden, kümelenmiş konu tanımlamasının yeri çok önemlidir. Konu tanımlamasını barındıracak en iyi kuyruk yöneticisini seçmek için, konu anasistemi yöneltmesini anlamanız gerekir. Bkz. [“Konu Yayınlama/abone olma kümelerinde anasistem yönlendirmesi” sayfa 79.](#)

Kümelenmiş bir konunuz ve yerel bir konu nesneniz varsa, yerel konu öncelikli olur. Bkz. [“Aynı ada sahip birden çok küme konusu tanımlaması” sayfa 95.](#)

Küme konularını görüntülemek için kullanılacak komutlara ilişkin bilgi için ilgili bilgilere bakın.

Kümelenmiş konu kalıtımı

Genellikle, kümelenmiş bir yayınlama/abone olma topolojisinde uygulamaları yayınlamak ve abone olmak, bağlı oldukları kümede hangi kuyruk yöneticisine bağlı olursa olsun, aynı çalışmayı bekler. Bu nedenle, kümelenmiş denetlenen konu nesnelere kümedeki her kuyruk yöneticisine yayılır.

Yönetilen bir konu nesnesi, davranışını konu ağacında daha üst düzeydeki diğer yönetilen konu nesnelere devralır. Bu kalıtım, bir konu parametresi için belirtik bir değer ayarlanmadığında oluşur.

Kümelenmiş yayınlama/abone olma durumunda, yayıncıların ve abonelerin hangi kuyruk yöneticisine bağlandıklarına bağlı olarak farklı davranma olasılığını ortaya koyduğundan, bu tür bir kalıtımı göz önünde bulundurmaktır önemlidir. Kümelenmiş bir konu nesnesi, daha yüksek konu nesnelere devralmak için herhangi bir parametreden ayrılırsa, konu kümedeki farklı kuyruk yöneticilerine farklı davranabilir. Benzer şekilde, konu ağacında kümelenmiş bir konu nesnesinin altında tanımlanan yerel olarak tanımlanan konu nesnelere, bu alt konuların hala kümelenmiş olduğu anlamına gelir, ancak yerel nesne, kümedeki diğer kuyruk yöneticilerinden farklı bir şekilde davranışını değiştirebilir.

Genel arama karakteri abonelikleri

Yetkili sunucu abonelikleri, kümelenmiş bir konu nesnesinde ya da altında çözülen bir konu dizisine yerel abonelikler yapıldığında oluşturulur. Konu sıradüzeninde herhangi bir küme konusundan daha yüksek bir genel arama karakteri aboneliği yapıldıysa, bu abonelik, eşleşen küme konusu için küme çevresinde gönderilen yetkili sunucu aboneliklerine sahip değildir ve bu nedenle kümenin diğer üyelerinden herhangi bir yayın almaz. Ancak, yerel kuyruk yöneticisinden yayınlar alır.

Ancak, küme konusuna ya da altında çözülen bir konu dizisine başka bir uygulama abone olursa, yetkili sunucu abonelikleri oluşturulur ve yayınlar bu kuyruk yöneticisine yayılır. Varişta, daha yüksek joker

karakter aboneliği, bu yayınların geçerli bir alıcısı olarak kabul edilir ve bir kopyasını alır. Bu davranış gerekli değilse, kümelenmiş konuda **WILDCARD (BLOCK)** değerini ayarlayın. Bu, özgün genel arama karakterinin geçerli bir abonelik olarak kabul edilmemesini sağlar ve küme konusu ya da alt konularına ilişkin herhangi bir yayını (yerel ya da kümenin başka bir yerinden) almasını durdurur.

İlgili kavramlar

Yönetimle ilgili konularla çalışma

Aboneliklerle çalışma

İlgili başvurular

DISPLAYTOPIC

DISPLAYTPSTATUS

DISPLAYSUB

Küme konusu öznitelikleri

Bir konu nesnesi küme adı özniteliği ayarlandığında, konu tanımlaması kümedeki tüm kuyruk yöneticilerine yayılır. Her kuyruk yöneticisi, yayınlama/abone olma uygulamalarının davranışını denetlemek için yayılan konu özniteliklerini kullanır.

Bir konu nesnesi, yayınlama/abone olma kümeleri için geçerli olan bir dizi özniteliğe sahiptir. Bazıları yayınlama ve abone olma uygulamalarının genel davranışını ve bazı konular kümede nasıl kullanıldığını denetler.

Kümelenmiş bir konu nesnesi tanımlaması, kümedeki tüm kuyruk yöneticilerinin doğru olarak kullanabileceği şekilde yapılandırılmalıdır.

Örneğin, yönetilen abonelikler için kullanılacak model kuyrukları (MDURMDL ve MNDURMDL) Varsayılan olmayan bir kuyruk adına ayarlanır; adlandırılmış model kuyruğu, yönetilen aboneliklerin yaratılacağı tüm kuyruk yöneticilerine tanımlanmalıdır.

Benzer şekilde, herhangi bir öznitelik ASPARENTolarak ayarlanırsa, konunun davranışı konu ağacındaki daha yüksek düğümlere bağlıdır (bkz. Yönetimle ilgili konu nesneleri) Kümedeki her bir kuyruk yöneticisinde. Bu, farklı kuyruk yöneticilerinden yayınlama ya da abone olma sırasında farklı davranışlara neden olabilir.

Küme genelinde yayınlama/abone olma davranışıyla doğrudan ilişkili ana öznitelikler şunlardır:

ÇALIŞ.

Bu parametre, yayıncıların bağlı olduğu kuyruk yöneticileri ile eşleşen aboneliklerin bulunduğu kuyruk yöneticileri arasındaki iletilerin yönlendirilmesini denetler.

- Rotayı, bu kuyruk yöneticileri arasında doğrudan olacak şekilde ya da kümelenmiş konunun tanımlamasını barındıran bir kuyruk yöneticisi aracılığıyla yapılandırabilirsiniz. Daha fazla ayrıntı için bkz. Kümelerin yayınlanması/abone olunması .
- **CLUSTER** parametresi ayarlanırken **CLROUTE** değiştirgesini değiştiremezsiniz. **CLROUTE**değerini değiştirmek için öncelikle **CLUSTER** özelliğini boş olarak ayarlayın. Bu, konuyu kullanan uygulamaların kümelenmiş bir şekilde davranmasını önler. Bu da yayınların aboneliklere teslim edilmesine neden olur, bu nedenle değişikliği yaparken yayınlama/abone olma mesajını da susturmanız gerekir.

PROXYSUB

Bu parametre, yetkili sunucu aboneliklerinin ne zaman yapıldığını denetler.

- **FIRSTUSE** varsayılan değerdir ve dağıtılmış yayınlama/abone olma topolojisindeki bir kuyruk yöneticisindeki yerel aboneliklere yanıt olarak yetkili sunucu aboneliklerinin gönderilmesine neden olur ve artık gerekli olmadığında iptal edilir. Bu özniteliği neden **FIRSTUSE**varsayılan değerinden değiştirmek isteyebileceğinize ilişkin ayrıntılar için Bireysel yetkili sunucu aboneliği iletme ve her yerde yayınlama başlıklı konuya bakın.
- *Her yerde yayınlama* seçeneğini etkinleştirmek için, **PROXYSUB** parametresini üst düzey bir konu nesnesi için **FORCE** olarak ayarlayın. Bu, konu ağacında bu konu nesnesinin altındaki tüm konularla eşleşen tek bir genel arama karakteri yetkili sunucu aboneliğiyle sonuçlanır.

Not: Büyük ya da meşgul bir yayınlama/abone olma kümesinde **PROXYSUB (FORCE)** özniteliğinin ayarlanması, sistem kaynaklarında aşırı yüke neden olabilir. **PROXYSUB (FORCE)** özniteliği, yalnızca konunun tanımlandığı kuyruk yöneticisine değil, her kuyruk yöneticisine yayılır. Bu, kümedeki her kuyruk yöneticisinin genel arama karakterli bir yetkili sunucu aboneliği yaratmasına neden olur.

Bu konuya ilişkin, kümedeki herhangi bir kuyruk yöneticisinde yayınlanan bir iletinin kopyası, **CLROUTE** ayarına bağlı olarak, kümedeki her kuyruk yöneticisine doğrudan ya da bir konu anasistem kuyruk yöneticisi aracılığıyla gönderilir.

Konu doğrudan yöneltildiğinde, her kuyruk yöneticisi diğer her kuyruk yöneticisine küme gönderen kanalları oluşturur. Konu konu anasistemi yönlendirildiğinde, her konu anasistem kuyruk yöneticisine ilişkin kanallar kümedeki her kuyruk yöneticisinden yaratılır.

Kümelere kullanıldığında **PROXYSUB** parametresi hakkında daha fazla bilgi için bkz. [Doğrudan yönlendirilmiş yayınlama/abone olma performansı](#).

PUBSCOPE ve SUBSCOPE

Bu değiştirgeler, bu kuyruk yöneticisinin yayınları topolojideki kuyruk yöneticilerine (yayınlama/abone olma kümesi ya da sıradüzeni) yayıp yaymayacağını ya da kapsamı yalnızca yerel kuyruk yöneticisiyle sınırlayıp sınırlamayacağını belirler. MQPMO_SCOPE_QMGR ve MQSO_SCOPE_QMGR kullanarak eşdeğer işi programlı olarak yapabilirsiniz.

PUBSCOPE

Bir küme konusu nesnesi **PUBSCOPE (QMGR)** ile tanımlanmışsa, tanımlama kümeyle paylaşılır, ancak bu konuya dayalı yayınların kapsamı yalnızca yereldir ve kümedeki diğer kuyruk yöneticilerine gönderilmez.

SUBSCOPE

Bir küme konusu nesnesi **SUBSCOPE (QMGR)** ile tanımlandıysa, tanımlama kümeyle paylaşılır, ancak bu konuya dayalı aboneliklerin kapsamı yalnızca yerel olur; bu nedenle, kümedeki diğer kuyruk yöneticilerine yetkili sunucu aboneliği gönderilmez.

Bu iki öznitelik genellikle, bir kuyruk yöneticisini belirli konularda kümenin diğer üyeleriyle etkileşimde bulunmaktan yalıtılmak için birlikte kullanılır. Kuyruk yöneticisi, bu konularla ilgili yayınları kümenin diğer üyelerinden ya da üyelerinden yayınlamaz ya da almaz. Konu nesnelere alt konularda tanımlandıysa, bu durum yayınlanmayı ya da aboneliği önlemez.

Bir konunun yerel tanımında **SUBSCOPE** ögesinin QMGR olarak ayarlanması, **SUBSCOPE (ALL)** ile konunun kümelenmiş bir sürümünü kullanıyorlarsa, kümedeki diğer kuyruk yöneticilerinin yetkili sunucu aboneliklerini kuyruk yöneticisine yaymalarını engellemez. Ancak, yerel tanımlama **PUBSCOPE** değerini QMGR olarak da ayarlarsa, bu yetkili sunucu aboneliklerine bu kuyruk yöneticisinden yayınlar gönderilmez.

İlgili kavramlar

[Yayın kapsamı](#)

[Abonelik kapsamı](#)

Aynı ada sahip birden çok küme konusu tanımlaması

Kümede birden çok kuyruk yöneticisinde aynı adlandırılmış küme konusu nesnesini tanımlayabilirsiniz ve bazı senaryolarda bu belirli bir davranışı etkinleştirir. Aynı ada sahip birden çok küme konusu tanımlaması varsa, özelliklerin çoğunluğu eşleşmelidir. Yoksa, uyumsuzluğun önemine bağlı olarak hatalar ya da uyarılar bildirilir.

Genel olarak, birden çok küme konusu tanımlamasının özelliklerinde bir uyumsuzluk varsa, uyarılar yayınlanır ve konu nesnesi tanımlamalarından biri kümedeki her kuyruk yöneticisi tarafından kullanılır. Her kuyruk yöneticisi tarafından kullanılan tanımlama belirleyici değil ya da kümedeki kuyruk yöneticileri arasında tutarlı değil. Bu tür uyumsuzluklar mümkün olduğunca çabuk çözülmelidir.

Küme kuruluşu ya da bakımı sırasında bazen aynı olmayan birden çok küme konusu tanımlaması yaratmanız gerekir. Ancak bu yalnızca geçici bir önlem olarak kullanışlıdır ve bu nedenle olası bir hata koşulu olarak kabul edilir.

Uyumsuzluk saptandığında, her kuyruk yöneticisinin hata günlüğüne aşağıdaki uyarı iletileri yazılır:

- **Multi** Çoklu platformlarsistemlerinde [AMQ9465](#) ve [AMQ9466](#).
- **z/OS** z/OS sistemlerinde [CSQX465I](#) ve [CSQX466I](#).

Her kuyruk yöneticisindeki herhangi bir konu dizgisi için seçilen özellikler, konu nesnesi tanımlamaları yerine konu durumu görüntülenerek (örneğin, **DISPLAY TPSTATUS** kullanılarak) belirlenebilir.

Bazı durumlarda, yapılandırma özelliklerindeki bir çakışma, yaratılmakta olan konu nesnesini durdurmak ya da eşleşmeyen nesnelerin geçersiz olarak işaretlenmesini ve kümeye yayılmamasını sağlamak için yeterince önemli olur (**DISPLAY TOPIC** içindeki **CLSTATE** konusuna bakın). Bu durumlar, küme yöneltme özelliğinde bir çakışma olduğunda oluşur (**CLROUTE**) konu tanımlarına bakın. Buna ek olarak, konu anasistem tarafından yönlendirilen tanımlarda tutarlılığın önemi nedeniyle, bu makalenin sonraki bölümlerinde ayrıntılı olarak açıklandığı gibi daha fazla tutarsızlık reddedilir.

Nesne tanımlandığında çakışma saptanırsa, yapılandırma değişikliği reddedilir. Daha sonra tam havuz kuyruğu yöneticileri tarafından algılanırsa, kuyruk yöneticilerinin hata günlüklerine aşağıdaki uyarı iletileri yazılır:

- **Multi** Çoklu platformlarsistemlerinde: [AMQ9879](#)
- **z/OS** z/OS üzerinde: [CSQX879E](#).

Kümeye aynı konu nesnesine ilişkin birden çok tanımlama tanımlandığında, yerel olarak tanımlanan bir tanımlama, uzaktan tanımlanan herhangi bir tanımlamadan önceliklidir. Bu nedenle, tanımlarda herhangi bir farklılık varsa, birden çok tanımlama barındıran kuyruk yöneticileri birbirinden farklı davranır.

Başka bir kuyruk yöneticisinden küme konusuyla aynı ada sahip küme dışı bir konu tanımlama etkisi

Kümeye bulunan bir kuyruk yöneticisinde kümelenmemiş denetlenen bir konu nesnesi tanımlanabilir ve aynı adı taşıyan konu nesnesini, farklı bir kuyruk yöneticisinde kümelenmiş konu tanımlaması olarak eşzamanlı olarak tanımlayabilir. Bu durumda, yerel olarak tanımlanan konu nesnesi, aynı adı taşıyan tüm uzak tanımlamalardan önceliklidir.

Bu, bu kuyruk yöneticisinden kullanıldığında konunun kümeleme davranışını önler. Yani, abonelikler uzak yayıncılardan yayın almayabilir ve yayıncılardan gelen iletiler kümedeki uzak aboneliklere yayılmamış olabilir.

Böyle bir sistemi yapılandırmadan önce dikkatli bir şekilde göz önünde bulundurulmalıdır, çünkü bu kafa karıştırıcı bir davranışa yol açabilir.

Not: Tek bir kuyruk yöneticisinin, konu başka bir yerde kümelenmiş olsa bile, yayınların ve aboneliklerin küme içinde yayılmasını önlemesi gerekiyorsa, diğer bir yaklaşım, yayınlama ve abonelik kapsamlarını yalnızca yerel kuyruk yöneticisine ayarlamaktır. Bkz. "[Küme konusu öznitelikleri](#)" sayfa 94.

Doğrudan yöneltile bir kümede birden çok küme konusu tanımlaması

Doğrudan yöneltme için, genellikle birden çok küme kuyruk yöneticisinde aynı küme konusunu tanımlamayın. Bunun nedeni, doğrudan yöneltmenin konuyu, hangi kuyruk yöneticisinde tanımlandıysa tanımlansın, kümedeki tüm kuyruk yöneticilerinin kullanımına açmaktır. Ayrıca, birden çok küme konusu tanımının eklenmesi, sistem etkinliğini ve yönetim karmaşıklığını önemli ölçüde artırır ve artan karmaşıklıkla birlikte insan hatası olasılığı da artar:

- Her tanımlama, diğer küme konusu anasistem kuyruk yöneticileri de içinde olmak üzere, kümedeki diğer kuyruk yöneticilerine ek bir küme konusu nesnesi gönderilmesiyle sonuçlanır.
- Bir kümedeki belirli bir konuya ilişkin tüm tanımlamaların aynı olması gerekir; tersi durumda, hangi konu tanımlamasının bir kuyruk yöneticisi tarafından kullanıldığını anlamak zordur.

Küme konusu tanımlaması tam havuz kuyruğu yöneticileri ve kısmi küme havuzlarındaki diğer tüm kuyruk yöneticileri tarafından önbelleğe alındığından, konunun küme çapında doğru çalışması için tek anasistem kuyruk yöneticisinin sürekli olarak kullanılabilir olması da önemli değildir. Ek bilgi için [Doğrudan yöneltme kullanan konu anasistem kuyruk yöneticilerinin kullanılabilirliği](#) başlıklı konuya bakın.

İkinci bir kuyruk yöneticisinde geçici olarak küme konusu tanımlamanız gerekebilecek bir durum için (örneğin, konunun var olan anasistemi kümeden kaldırılacaksa) bkz. [Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması](#).

Bir küme konusu tanımlamasını değiştirmeniz gerekiyorsa, tanımladığı kuyruk yöneticisinde değiştirmeyi dikkatli olun. Bunu başka bir kuyruk yöneticisinden değiştirme girişimi, konu öznitelikleriyle çakışan konunun ikinci bir tanımını yanlışlıkla yaratabilir.

Bir konu anasistemi tarafından yönlendirilen kümede birden çok küme konusu tanımlaması

Bir küme konusu *konu anasistemi* küme rotasıyla tanımlandığında, konu, *doğrudan* yöneltilen konular gibi kümedeki tüm kuyruk yöneticilerine yayılır. Ayrıca, bu konuya ilişkin tüm yayınlama/abone olma ileti sistemi, o konunun tanımlandığı kuyruk yöneticileri aracılığıyla yönlendirilir. Bu nedenle, kümedeki konunun yeri ve tanım sayısı önemli hale gelir (bkz. [“Konu Yayınlama/abone olma kümelerinde anasistem yönlendirmesi” sayfa 79](#)).

Yeterli düzeyde kullanılabilirlik ve ölçeklenebilirlik sağlamak için, mümkünse, birden çok konu tanımına sahip olmak yararlıdır. Bkz. [Konu anasistem yöneltmesini kullanan konu anasistem kuyruk yöneticilerinin kullanılabilirliği](#).

Bir kümede *konu anasistemi* tarafından yönlendirilen bir konuya ilişkin ek tanımlar eklerken ya da kaldırırken, yapılandırma değişikliği sırasında iletilerin akışını göz önünde bulundurmanız gerekir. Değişiklik sırasında kümede konu ile ilgili iletiler yayınlanıyorsa, konu tanımlaması eklemek ya da kaldırmak için aşamalı bir işlem gerekir. Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması ve [Konu anasistemine yöneltilen küme ek konu anasistemleri eklenmesibaşlıklı konuya bakın](#).

Daha önce açıklandığı gibi, birden çok tanımlamanın özellikleri, sonraki kısımda açıklandığı gibi, olası **PUB** değiştirgesi dışında eşleşmelidir. Yayınlar konu anasistem kuyruk yöneticileri aracılığıyla yönlendiğinde, birden çok tanımlamanın tutarlı olması daha da önemlidir. Bu nedenle, konu tanımlamalarından biri ya da daha fazlası konu anasistem kümesi yönlendirmesi için yapılandırıldıysa, konu dizisinde ya da küme adında bir tutarsızlık saptandı.

Not: Var olan kümelenmiş konu tanımlamasının konu anasistem yöneltmesi için yapılandırıldığı konu ağacında başka bir konunun üstünde ya da altında yapılandırma girişiminde bulunulursa, küme konusu tanımlamaları da reddedilir. Bu, genel arama karakterli aboneliklerle ilgili olarak yayınların yönlendirilmesinde belirsizliği önler.

PUB parametresi için özel işleme

PUB parametresi, uygulamaların bir konuya ne zaman yayınlatabileceğini denetlemek için kullanılır. Bir kümede konu anasistem yönlendirmesi olması durumunda, yayınları yönlendirmek için hangi konu anasistem kuyruk yöneticilerinin kullanılacağını da denetleyebilir. Bu nedenle, **PUB** parametresi için farklı ayarlarla kümede aynı konu nesnesine ilişkin birden çok tanımlama olmasına izin verilir.

Bir konuya ilişkin birden çok uzak kümelenmiş tanımlama bu parametre için farklı ayarlara sahipse, konu, aşağıdaki koşullar karşılandıysa yayınların gönderilmesine ve aboneliklere teslim edilmesine izin verir:

- Kuyruk yöneticisinde, **PUB (DISABLED)** olarak ayarlanan, yayınlayıcının bağlı olduğu eşleşen bir konu nesnesi tanımlı değil.
- Kümedeki birden çok konu tanımlamasının biri ya da daha fazlası **PUB (ENABLED)** olarak ayarlandı ya da birden çok konu tanımlamasının biri ya da daha fazlası **PUB (ASPARENT)** olarak ayarlandı ve yayınlayıcının bağlı olduğu yerel kuyruk yöneticileri ve tanımlanan abonelik, konu ağacının daha yüksek bir noktasında **PUB (ENABLED)** olarak ayarlandı.

Konu anasistem yöneltmesi için, iletiler konu anasistemi olmayan kuyruk yöneticilerine bağlı uygulamalar tarafından yayınlandığında, iletiler yalnızca, **PUB** değiştirgesinin belirttik olarak **DISABLED** olarak ayarlanmadığı konu barındıran kuyruk yöneticilerine yöneltilir. Bu nedenle, belirli konu anasistemleri üzerinden ileti trafiğini durdurmak için **PUB (DISABLED)** ayarını kullanabilirsiniz. Bunu, bir kuyruk yöneticisinin bakımına ya da kaldırılmasına hazırlanmak için ya da [Konu anasistem tarafından](#)

yönlendirilen bir kümeye ek konu anasistemleri eklenmesibaşlıklı konuda açıklanan nedenlerle yapmak isteyebilirsiniz.

Küme konusu anasistem kuyruk yöneticilerinin kullanılabilirliği

Yayınla/abone olma kümenizi, bir konu anasistem kuyruk yöneticisinin kullanılamaması durumunda kümenin artık konuya ilişkin trafiği işleyemeyeceği riskini en aza indirecek şekilde tasarlayın. Bir konu anasistem kuyruk yöneticisinin kullanılamaz duruma gelmesinin etkisi, kümenin konu anasistem yöneltmesini mi, yoksa doğrudan yöneltmeyi mi kullandığına bağlıdır.

Doğrudan yöneltmeyi kullanan konu anasistem kuyruk yöneticilerinin kullanılabilirliği

Doğrudan yöneltme için, genellikle birden çok küme kuyruk yöneticisinde aynı küme konusunu tanımlamayın. Bunun nedeni, doğrudan yöneltmenin konuyu, hangi kuyruk yöneticisinde tanımlandıysa tanımlansın, kümedeki tüm kuyruk yöneticilerinin kullanımına açmaktır. Bkz. Doğrudan yönelttilen bir kümede birden çok küme konusu tanımlaması.

Bir kümede, kümelenmiş bir nesnenin anasistemi (örneğin, kümelenmiş bir kuyruk ya da kümelenmiş bir konu) uzun bir süre kullanılamaz duruma geldiğinde, kümenin diğer üyeleri bu nesnelerin bilgisini sona erer. Kümelenmiş bir konu söz konusu olduğunda, küme konusu anasistem kuyruk yöneticisi kullanılamaz duruma gelirse, diğer kuyruk yöneticileri konuya ilişkin yayınla/abone olma isteklerini doğrudan kümelenmiş bir şekilde (yani, uzak kuyruk yöneticilerine aboneliklere yayın gönderme), kuyruk yöneticisinin en son havuz kuyruk yöneticileriyle iletişimde olduğu tarihten itibaren en az 60 gün boyunca işlemeye devam eder. Küme konusu nesnesini tanımladığınız kuyruk yöneticisi bir daha kullanılabilir kılınmazsa, diğer kuyruk yöneticilerindeki önbelleğe alınan konu nesnelere silinir ve konu yerel bir konuya döner; bu durumda, abonelikler uzak kuyruk yöneticilerine bağlı uygulamalardan yayın almayı durdurur.

Küme konusu nesnesi tanımladığınız kuyruk yöneticisini kurtarmak için 60 günlük süre ile, bir küme konusu anasisteminin kullanılabilir durumda kalmasını sağlamak için özel önlemler alınması gerekmez (ancak, kullanılmayan küme konusu anasisteminde tanımlanan aboneliklerin kullanılabilir kalmadığını unutmayın). 60 günlük süre teknik sorunları gidermek için yeterlidir ve yalnızca idari hatalar nedeniyle aşılması muhtemeldir. Bu olasılığı azaltmak için, küme konusu anasistemi kullanılamıyorsa, kümenin tüm üyeleri önbelleğe alınan küme konusu nesnesinin yenilenmediğini belirten hata günlüğü iletilerini saatlik olarak yazar. Küme konusu nesnesinin tanımlandığı kuyruk yöneticisinin çalıştığından emin olarak bu iletilere yanıt verin. Küme konusu anasistem kuyruk yöneticisini yeniden kullanılabilir kılmak mümkün değilse, kümedeki başka bir kuyruk yöneticisinde aynı özniteliklerle aynı kümelenmiş konu tanımlamasını tanımlayın.

Konu anasistem yöneltmesini kullanan konu anasistem kuyruk yöneticilerinin kullanılabilirliği

Konu anasistem yönlendirmesi için, bir konuya ilişkin tüm yayınla/abone olma ileti sistemi, o konunun tanımlandığı kuyruk yöneticileri aracılığıyla yönlendirilir. Bu nedenle, kümedeki bu kuyruk yöneticilerinin sürekli kullanılabilirliğini göz önünde bulundurmak çok önemlidir. Bir konu anasistemi kullanılamaz hale gelirse ve konu için başka bir anasistem yoksa, yayıncılardan farklı kuyruk yöneticilerine kadar olan trafik bu konu için hemen durdurur. Ek konu anasistemleri varsa, küme kuyruğu yöneticileri yeni yayın trafiğini bu konu anasistemleri üzerinden yönlendirerek ileti rotalarının sürekli kullanılabilirliğini sağlar.

Doğrudan konulara gelince, 60 gün sonra, ilk konu anasistemi hala kullanılamıyorsa, o konu anasisteminin konusunun bilgisi kümeden kaldırılır. Bu, kümedeki bu konu için kalan son tanımsa, diğer tüm kuyruk yöneticileri, yayınları yöneltme amacıyla herhangi bir konu anasistemine iletmeyi durdurmaktadır.

Bu nedenle, uygun kullanılabilirliği ve ölçeklenebilirliği sağlamak için, mümkünse, her konuyu en az iki küme kuyruğu yöneticisinde tanımlamak yararlı olacaktır. Bu, belirli bir konu anasistem kuyruk yöneticisinin kullanılamaz olmasına karşı koruma sağlar. Ayrıca bkz. Konu anasistem tarafından yönlendirilen kümede birden çok küme konusu tanımlaması.

Birden çok konu anasistemi yapılandıramıyorsanız (örneğin, ileti sıralamasını korumalısınız) ve yalnızca bir konu anasistemi yapılandıramıyorsanız (tek bir kuyruk yöneticisinin kullanılabilirliği, kümedeki tüm kuyruk yöneticilerindeki aboneliklere ilişkin yayınların akışını etkilememeli), konuyu doğrudan yönelttilen

bir konu olarak yapılandırmayı düşünebilirsiniz. Bu, tüm küme için tek bir kuyruk yöneticisine güvenmeyi önler, ancak yerel olarak barındırılan abonelikleri ve yayıncıları işlemesi için her bir kuyruk yöneticisinin kullanılabilir olmasını gerektirir.

Kümelenmiş yayınlama/abone olmayı engelleme

Doğrudan kümelenmiş ilk konunun bir kümede tanıtılması, kümedeki her kuyruk yöneticisinin diğer kuyruk yöneticilerinden haberdar olmasını sağlar ve potansiyel olarak bunların birbirine kanal oluşturmasına neden olur. Bu arzu edilmezse, konu anasistemi tarafından yönlendirilen yayınlama/abone olma özelliğini yapılandırmanız gerekir. Doğrudan yönlendirilmiş kümelenmiş bir konunun varlığı, her kuyruk yöneticisinin ölçekleme endişeleri nedeniyle kümenin dengesini tehlikeye atabilirse, kümedeki her kuyruk yöneticisinde **PSCLUS** ayarını DISABLED olarak ayarlayarak kümelenmiş yayınlama/abone olma işlevini tamamen geçersiz kılabilirsiniz.

“Yayınlama/abone olma kümelerinde doğrudan yönlendirme” sayfa 75’inde açıklandığı gibi, bir kümeye doğrudan yönlendirilmiş kümelenmiş bir konu sunduğunuzda, tüm kısmi havuzlara kümenin diğer tüm üyeleri otomatik olarak bildirilir. Kümelenmiş konu, diğer tüm düğümlerde de abonelikler yaratabilir (örneğin, **PROXYSUB (FORCE)** belirtilirse) ve yerel abonelik olmadığında bile çok sayıda kanalın bir kuyruk yöneticisinden başlatılmasına neden olabilir. Bu, kümedeki her kuyruk yöneticisine hemen ek bir yük getirir. Çok sayıda kuyruk yöneticisi içeren bir küme için bu, performansta önemli bir düşüşe neden olabilir. Bu nedenle, bir kümeye doğrudan yönlendirilmiş yayınlama/abone olma girişinin dikkatli bir şekilde planlanması gerekir.

Bir kümenin doğrudan yönetilen yayınlama/abone olma ek giderlerini karşılayamayacağını biliyorsanız, bunun yerine konu anasistemi yönlendirilen yayınlama/abone olma özelliğini kullanabilirsiniz. Farklılıklara genel bakış için bkz. “Yayınlama/abone olma kümeleri tasarlanması” sayfa 72.

Küme için yayınlama/abone olma işlevselliğini tamamen devre dışı bırakmayı tercih ediyorsanız, **PSCLUS** kuyruk yöneticisi özneteliğini kümedeki her kuyruk yöneticisinde DEVRE DİŞI olarak ayarlayarak bunu yapabilirsiniz. Bu ayar, kuyruk yöneticisi işlevinin üç özelliğini değiştirerek, kümede hem doğrudan yönetilen hem de konu anasistemi tarafından yönlendirilen yayınlama/abone olma özelliğini devre dışı bırakır:

- Bu kuyruk yöneticisinin bir yöneticisi artık bir Topic nesnesini kümelenmiş olarak tanımlayamaz.
- Diğer kuyruk yöneticilerinden gelen konu tanımlamaları ya da yetkili sunucu abonelikleri reddedilir ve yöneticiye yanlış yapılandırma hakkında bilgi vermek için bir uyarı iletisi kaydedilir.
- Tam havuzlar, bir konu tanımlaması aldıklarında, her kuyruk yöneticisine ilişkin bilgileri diğer tüm kısmi havuzlarla otomatik olarak paylaşmaz.

PSCLUS , bir kümedeki her bir kuyruk yöneticisinin bir parametresi olsa da, kümedeki kuyruk yöneticilerinin bir alt kümesinde yayınlama/abone olma özelliğini seçmeli olarak devre dışı bırakması amaçlanmamıştır. Bu şekilde seçmeli olarak devre dışı bırakırsanız, sık sık hata iletileri görürsünüz. Bunun nedeni, **PSCLUS** olanağının etkinleştirildiği bir kuyruk yöneticisinde bir konu kümelenirse yetkili sunucu aboneliklerinin ve konu tanımlamalarının sürekli olarak görülmesidir ve reddedilmesidir.

Bu nedenle, kümedeki her kuyruk yöneticisinde **PSCLUS** değerini DEVRE DİŞI olarak ayarlamanız gerekir. Ancak pratikte bu durumun sağlanması ve sürdürülmesi zor olabilir; örneğin, kuyruk yöneticileri herhangi bir zamanda kümeye katılabilir ve kümeden ayrılabilir. En azından, tüm havuz kuyruğu yöneticilerinizde **PSCLUS** değerinin DEVRE DİŞI olarak ayarlandığından emin olmanız gerekir. Bunu yaparsanız ve kümelenmiş bir konu kümedeki bir ETKİNLEŞTİRİLMİŞ kuyruk yöneticisinde daha sonra tanımlanırsa, bu durum tüm havuzların diğer kuyruk yöneticilerinin her kuyruk yöneticisine bilgi vermesine neden olmaz ve kümeniz tüm kuyruk yöneticilerine ilişkin olası ölçekleme sorunlarından korunur. Bu senaryoda, kümelenmiş konunun kaynağı, tam havuz kuyruğu yöneticilerinin hata günlüklerinde bildirilir.

Bir kuyruk yöneticisi bir ya da daha fazla yayınlama/abone olma kümesine ve bir ya da daha fazla noktadan noktaya kümeye katılıyorsa, o kuyruk yöneticisinde **PSCLUS** değerini ENABLED olarak ayarlamanız gerekir. Bu nedenle, bir noktadan noktaya iletişim kümesini yayınlama abone olma kümesiyle örtüştüğünde, her kümede ayrı bir tam havuz kümesi kullanmanız gerekir. Bu yaklaşım, her kuyruk yöneticisine ilişkin konu tanımlamalarının ve bilgilerin yalnızca yayınlama/abone olma kümesinde akmasına olanak sağlar.

PSCLUS değerini ETKİNLEŞTİRİLMİŞ olarak DEVRE Dışı olarak değiştirdiğinizde tutarsız yapılandırmaları önlemek için, bu kuyruk yöneticisinin üyesi olduğu herhangi bir kümede kümelenmiş konu nesnelere bulunamaz. Bu tür konular, uzaktan tanımlanmış konular da dahil olmak üzere, **PSCLUS** değeri DEVRE Dışı olarak değiştirilmeden önce silinmelidir.

PSCLUS ile ilgili ek bilgi için bkz. [ALTER QMGR \(PSCLUS\)](#).

İlgili kavramlar

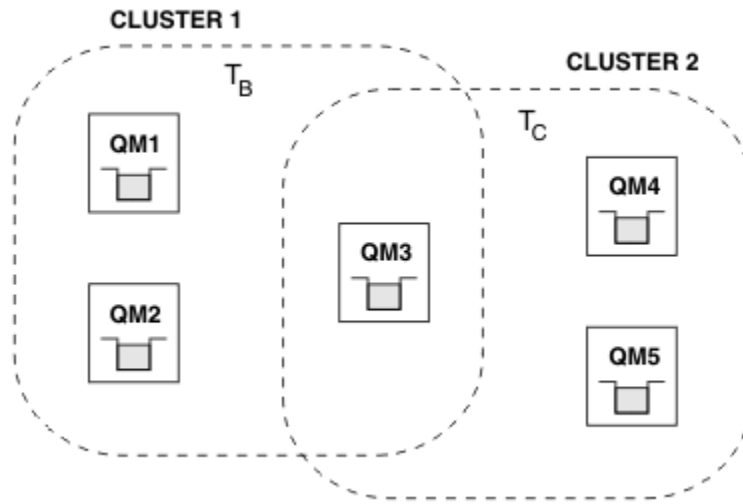
[Doğrudan yöneltilen yayınlama/abone olma kümesi performansı](#)

Yayınlama/abone olma ve birden çok küme

Tek bir kuyruk yöneticisi birden çok kümenin üyesi olabilir. Bu düzenleme bazen *üst üste binen kümeler* olarak bilinir. Bu tür bir çakışma ile, kümelenmiş kuyruklar birden çok kümeden erişilebilir hale getirilebilir ve noktadan noktaya ileti trafiği bir kümedeki kuyruk yöneticilerinden başka bir kümedeki kuyruk yöneticilerine yönlendirilebilir. Yayınlama/abone olma kümelerindeki kümelenmiş konular aynı yeteneği sağlamaz. Bu nedenle, birden çok küme kullanılırken davranışlarının açıkça anlaşılması gerekir.

Bir kuyruktan farklı olarak, bir konu tanımlamasını birden çok kümeyle ilişkilendiremezsiniz. Kümelenmiş bir konunun kapsamı, konunun tanımlandığı kümede yer alan kuyruk yöneticileriyle sınırlıdır. Bu, yayınların yalnızca aynı kümedeki kuyruk yöneticilerindeki aboneliklere yayılmasına izin verir.

Bir kuyruk yöneticisinin konu ağacı



Şekil 28. Çakışan kümeler: Her biri farklı konulara abone olan iki küme

Bir kuyruk yöneticisi birden çok kümenin üyesi ise, bu kümelerin her birinde tanımlanan tüm kümelenmiş konulardan haberdar olur. Örneğin, önceki şekilde QM3, hem T_B hem de T_C denetimli kümelenmiş konu nesnelere tanıyabilir; oysa QM1 yalnızca T_B'yi bilir. QM3, her iki konu tanımını da yerel konusuna uygular ve bu nedenle, belirli konular için QM1'e farklı bir işlev uygular. Bu nedenle, farklı kümelerdeki kümelenmiş konuların birbirine karışmaması önemlidir. Parazit, farklı bir kümede (örneğin, /Sport ve /Sport/Football konu dizilerine sahip) ya da her ikisinde de aynı konu dizisi için başka bir kümelenmiş konunun üstünde ya da altında tanımlandığında ortaya çıkabilir. Başka bir müdahale biçimi, denetlenen kümelenmiş konu nesnelere farklı kümelerde aynı nesne adıyla, ancak farklı konu dizileri için tanımlanmasıdır.

Böyle bir yapılandırma yapılırsa, yayınların eşleşen aboneliklere teslim edilmesi, yayıncıların ve abonelerin kümeyle ilişkin görece konumlarına çok bağlıdır. Bu nedenle, böyle bir yapılandırmaya güvenemezsiniz ve bunu, müdahale eden konuları kaldırmak için değiştirmeniz gerekir.

Yayınlama/abone olma ileti sistemiyle çakışan bir küme topolojisi planlarken, konu ağacını ve kümelenmiş konu nesnelere adlarını topolojideki tüm çakışan kümelere yayılıyor gibi işleyerek herhangi bir parazit oluşmasını önleyebilirsiniz.

Birden çok yayınlama/abone olma kümesinin bütünleştirilmesi

Farklı kümelerdeki kuyruk yöneticilerini kapsamak için yayınlama/abone olma ileti sistemi gereksinimi varsa, iki seçenek vardır:

- Bir yayınlama/abone olma sıradüzeni yapılandırması kullanarak kümeleri birbirine bağlayın. Bkz. [Birden çok kümenin konu alanlarını birleştirme](#).
- Var olan kümelerle örtüşen ve belirli bir konuyu yayınlaması ya da bu konuya abone olması gereken tüm kuyruk yöneticilerini içeren ek bir küme oluşturun.

İkinci seçenikle, kümenin boyutunu ve en etkili küme yönlendirme mekanizmasını dikkatli bir şekilde göz önünde bulundurmalısınız. Bkz. "[Yayınlama/abone olma kümeleri tasarlanması](#)" sayfa 72.

Yayınlama/abone olma kümelerinde tutulan yayınlar için tasarımla ilgili dikkat edilecek noktalar

Alıkonan yayınlarla çalışmak için bir yayınlama/abone olma kümesi tasarlarırken dikkate alınması gereken birkaç kısıtlama vardır.

Dikkat Edilecekler

Dikkate alınacak 1: Aşağıdaki küme kuyruğu yöneticileri her zaman korunan bir yayının en son sürümünü saklar:

- Yayınlayıcının kuyruk yöneticisi
- Konu anasistemi tarafından yönlendirilen bir kümede, konu anasistemi (bu makalenin sonraki bölümünde açıklandığı gibi, konu için tek bir konu anasistemi olması koşuluyla)
- Alıkonan yayının konu dizgisiyle eşleşen abonelikleri olan tüm kuyruk yöneticileri

Dikkate alınması 2: Kuyruk yöneticileri, abonelikleri yokken güncellenmiş alıkonan yayınları almazlar. Bu nedenle, artık konuya abone olmayan bir kuyruk yöneticisinde saklanan alıkonan yayınlar eskimiş olur.

Dikkate Alınma 3: Herhangi bir abonelik oluşturulurken, konu dizesi için alıkonan bir yayının yerel bir kopyası varsa, yerel kopya aboneliğe teslim edilir. Belirli bir konu dizgisinin ilk abonesiyeniz, aşağıdaki küme üyelerinden birinden eşleşen bir alıkonan yayın da teslim edilir:

- Doğrudan yöneltile bir kümede, yayınlayıcının kuyruk yöneticisi
- Konu anasistemi tarafından yönlendirilen bir kümede, verili konuya ilişkin konu anasistemleri

Alıkonan bir yayının konu anasisteminden ya da yayınlama kuyruk yöneticisinden abone olan kuyruk yöneticisine teslim edilmesi, MQSUB çağrılarında zamanuyumsuzdur. Bu nedenle, MQSUBRQ çağrısı kullanırsanız, sonraki bir MQSUBRQ çağrısına kadar en son alıkonan yayın gözden kaçırılabilir.

Etkileri

Herhangi bir yayınlama/abone olma kümesi için, ilk abonelik yapıldığında, yerel kuyruk yöneticisi alıkonan bir yayının eski bir kopyasını saklıyor olabilir ve bu, yeni aboneliğe teslim edilen kopyadır. Yerel kuyruk yöneticisinde bir abonelik bulunması, alıkonan yayının bir sonraki güncellenişinde bunun çözüleceği anlamına gelir.

Bir konu anasistemi tarafından yönlendirilen yayınlama/abone olma kümesi için, belirli bir konu için birden çok konu anasistemi yapılandırırsanız, yeni aboneler en son tutulan yayını bir konu anasisteminden alabilir ya da başka bir konu anasisteminden (en sonuncusu kaybolmuş) eski bir yayın alabilir. Konu anasistem yöneltmesi için, belirli bir konu için birden çok konu anasistemi yapılandırılması olağandır. Ancak, uygulamaların alıkonan yayınları kullanmasını bekliyorsanız, her konu için yalnızca bir konu anasistemi yapılandırmanız gerekir.

Herhangi bir konu dizgisi için, tek bir yayınlayıcı kullanmanız ve yayınlayıcının her zaman aynı kuyruk yöneticisini kullandığından emin olmanız gerekir. Bunu yapmazsanız, alıkonan farklı yayınlar aynı konu için farklı kuyruk yöneticilerinde etkin olabilir ve bu da beklenmeyen bir davranışa yol açabilir. Birden çok yetkili sunucu aboneliği dağıtıldığından, birden çok alıkonan yayın alınabilir.

Abonelerin eski yayınları kullanmasından endişe ediyorsanız, alıkonan her yayını oluşturduğunuzda bir ileti süre bitimi ayarlamayı düşünebilirsiniz.

Alıkonan bir yayını bir yayınlama/abone olma kümesinden kaldırmak için **CLEAR TOPICSTR** komutunu kullanabilirsiniz. Belirli durumlarda, **CLEAR TOPICSTR** başlıklı konuda açıklandığı gibi, komutu yayınlama/abone olma kümesinin birden çok üyesinde vermeniz gerekebilir.

Genel arama karakteri abonelikleri ve alıkonan yayınlar

Genel arama karakteri abonelikleri kullanıyorsanız, yayınlama/abone olma kümesinin diğer üyelerine teslim edilen ilgili yetkili sunucu abonelikleri, ilk genel arama karakterinden hemen önce konu ayırıcısından genel arama karakteri olarak kullanılır. Bkz. [Genel arama karakterleri ve küme konuları](#).

Bu nedenle, kullanılan joker karakter, abone olan uygulamayla eşleşenden daha fazla konu dizesi ve daha fazla alıkonan yayınla eşleşebilir.

Bu, alıkonan yayınlar için gereken depolama miktarını artırır ve bu nedenle, barındırma kuyruğu yöneticilerinin yeterli depolama kapasitesine sahip olduğundan emin olmanız gerekir.

İlgili kavramlar

[Yayınlara alıkoyma](#)

[Tek tek yetkili sunucu aboneliğinin iletilmesi ve her yerde yayınlanması](#)

Yayınlama/abone olma kümeleri için REFRESH CLUSTER ile ilgili önemli noktalar

REFRESH CLUSTER komutunun verilmesi, kuyruk yöneticisinin küme konuları ve ilişkili yetkili sunucu abonelikleri de içinde olmak üzere, bir kümeyle ilgili yerel olarak tutulan bilgileri geçici olarak atmasına neden olur.

REFRESH CLUSTER komutunun, kuyruk yöneticisinin kümelenmiş yayınlama/abone olma için gerekli bilgileri yeniden edindiği noktaya kadar geçen süre, kümenin boyutuna, kullanılabilirliğine ve tam havuz kuyruğu yöneticilerinin yanıt verebilirliğine bağlıdır.

Yenileme işlemi sırasında, yayınlama/abone olma kümesinde trafiği yayınlama/abone olma kesintisi oluşur. Büyük kümeler için, **REFRESH CLUSTER** komutunun kullanımı devam ederken kümeyi kesintiye uğratabilir ve bundan sonra küme nesnelere otomatik olarak ilgili tüm kuyruk yöneticilerine durum güncellemeleri gönderdiğinde kümeyi 27 günlük aralıklarla yeniden bozabilir. Bkz. [Büyük bir kümede yenilenmesi, kümenin performansını ve kullanılabilirliğini etkileyebilir](#). Bu nedenlerle, **REFRESH CLUSTER** komutu yalnızca IBM Destek Merkezi 'nin rehberliğinde bir yayınlama/abone olma kümesinde kullanılmalıdır.

Kümedeki kesinti, aşağıdaki belirtiler olarak dışarıdan görünebilir:

- Bu kuyruk yöneticisindeki küme konularına ilişkin abonelikler, kümedeki diğer kuyruk yöneticilerine bağlı yayıncılardan yayın almıyor.
- Bu kuyruk yöneticisindeki küme konularına yayınlanan iletiler, diğer kuyruk yöneticilerindeki aboneliklere yayılmıyor.
- Bu dönemde yaratılan bu kuyruk yöneticisindeki küme konularına ilişkin abonelikler, kümenin diğer üyelerine sürekli olarak yetkili sunucu abonelikleri göndermez.
- Bu dönemde silinen bu kuyruk yöneticisindeki küme konularına ilişkin abonelikler, kümenin diğer üyelerinden sürekli olarak yetkili sunucu aboneliklerini kaldırmaz.
- İleti tesliminde 10 saniye ya da daha uzun süre duraklar.
- **MQRPUT** başarısızlıkları; örneğin, **MQR_PUBLICAION_FAILURE**.
- **MQR_UNKNOWN_REMOTE_Q_MGR** nedeniyle teslim edilmeyen iletiler kuyruğuna yerleştirilen yayınlar

Bu nedenlerden ötürü, **REFRESH CLUSTER** komutu verilmeden önce yayınlama/abone olma uygulamalarının susturulması gerekir.

Yayınlama/abone olma kümesindeki bir kuyruk yöneticisinde **REFRESH CLUSTER** komutu verildikten sonra, tüm küme kuyruğu yöneticileri ve küme konuları başarıyla yenileninceye kadar bekleyin ve [Yetkili sunucu aboneliklerinin yeniden eşzamanlanması](#) başlıklı konuda açıklandığı gibi yetkili sunucu

aboneliklerini yeniden eşitleyin. Tüm yetkili sunucu abonelikleri doğru şekilde yeniden eşitlendiğinde, yayınlama/abone olma uygulamalarınızı yeniden başlatın.

Bir **REFRESH CLUSTER** komutunun tamamlanması uzun sürerse, **SYSTEM . CLUSTER . COMMAND . QUEUECURDEPTH (CURDEPTH)** seçeneğine bakarak komutu izleyin.

İlgili kavramlar

“Kümeleme: REFRESH CLUSTER en iyi uygulamalarını kullanma” sayfa 67

Bir kümeyle ilgili yerel olarak tutulan tüm bilgileri atmak ve kümedeki tüm havuzlardan bu bilgileri yeniden oluşturmak için **REFRESH CLUSTER** komutunu kullanabilirsiniz. Kural dışı durumlar dışında bu komutu kullanmanıza gerek yoktur. Bunu kullanmanız gerekiyorsa, nasıl kullanacağınıza ilişkin özel konular vardır. Bu bilgiler, müşterilerden gelen testlere ve geribildirime dayalı bir kılavuздur.

İlgili başvurular

[REFRESH CLUSTER çalıştırılırken uygulama sorunları görüldü](#)

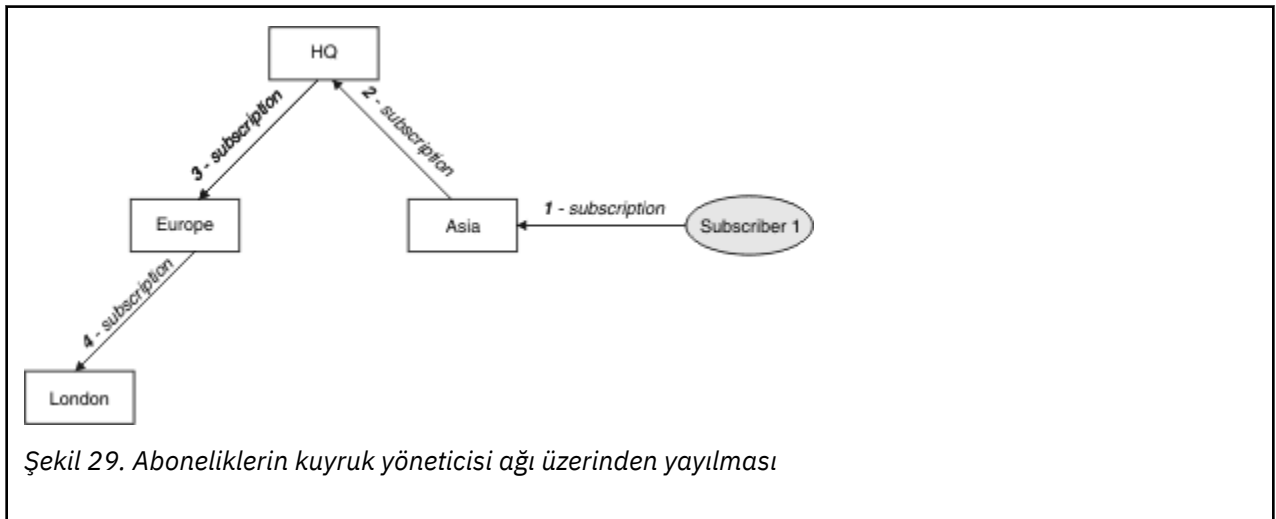
[MQSC Komutları başvurusu: REFRESH CLUSTER](#)

Yayınlama/abone olma sıradüzenlerinde yönlendirme

Dağıtılmış kuyruk yöneticisi topolojiniz bir yayınlama/abone olma sıradüzeniye ve bir kuyruk yöneticisinde abonelik yapılırsa, varsayılan olarak sıradüzendeki her kuyruk yöneticisinde bir yetkili sunucu aboneliği yaratılır. Herhangi bir kuyruk yöneticisinde alınan yayınlar, hiyerarşi aracılığıyla eşleşen bir aboneliği barındıran her kuyruk yöneticisine yönlendirilir.

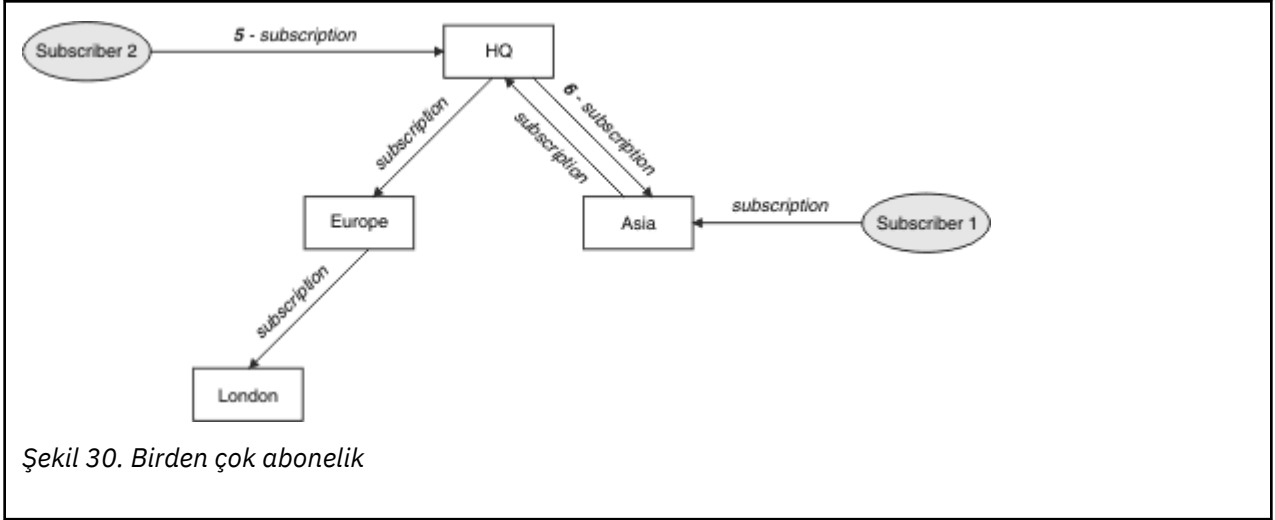
Yayınlama/abone olma sıradüzenlerindeki ve kümelerdeki kuyruk yöneticileri arasında iletilerin nasıl yönlendirildiğine ilişkin bir giriş için bkz. [Dağıtılmış yayınlama/abone olma ağları](#).

Dağıtılmış bir yayınlama/abone olma sıradüzenindeki bir kuyruk yöneticisinde bir konuya abonelik yapıldığında, kuyruk yöneticisi, aboneliğin bağlı kuyruk yöneticilerine dağıtıldığı işlemi yönetir. *Yetkili sunucu abonelikleri*, ağdaki tüm kuyruk yöneticilerine akar. Yetkili sunucu aboneliği, bir kuyruk yöneticisine bir yayını, o konuya ilişkin abonelikleri barındıran kuyruk yöneticilerine iletmek için gereken bilgileri verir. Bir yayınlama/abone olma sıradüzenindeki her kuyruk yöneticisi yalnızca doğrudan ilişkilerini bilir. Bir kuyruk yöneticisine yerleştirilen yayınlar, doğrudan ilişkileri aracılığıyla abonelikleri olan kuyruk yöneticilerine gönderilir. Bu, *Abone 1* 'in *Asya* kuyruk yöneticisinde (1) belirli bir konuya ilişkin aboneliği kaydettirdiği aşağıdaki şekilde gösterilmektedir. *Asya* kuyruk yöneticisindeki bu aboneliğe ilişkin yetkili sunucu abonelikleri, ağdaki diğer tüm kuyruk yöneticilerine iletilir (2.3, 4).

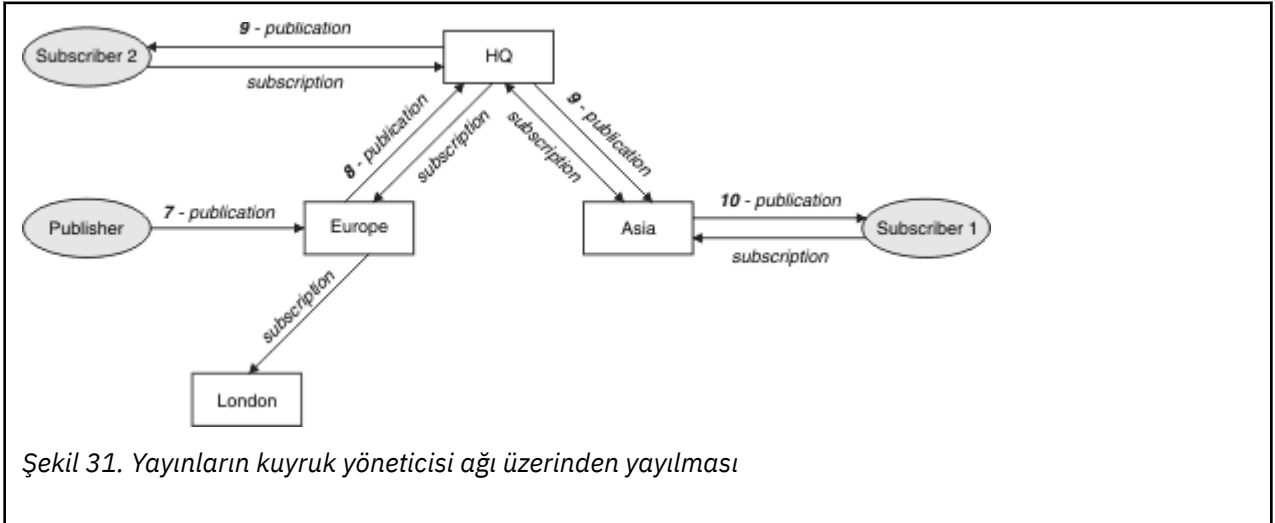


Şekil 29. Aboneliklerin kuyruk yöneticisi ağı üzerinden yayılması

Bir kuyruk yöneticisi, yerel uygulamalardan ya da uzak kuyruk yöneticilerinden yaratılan tüm abonelikleri birleştirir. Bir yetkili abonelik yoksa, komşularıyla abonelik konuları için yetkili sunucu abonelikleri oluşturur. Bu, *Abone 2* 'nin *HQ* kuyruk yöneticisinde (5) [Şekil 29 sayfa 103](#) ile aynı konuya abonelik kaydettirdiği aşağıdaki şekilde gösterilmektedir. Bu konuya ilişkin abonelik *Asya* kuyruk yöneticisine iletilir; böylece, aboneliklerin ağda (6) başka bir yerde var olduğunu fark eder. Bu konuya ilişkin bir abonelik zaten kayıtlı olduğundan, abonelik *Avrupa* kuyruk yöneticisine iletilmez; [Şekil 29 sayfa 103](#) içindeki 3. adımı bakın.



Bir uygulama bilgileri bir konuya yayınladığında, alan kuyruk yöneticisi varsayılan olarak bu bilgiyi, konuya ilişkin geçerli abonelikleri olan tüm kuyruk yöneticilerine iletir. Bir ya da daha çok ara kuyruk yöneticisi aracılığıyla iletilebilir. Bu, bir yayıncının Şekil 30 sayfa 104 ile aynı konuda *Avrupa* kuyruk yöneticisine (7) bir yayın gönderdiği aşağıdaki şekilde gösterilmektedir. Bu konu için HQ'dan Europe'a bir abonelik olduğundan, yayın HQ kuyruk yöneticisine (8) iletir. Ancak, Londra'dan Avrupa'ya (yalnızca Avrupa'dan Londra'ya) abonelik olmadığından, yayın Londra kuyruk yöneticisine iletmez. HQ kuyruk yöneticisi, yayını doğrudan Abone 2'ye ve Asya kuyruk yöneticisine (9) gönderir. Yayın, Asya (10) ülkelerinden Abone 1'e iletir.



Bir kuyruk yöneticisi başka bir kuyruk yöneticisine herhangi bir yayın ya da abonelik gönderdiğinde, iletide kendi kullanıcı kimliğini ayarlar. Bir yayınlama/abone olma sıradüzeni kullanıyorsanız ve gelen kanal, kullanıcı kimliğinin yetkisine sahip iletileri iletiye koyacak şekilde ayarlandıysa, gönderen kuyruk yöneticisinin kullanıcı kimliğini yetkilendirmeniz gerekir. Bkz. [Kuyruk yöneticisi sıradüzeniyle varsayılan kullanıcı kimliklerinin kullanılması](#).

Not: Bunun yerine yayınlama/abone olma kümelerini kullanırsanız, yetkilendirme küme tarafından işlenir.

Özet ve diğer önemli noktalar

Bir yayınlama/abone olma sıradüzeni, kuyruk yöneticileri arasındaki ilişki üzerinde size kesin denetim sağlar. Yaratıldıktan sonra, yönetmek için çok az manuel müdahaleye ihtiyaç duymaktadır. Ancak, sisteminize bazı kısıtlamalar da getirir:

- Sıradüzendeki yüksek düğümler, özellikle de kök düğüm, sağlam, yüksek kullanılabilirlikli ve yüksek performanslı ekipmanlarda barındırılmalıdır. Bunun nedeni, bu düğümler arasında daha fazla yayın trafiğinin akması beklenmektedir.
- Sıradüzendeki her yaprak olmayan kuyruk yöneticisinin kullanılabilirliği, ağır yayıncılardan diğer kuyruk yöneticilerindeki abonelere ileti akışı yeteneğini etkiler.
- Varsayılan olarak, abone olunan tüm konu dizgileri sıradüzeni boyunca yayılır ve yayınlar yalnızca ilişkili konuya aboneliği olan uzak kuyruk yöneticilerine yayılır. Bu nedenle abonelik kümesinde yapılan hızlı değişiklikler sınırlayıcı bir faktör olabilir. Bu varsayılan davranışı değiştirebilir ve bunun yerine tüm yayınları tüm kuyruk yöneticilerine yayarak yetkili sunucu abonelikleri gereksinimini ortadan kaldıracaktır. Bkz. [Yayınlama/abone olma ağlarında abonelik performansı](#).

Not: Benzer bir kısıtlama doğrudan yönlendirilen kümeler için de geçerlidir.

- Yayınlama/abone olma kuyruk yöneticilerinin birbiriyle bağlantılı olması nedeniyle, yetkili sunucu aboneliklerinin ağdaki tüm düğümlere yayılması zaman alır. Uzak yayınların hemen abone olmaya başlaması gerekmez; bu nedenle, yeni bir konu dizisi aboneliğinden sonra erken yayınlar gönderilmeyebilir. Tüm yayınları tüm kuyruk yöneticilerine yayarak abonelik gecikmesinin neden olduğu sorunları kaldırabilirsiniz; bu, yetkili sunucu abonelikleri gereksinimini ortadan kaldırır. Bkz. [Yayınlama/abone olma ağlarında abonelik performansı](#).

Not: Bu kısıtlama, doğrudan yönlendirilen kümeler için de geçerlidir.

- Bir yayınlama/abone olma sıradüzeni için, kuyruk yöneticilerinin eklenmesi ya da kaldırılması, bu kuyruk yöneticilerinin konumunu ve diğer kuyruk yöneticilerine güvenmelerini dikkatli bir şekilde göz önünde bulundurarak, sıradüzenine el ile yapılandırma yapılmasını gerektirir. Sıradüzeninin alt kısmındaki kuyruk yöneticilerini ekliyor ya da kaldırmıyorsanız ve bu nedenle bunların altında başka dal yoksa, sıradüzendeki diğer kuyruk yöneticilerini de yapılandırmanız gerekir.

Yöneltilme mekanizmanız olarak bir yayınlama/abone olma sıradüzenini kullanmadan önce, [“Yayınlama/abone olma kümelerinde doğrudan yönlendirme” sayfa 75](#) ve [“Konu Yayınlama/abone olma kümelerinde anasistem yönlendirmesi” sayfa 79](#) içinde ayrıntılı olarak açıklanan alternatif yaklaşımları keşfedin.

Dağıtılmış yayınlama/abone olma sistem kuyrukları

Yayınlama/abone olma ileti sistemi için kuyruk yöneticileri tarafından dört sistem kuyruğu kullanılır. Yalnızca sorun belirleme ve kapasite planlama amaçlarıyla varlıklarını bilmelisiniz.

Bu kuyrukların nasıl izleneceğine ilişkin yönergeler için [Yayınlama/abone olma ağlarında üreticilerin ve tüketicilerin dengelenmesi](#) başlıklı konuya bakın.

<i>Çizelge 6. Çoklu platformlarda sistem kuyruklarını yayınla/abone ol</i>	
Sistem kuyruğu	Amaç
SYSTEM.INTER.QMGR.CONTROL	IBM MQ dağıtılmış yayınlama/abone olma denetim kuyruğu
SYSTEM.INTER.QMGR.FANREQ	IBM MQ dağıtılmış yayınlama/abone olma iç yetkili sunucu aboneliği çıkış işlemi giriş kuyruğu
SYSTEM.INTER.QMGR.PUBS	IBM MQ dağıtılmış yayınlama/abone olma yayınları
SYSTEM.HIERARCHY.STATE	IBM MQ dağıtılmış yayınlama/abone olma sıradüzeni ilişkisi durumu

z/OS z/OS sistemlerinde, kuyruk yöneticisini yaratırken gereken sistem nesnelerini, CSQINP2 kullanıma hazırlama giriş verileri kümesine CSQ4INSX, CSQ4INSR ve CSQ4INSG örneklerini ekleyerek ayarlarsanız. Daha fazla bilgi için bkz. [Görev 13: Kullanıma hazırlama giriş veri kümelerinin uyarlanması](#).

Yayınlama/abone olma sistem kuyruklarının öznitelikleri [Çizelge 7 sayfa 106](#) içinde gösterilir.

Çizelge 7. Yayınlama/abone olma sistem kuyruklarının öz nitelikleri	
Öznitelik	Varsayılan değer
DEFPSIST	Evet
DEFSOPT	PAYLAŞILAN
MAXMSGL	<p>Multi Çoklu platformlar üzerinde: ALTER QMGR komutunun MAXMSGL parametresinin değeri</p> <p>z/OS z/OS üzerinde: 4194304 (yani, 4 MB)</p>
DERINLIKS	999999999
PAYLAŞ	Yok
<p>z/OS</p> <p>z/OS</p> <p>STGCLASS</p>	Bu öznitelik yalnızca z/OS platformlarında kullanılır

Not: Uygulamalar tarafından konan iletileri içeren tek kuyruk SYSTEM . INTER . QMGR . PUBS. **MAXDEPTH** , kesintiler ya da aşırı yükleme süreleri sırasında yayınlanan iletilerin geçici olarak oluşturulmasına izin vermek için bu kuyruk için maksimum değerine ayarlanır. Kuyruk yöneticisi, o kuyruk derinliğinin bulunmadığı bir sistemde çalışıyorsa, bu ayarlanmalıdır.

İlgili görevler

[Dağıtılmış yayınlama/abone olma sorunlarının giderilmesi](#)

Dağıtılmış yayınlama/abone olma sistem kuyruğu hataları

Dağıtılmış yayınlama/abone olma kuyruk yöneticisi kuyrukları kullanılmadığında hatalar oluşabilir. Bu, abonelik bilgisinin yayınlama/abone olma ağı çapında yayılmasını ve uzak kuyruk yöneticilerindeki aboneliklere yayınlanmasını etkiler.

SYSTEM . INTER . QMGR . FANREQ dışarı iletme isteği kuyruğu kullanılmıyorsa, abonelik oluşturulması bir hata oluşturabilir ve yetkili sunucu aboneliklerinin doğrudan bağlı kuyruk yöneticilerine teslim edilmesi gerekirken kuyruk yöneticisi hata günlüğüne hata iletileri yazılır.

Sıradüzen ilişkisi durum kuyruğu SYSTEM . HIERARCHY . STATE kullanılmıyorsa, kuyruk yöneticisi hata günlüğüne bir hata iletileri yazılır ve yayınlama/abone olma altyapısı COMPAT kipine girer. Yayınlama/abone olma kipini görüntülemek için DISPLAY QMGR PSMODE komutunu kullanın.

SYSTEM . INTER . QMGR kuyruklarından herhangi biri kullanılmıyorsa, kuyruk yöneticisi hata günlüğüne bir hata iletileri yazılır ve işlev devre dışı bırakılmasa da, yayınlama/abone olma iletileri bu ya da uzak kuyruk yöneticilerindeki kuyruklarda oluşturulabilir.

Yayınlama/abone olma sistem kuyruğu ya da bir üst, alt öge ya da yayınlama/abone olma küme kuyruk yöneticisi için gerekli iletiler kuyruğu kullanılmıyorsa, aşağıdaki sonuçlar ortaya çıkar:

- Yayınlar teslim edilmez ve yayınlama uygulaması bir hata alabilir. Yayınlama uygulamasının bir hata almasıyla ilgili ayrıntılar için **DEFINE TOPIC** komutunun şu değiştiricilerine bakın: **PMSGDLV** , **NPMSGDLV** ve **USEDLQ** .
- Alınan kuyruklar arası yönetici yayınları giriş kuyruğuna geriletildi ve daha sonra yeniden denendi. Geriletme eşliğine ulaşırsa, teslim edilmeyen yayınlar gitmeyen iletiler kuyruğuna yerleştirilir. Kuyruk yöneticisi hata günlüğü, sorunun ayrıntılarını içerir.
- Teslim edilmemiş bir yetkili sunucu aboneliği, fanout istek kuyruğuna geriletir ve daha sonra yeniden denir. Geriletme eşliğine ulaşırsa, teslim edilmeyen yetkili sunucu aboneliği bağlı herhangi bir kuyruk yöneticisine teslim edilmez ve gitmeyen ileti kuyruğuna yerleştirilir. Kuyruk yöneticisi hata günlüğü, gerekli düzeltme işlemi ayrıntıları da içinde olmak üzere, sorunun ayrıntılarını içerir.

- Sıradüzen ilişkisi iletişim kuralı iletileri başarısız olur ve bağlantı durumu ERROR olarak işaretlenir. Bağlantı durumunu görüntülemek için **DISPLAY PUBSUB** komutunu kullanın.

İlgili görevler

[Dağıtılmış yayınlama/abone olma sorunlarının giderilmesi](#)

Multi Çoklu platformlarda depolama ve performans gereksinimlerinin planlanması

IBM MQ sisteminiz için gerçekçi ve ulaşılabilir depolama ve performans hedefleri belirlemelisiniz. Platformunuzda depolamayı ve performansı etkileyen etkenler hakkında bilgi almak için bağlantıları kullanın.

Gereksinimler, IBM MQ işletim sistemini kullanmakta olduğunuz sistemlere ve hangi bileşenleri kullanmak istediğinize bağlı olarak değişir.

Desteklenen donanım ve yazılım ortamlarına ilişkin en son bilgiler için bkz. [IBM MQ için Sistem Gereksinimleri](#).

IBM MQ , kuyruk yöneticisi verilerini dosya sisteminde saklar. IBM MQ ile kullanılmak üzere dizin yapılarının planlanmasına ve yapılandırılmasına ilişkin bilgi edinmek için aşağıdaki bağlantıları kullanın:

- [“Multiplatforms üzerinde dosya sistemi desteğinin planlanması” sayfa 111](#)
- [“Çoklu platformlarda paylaşılan dosya sistemlerine ilişkin gereksinimler” sayfa 112](#)
- [“IBM MQ dosyalarını Multiplatforms üzerinde paylaşma” sayfa 121](#)
- [Linux AIX “AIX and Linux sistemlerinde dizin yapısı” sayfa 123](#)
- [Windows “Windows sistemlerinde dizin yapısı” sayfa 132](#)
- [IBM i “IBM i üzerinde dizin yapısı” sayfa 135](#)

AIX and Linux üzerinde sistem kaynakları, paylaşılan bellek ve işlem önceliği hakkında bilgi için aşağıdaki bağlantıları kullanın:

- [Linux AIX “IBM MQ ve UNIX System V IPC kaynakları” sayfa 140](#)
- [AIX “AIX üzerinde paylaşılan bellek” sayfa 139](#)
- [Linux AIX “IBM MQ ve UNIX Süreç Önceliği” sayfa 140](#)

Günlük dosyalarıyla ilgili bilgi için aşağıdaki bağlantıları kullanın:

- [“Multiplatforms üzerinde dairesel ya da doğrusal günlük kaydının seçilmesi” sayfa 139](#)
- [Günlüğün Boyutunun Hesaplanması](#)

İlgili kavramlar

[“z/OS üzerinde IBM MQ ortamının planlanması” sayfa 140](#)

IBM MQ ortamınızı planlarken, veri kümeleri, sayfa kümeleri, Db2, Coupling Facilities ve günlüğe kaydetme ve yedekleme olanaklarına ilişkin kaynak gereksinimlerini göz önünde bulundurmanız gerekir. IBM MQ ' in çalıştığı ortamı planlamak için bu konuyu kullanın.

İlgili görevler

[“IBM MQ mimarisinin planlanması” sayfa 5](#)

IBM MQ ortamınızı planlarken, IBM MQ ' un tek ve birden çok kuyruk yöneticisi mimarisi ve noktadan noktaya iletişim ve yayınlama/abone olma ileti sistemi stilleri için sağladığı desteği göz önünde bulundurun. Ayrıca, kaynak gereksinimlerinizi ve günlük kaydı ve yedekleme olanaklarını kullanımınızı da planlayın.

İlgili başvurular

[AIX and Linux üzerinde donanım ve yazılım gereksinimleri](#)

[Windows üzerinde donanım ve yazılım gereksinimleri](#)

Çoklu platformlarda disk alanı gereksinimleri

IBM MQ için depolama gereksinimleri, kurduğunuz bileşenlere ve ne kadar çalışma alanına gereksinim duyacağınıza bağlıdır.

Disk depolama alanı, gereksinim duydukları önkoşul bileşenleri de içinde olmak üzere, kurmayı seçtiğiniz isteğe bağlı bileşenler için gereklidir. Toplam depolama gereksinimi, kullandığınız kuyruk sayısına, kuyruklardaki iletilerin sayısına ve boyutuna ve iletilerin kalıcı olup olmadığına da bağlıdır. Ayrıca, disk, manyetik bant ya da diğer ortamlarda arşivleme kapasitesine ve kendi uygulama programlarınıza ilişkin alana da gereksinim duyarsınız.

Aşağıdaki çizelgede, ürünün çeşitli birleşimlerini farklı altyapılara kurduğunuzda gereken yaklaşık disk yeri gösterilir. (Değerler, MB 'nin 1.048.576 bayt olduğu en yakın 5 MB' ye yuvarlanır.)

- ▶ **LTS** “Long Term Support için disk alanı gereksinimleri” sayfa 108
- ▶ **CD** “Continuous Delivery için disk alanı gereksinimleri” sayfa 109

Long Term Support için disk alanı gereksinimleri

V 9.3.0 LTS

Çizelge 8. IBM MQ Çoklu Platformlar için disk alanı gereksinimleri Long Term Support			
Hizmet olarak sunulan	İstemci kurulumu “1” sayfa 108	Sunucu Kurulumu “2” sayfa 108	Tam kurulum “3” sayfa 108
▶ AIX AIX	335 MB	375 MB	1810 MB
▶ IBM i IBM i (bkz. 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 (64 bit kurulum) “4” sayfa 108	295 MB	415 MB	2310 MB

Notlar:

- İstemci kurulumu aşağıdaki bileşenleri içerir:
 - Çalıştırma zamanı
 - Müşteri
- Sunucu kurulumu aşağıdaki bileşenleri içerir:
 - Çalıştırma zamanı
 - Sunucu
- Tam kurulum, kullanılabilir tüm bileşenleri içerir.
- ▶ **Windows** Burada listelenen bileşenlerin tümü Windows sistemlerinde kurulabilir özellikler değildir; işlevleri bazen diğer özelliklerde de bulunur. Bkz. [Windows sistemleri için IBM MQ özellikleri](#).

IBM i için ek notlar: **IBM i**

1. IBM i ' da yerel istemciyi sunucudan ayıramazsınız. Çizelgedeki sunucu şekli, İngilizce Dil Yüğü (2924) ile birlikte Javaolmadan 5724H72*BASE için geçerlidir. 22 olası benzersiz dil yüğü vardır.
2. Çizelgedeki şekil, yerel istemci 5725A49 *BASE için kullanılır (Javaolmadan).
3. Java ve JMS sınıfları hem sunucu hem de istemci bağ tanımlarına eklenebilir. Bu özellikleri eklemek istiyorsanız 110 MB ekleyin.
4. İstemciye ya da sunucuya örnek kaynağı eklendiğinde ek 10 MB eklenir.
5. Java ve JMS sınıflarına örnek eklenmesi fazladan 5 MB ekler.

Continuous Delivery için disk alanı gereksinimleri**V 9.3.0** **CD**

<i>Çizelge 9. IBM MQ Çoklu Platformlar için disk alanı gereksinimleri Continuous Delivery</i>			
Platform/CD yayın düzeyi	İstemci kuruluşu "1" sayfa 111	Sunucu Kuruluşu "2" sayfa 111	Tam kuruluş "3" sayfa 111
AIX AIX			
V 9.3.0 IBM MQ 9.3.0	330 MB	375 MB	1760 MB
V 9.3.1 IBM MQ 9.3.1	340 MB	375 MB	1815 MB
V 9.3.2 IBM MQ 9.3.2	355 MB	390 MB	1440 MB
V 9.3.3 IBM MQ 9.3.3	355 MB	390 MB	1440 MB
V 9.3.4 IBM MQ 9.3.4	355 MB	390 MB	1440 MB
V 9.3.5 IBM MQ 9.3.5	355 MB	390 MB	1440 MB
Linux Linux for x86-64 (64 bit)			
V 9.3.0 IBM MQ 9.3.0	265 MB	295 MB	2010 MB
V 9.3.1 IBM MQ 9.3.1	275 MB	295 MB	2010 MB
V 9.3.2 IBM MQ 9.3.2	280 MB	295 MB	1195 MB
V 9.3.3 IBM MQ 9.3.3	280 MB	295 MB	1195 MB
V 9.3.4 IBM MQ 9.3.4	280 MB	295 MB	1195 MB

Çizelge 9. IBM MQ Çoklu Platformlar için disk alanı gereksinimleri Continuous Delivery (devamı var)

Platform/CD yayın düzeyi	İstemci kurulumu "1" sayfa 111	Sunucu Kurulumu "2" sayfa 111	Tam kurulum "3" sayfa 111
> V 9.3.5 IBM MQ 9.3.5	280 MB	295 MB	1195 MB
Linux Linux on Power Systems - Little Endian			
> V 9.3.0 IBM MQ 9.3.0	170 MB	190 MB	1350 MB
> V 9.3.1 IBM MQ 9.3.1	170 MB	195 MB	1400 MB
> V 9.3.2 IBM MQ 9.3.2	170 MB	195 MB	1075 MB
> V 9.3.3 IBM MQ 9.3.3	170 MB	195 MB	1075 MB
> V 9.3.4 IBM MQ 9.3.4	170 MB	195 MB	1075 MB
> V 9.3.5 IBM MQ 9.3.5	170 MB	195 MB	1075 MB
Linux Linux for IBM Z			
> V 9.3.0 IBM MQ 9.3.0	255 MB	290 MB	1435 MB
> V 9.3.1 IBM MQ 9.3.1	255 MB	290 MB	1485 MB
> V 9.3.2 IBM MQ 9.3.2	260 MB	290 MB	1160 MB
> V 9.3.3 IBM MQ 9.3.3	260 MB	290 MB	1160 MB
> V 9.3.4 IBM MQ 9.3.4	260 MB	290 MB	1160 MB
> V 9.3.5 IBM MQ 9.3.5	260 MB	290 MB	1160 MB
Windows Windows (64 bit kurulum) "4" sayfa 111			
> V 9.3.0 IBM MQ 9.3.0	290 MB	410 MB	2095 MB
> V 9.3.1 IBM MQ 9.3.1	295 MB	415 MB	2310 MB

Çizelge 9. IBM MQ Çoklu Platformlar için disk alanı gereksinimleri Continuous Delivery (devamı var)

Platform/CD yayın düzeyi	İstemci kurulumu "1" sayfa 111	Sunucu Kurulumu "2" sayfa 111	Tam kurulum "3" sayfa 111
V 9.3.2 IBM MQ 9.3.2	300 MB	415 MB	1785 MB
V 9.3.3 IBM MQ 9.3.3	300 MB	415 MB	1785 MB
V 9.3.4 IBM MQ 9.3.4	300 MB	415 MB	1785 MB
V 9.3.5 IBM MQ 9.3.5	300 MB	415 MB	1785 MB

Notlar:

- İstemci kurulumu aşağıdaki bileşenleri içerir:
 - Çalıştırma zamanı
 - Müşteri
- Sunucu kurulumu aşağıdaki bileşenleri içerir:
 - Çalıştırma zamanı
 - Sunucu
- Tam kurulum, kullanılabilir tüm bileşenleri içerir.
- Windows** Burada listelenen bileşenlerin tümü Windows sistemlerinde kurulabilir özellikler değildir; işlevleri bazen diğer özelliklerde de bulunur. Bkz. [Windows sistemleri için IBM MQ özellikleri](#).

İlgili kavramlar

[IBM MQ bileşenleri ve özellikleri](#)

Multi

Multiplatforms üzerinde dosya sistemi desteğinin planlanması

Kuyruk yöneticisi verileri dosya sisteminde saklanır. Bir kuyruk yöneticisi, birden çok yönetim ortamı kuyruk yöneticisinin aynı anda etkin olmasını önlemek için dosya sistemi kilitlemesini kullanır.

Paylaşılan dosya sistemleri

Paylaşılan dosya sistemleri, birden çok sistemin aynı fiziksel depolama aygıtına eşzamanlı olarak erişmesini sağlar. Kilitleme ve eşzamanlılık denetimini zorunlu hale getirmeden birden çok sistem aynı fiziksel depolama aygıtına doğrudan erişirse bozulmalar meydana gelir. İşletim sistemleri, yerel işlemler için kilitleme ve eşzamanlılık denetimi ile yerel dosya sistemleri sağlar; ağ dosya sistemleri, dağıtılmış sistemler için kilitleme ve eşzamanlılık denetimi sağlar.

Geçmişte, ağ bağlantılı dosya sistemleri iletileri günlüğe kaydetme gereksinimlerini karşılamak için yeterince hızlı performans göstermemiş veya yeterli kilitleme ve eşzamanlılık denetimi sağlamıştır. Bugün, ağ bağlantılı dosya sistemleri, *RFC 3530, Network File System (NFS) sürüm 4 iletişim kuralı* gibi güvenilir ağ dosya sistemi iletişim kurallarının uygulamalarını ve iyi performans sağlayabilir ve iletileri güvenilir bir şekilde günlüğe kaydetme gereksinimlerini karşılar.

Paylaşılan dosya sistemleri ve IBM MQ

Çok eşgönlümlü bir kuyruk yöneticisine ilişkin kuyruk yöneticisi verileri paylaşılan bir ağ dosya sisteminde saklanır. AIX, Linux, and Windows sistemlerinde, kuyruk yöneticisinin veri dosyaları ve

günlük dosyaları paylaşılan ağ dosya sistemine yerleştirilmelidir. **IBM i** IBM üzerinde, günlük dosyaları yerine günlükler kullanılır ve günlükler paylaşılabilir. IBM i üzerindeki çok eşgörsümlü kuyruk yöneticileri, günlükleri farklı kuyruk yöneticisi eşgörsümleri arasında kullanılabilir kılmak için günlük eşlemeyi ya da değişimli kullanılabilir günlükleri kullanır.

IBM MQ , aynı anda aynı çok eşgörsümlü kuyruk yöneticisinin birden çok eşgörsümlünün etkin olmasını önlemek için kilitlemeyi kullanır. Aynı kilitleme, iki ayrı kuyruk yöneticisinin aynı kuyruk yöneticisi veri dosyası kümesini istemeden kullanamamasını da sağlar. Bir kuyruk yöneticisinin aynı anda yalnızca bir yönetim ortamının kilidi olabilir. Sonuç olarak IBM MQ , paylaşılan dosya sistemi olarak erişilen ağ üzerinde saklanan kuyruk yöneticisi verilerini destekler.

Ağ dosya sistemlerinin tüm kilitleme iletişim kuralları sağlam olmadığından ve bir dosya sistemi veri bütünlüğü yerine performans için yapılandırılabilirliğinden, bir ağ dosya sisteminin kuyruk yöneticisi verilerine ve günlüklerine erişimi doğru şekilde denetleyip denetlemeyeceğini sınamak için **amqmfsc** komutunu çalıştırmanız gerekir. Bu komut yalnızca UNIX, Linux ve IBM i sistemleri için geçerlidir. Windows işletim sisteminde yalnızca bir desteklenen ağ dosya sistemi vardır ve **amqmfsc** komutu gerekli değildir.

İlgili görevler

“Çoklu platformlarda paylaşılan dosya sistemi davranışının doğrulanması” sayfa 114

AIX, Linux ya da IBM i üzerindeki bir paylaşılan dosya sisteminin çok eşgörsümlü bir kuyruk yöneticisinin kuyruk yöneticisi verilerini saklamaya ilişkin gereksinimleri karşılayıp karşılamadığını denetlemek için **amqmfsc** komutunu çalıştırın. (Bir Windows yapılandırmasının tek gereksinimi, paylaşılan depolama için KOBİ 3 kullanmasıdır.)

Multi Çoklu platformlarda paylaşılan dosya sistemlerine ilişkin gereksinimler

Paylaşılan dosya sistemleri, IBM MQ ile güvenilir bir şekilde çalışabilmek için veri yazma bütünlüğü, dosyalara özel erişim ve serbest bırakma kilitleri sağlamalıdır.

Paylaşılan bir dosya sisteminin karşılaması gereken gereksinimler

Paylaşılan bir dosya sisteminin IBM MQ ile güvenilir bir şekilde çalışması için yerine getirmesi gereken üç temel gereksinim vardır:

1. Veri yazma bütünlüğü

Veri yazma bütünlüğü bazen *Diske veri iletiminde yazolarak* adlandırılır. Kuyruk yöneticisi, fiziksel aygıtta başarıyla kesinleştirilen verilerle eşzamanlanabilmelidir. Bir işlemsel sistemde, diğer işlemeye devam etmeden önce bazı yazma işlemlerinin güvenli bir şekilde kesinleştirildiğinden emin olun.

Daha spesifik olarak, IBM MQ for AIX or Linux platformları *O_SYNC* açma seçeneğini ve *fsync()* sistem çağrısını kurtarılabilir ortama yazma işlemini açıkça zorlamak için kullanır ve yazma işlemi bu seçeneklere bağlıdır.



Uyarı: **Linux** Dosya sistemini, zamanuyumlu yazma seçeneklerini desteklemeye devam eden ve eşitleme seçeneğinden daha iyi performans sağlayan *async* seçeneğiyle bağlamanız gerekir.

Ancak, dosya sistemi Linux' den dışı aktarıldıysa, dosya sistemini eşitleme seçeneğini kullanarak yine de dışı aktarmanız gerektiğini unutmayın.

2. Dosyalara dışlayıcı erişim garantisi

Birden çok kuyruk yöneticisini uyumlulaştırmak için, bir kuyruk yöneticisinin bir dosyada özel bir kilit elde etmesi için bir mekanizma olması gerekir.

3. Hata durumunda kilitleri serbest bırak

Bir kuyruk yöneticisi başarısız olursa ya da dosya sistemiyle iletişim başarısız olursa, kuyruk yöneticisi tarafından kilitlenen dosyaların kilidi açılmalı ve kuyruk yöneticisinin dosya sistemine yeniden bağlanmasını beklemeden diğer işlemler tarafından kullanılabilir kılınmalıdır.

Paylaşılan bir dosya sisteminin IBM MQ ' in güvenilir bir şekilde çalışması için bu gereksinimleri karşılaması gerekir. Yoksa, çok eşgörünümlü bir kuyruk yöneticisi yapılanışında paylaşılan kütük sistemi kullanılırken kuyruk yöneticisi verileri ve günlükleri bozulur.

Microsoft Windows üzerindeki çok eşgörünümlü kuyruk yöneticileri için, ağa bağlı depolamaya Microsoft Windows ağları tarafından kullanılan Sunucu İleti Bloğu (SMB) iletişim kuralı tarafından erişilmelidir. Server Message Block (SMB) istemcisi, Microsoft Windows dışındaki platformlarda anlambilimi kilitlemek için IBM MQ gereksinimlerini karşılamıyor, bu nedenle Microsoft Windows dışındaki platformlarda çalışan çok eşgörünümlü kuyruk yöneticileri paylaşılan dosya sistemi olarak Server Message Block (SMB) kullanmamalıdır.

Desteklenen diğer platformlardaki çok eşgörünümlü kuyruk yöneticileri için, depolamaya Posix uyumlu ve kira tabanlı kilitlemeyi destekleyen bir ağ dosya sistemi iletişim kuralı tarafından erişilmelidir. Ağ Dosya Sistemi 4 bu gereksinimi karşılar. Bir hatadan sonra kilitleri serbest bırakmak için güvenilir bir mekanizmaya sahip olmayan Ağ Dosya Sistemi Sürüm 3 gibi eski dosya sistemleri, çok eşgörünümlü kuyruk yöneticileriyle birlikte kullanılmamalıdır.

Paylaşılan dosya sisteminin gereksinimleri karşılayıp karşılamadığını denetler



Kullanmayı planladığınız paylaşılan dosya sisteminin bu gereksinimleri karşılayıp karşılamadığını denetlemeniz gerekir. Dosya sisteminin güvenilirlik için doğru yapılandırılıp yapılandırılmadığını da denetlemeniz gerekir. Paylaşılan dosya sistemleri bazen güvenilirlik pahasına performansı artırmak için yapılandırma seçenekleri sağlar.

Ek bilgi için [Testing statement for IBM MQ multi-instance queue manager file systems](#) başlıklı konuya bakın.

Olağan koşullarda IBM MQ , öznitelik önbelleğe alma ile doğru çalışır ve önbelleğe almayı devre dışı bırakmak gerekmez; örneğin, bir NFS bağlamasında NOAC ayarı yapılarak. Birden çok dosya sistemi istemcisi dosya sistemi sunucusunda aynı dosyaya yazma erişimi için çekişirken, her istemci tarafından kullanılan önbelleğe alınan öznitelikler sunucudaki özniteliklerle aynı olmayabileceğinden, özniteliklerin önbelleğe alınması sorunlara neden olabilir. Bu yolla erişilen dosyaların bir örneği, çok eşgörünümlü bir kuyruk yöneticisine ilişkin kuyruk yöneticisi hata günlükleridir. Kuyruk yöneticisi hata günlüklerine hem etkin hem de yedek kuyruk yöneticisi yönetim ortamı tarafından yazılabilir ve önbelleğe alınan dosya öznitelikleri, dosyaların aktarımı gerçekleşmeden önce hata günlüklerinin beklenenden daha fazla büyümesine neden olabilir.

Dosya sisteminin denetlemenize yardımcı olması için [Paylaşılan dosya sistemi davranışını doğrulamagörevini](#) çalıştırın. Bu görev, paylaşılan dosya sisteminizin 2 ve 3 gereksinimlerini karşılayıp karşılamadığını denetler. Paylaşılan dosya sistemi belgelerinizde ya da verileri diske kaydetmeyi deneyerek gereksinimi 1 doğrulamanız gerekir.

Disk hataları, IBM MQ ' ün First Failure Data Capture hataları olarak bildirdiği diske yazılırken hatalara neden olabilir. Paylaşılan dosya sisteminde disk hataları olup olmadığını denetlemek için işletim sisteminize ilişkin dosya sistemi denetleyicisini çalıştırabilirsiniz. Örneğin:

-  Linux ve AIX AIX and Linux üzerinde dosya sistemi denetleyicisi fsck olarak adlandırılır.
-  Windows Windows platformlarında dosya sistemi denetleyicisi CHKDSK ya da SCANDISK olarak adlandırılır.

NFS sunucusu güvenliği

Notlar:

- IBM MQ kuruluş dizinini tutmak için kullanılan bir bağlama noktası için **nosuid** ya da **noexec** seçeneklerini kullanamazsınız. Bunun nedeni, IBM MQ ' in setuid/setgid yürütülür programlarını içermesi ve bunların düzgün çalışması engellenmemesi gerektiğidir.
- Kuyruk yöneticisi verilerini yalnızca bir Ağ Dosya Sistemi (NFS) sunucusuna yerleştirdiğinizde, sistemi güvenli kılmak için sisteme bağlama komutuyla birlikte aşağıdaki üç seçeneği kullanabilirsiniz ve kuyruk yöneticisinin çalışmasını olumsuz etkilemeden:

noexec

Bu seçeneği kullanarak, uzak bir kullanıcının sistemde istenmeyen kodu çalıştırmasını önleyen NFS' de ikili dosyaların çalıştırılmasını engelleyebilirsiniz.

nosuid

Bu seçeneği kullanarak, uzak bir kullanıcının daha yüksek ayrıcalıklar elde etmesini engelleyen set-user-identifier ve set-group-identifier bitlerinin kullanılmasını engelleyebilirsiniz.

nodev

Bu seçeneği kullanarak, uzak bir kullanıcının bir chroot hapishanesinden çıkmasını engelleyen karakter ve özel aygıtların kullanılmasını ya da tanımlanmasını engelleyebilirsiniz.

IBM i Linux AIX Çoklu platformlarda paylaşılan dosya sistemi

davranışının doğrulanması

AIX, Linux ya da IBM i üzerindeki bir paylaşılan dosya sisteminin çok eşgörunümlü bir kuyruk yöneticisinin kuyruk yöneticisi verilerini saklamaya ilişkin gereksinimleri karşılayıp karşılamadığını denetlemek için **amqmfsc** komutunu çalıştırın. (Bir Windows yapılandırmasının tek gereksinimi, paylaşılan depolama için KOBİ 3 kullanmasıdır.)

Başlamadan önce

Ağa bağlı depolama alanına sahip bir sunucuya ve IBM MQ ' in kurulu olduğu diğer iki sunucuya gereksinim duyarsınız. Dosya sistemini yapılandırmak için yönetici (root) yetkinizin olması ve IBM MQ Sistem yöneticisi olarak **amqmfsc** komutunu çalıştırmanız gerekir.

Bu görev hakkında

“Çoklu platformlarda paylaşılan dosya sistemlerine ilişkin gereksinimler” sayfa 112 , çok eşgörunümlü kuyruk yöneticileriyle paylaşılan bir dosya sisteminin kullanılmasına ilişkin dosya sistemi gereksinimlerini açıklar. IBM MQ technote Testing statement for IBM MQ multi-instance queue manager file systems (Çok eşgörunümlü kuyruk yöneticisi dosya sistemleri için sınamaya), IBM ' in önceden test ettiği paylaşılan dosya sistemlerini listeler. Bu görevdeki yordam, listelenmeyen bir dosya sisteminin veri bütünlüğünü koruyup korumadığını değerlendirmenize yardımcı olmak için bir dosya sisteminin nasıl test edileceğini açıklar.

Çok eşgörunümlü bir kuyruk yöneticisinin yedeği devreye sokması, kuyruk yöneticisinin verilerine ya da günlük dosyalarına yazmasını engelleyen ağ sorunları da dahil olmak üzere donanım ya da yazılım hataları tarafından tetiklenebilir. Temel olarak, dosya sunucusunda başarısızlıklara neden olmak ilginizi çekmektedir. Ancak kilitlerin başarıyla serbest bırakıldığını sınamak için IBM MQ sunucularının da başarısız olmasına neden olmanız gerekir. Paylaşılan bir dosya sistemine güvenmek için, aşağıdaki hataların tümünü ve ortamınıza özgü diğer hataları test edin:

1. Disklerin eşitlenmesi de içinde olmak üzere dosya sunucusundaki işletim sistemi kapatılıyor.
2. Diskleri eşitlemeden dosya sunucusundaki işletim sistemini durduruyor.
3. Sunucuların her birinde sıfırlama düğmesine basılması.
4. Ağ kablosunu her bir sunucudan dışarı doğru çekerek.
5. Güç kablosunu her bir sunucudan dışarı doğru çekerek.
6. Her bir sunucu kapatılıyor.

Kuyruk yöneticisi verilerini ve günlüklerini paylaşmak için kullanacağınız ağ deposunda dizini oluşturun. Dizin sahibi bir IBM MQ Yöneticisi ya da başka bir deyişle, AIX and Linux üzerindeki mqm grubunun üyesi olmalıdır. Testleri çalıştıran kullanıcının IBM MQ Administrator yetkisi olmalıdır.

Dosya sistemini yapılandırmanıza yardımcı olması için [Linux üzerinde çok eşgörunümlü kuyruk yöneticisi yaratılması](#) ya da [Günlük ikizleme ve NetServer on IBM i](#) kullanılarak çok eşgörunümlü kuyruk yöneticisi yaratılması içinde bir dosya sistemini dışa aktarma ve sisteme bağlama örneğini kullanın. Farklı dosya sistemleri farklı yapılandırma adımları gerektirir. Dosya sistemi belgelerini okuyun.

Not: Bir kuyruk yöneticisinin bir hata sırasında ileti bütünlüğünü korumasını göstermek için IBM MQ MQI client örnek program **amqsfhac** ' ı **amqmfsc** ile paralel olarak çalıştırın.

Yordam

Denetimlerin her birinde, dosya sistemi denetleyicisi çalışırken önceki listedeki tüm hatalara neden olun. **amqsfhac** komutunu **amqmfscck** ile aynı anda çalıştırmak istiyorsanız, görevi “İleti bütünlüğünü sınamak için amqsfhac çalıştırılıyor” sayfa 119 bu görevle paralel olarak gerçekleştirin.

1. Dışa aktarılan dizini iki IBM MQ sunucusuna bağlayın.

Dosya sistemi sunucusunda, çok eşgörunümlü kuyruk yöneticilerine ilişkin verileri saklamak için bir paylaşılan dizin `shared` ve bir alt dizin (`qmdata`) yaratın. Linux üzerinde çok eşgörunümlü kuyruk yöneticileri için paylaşılan dizin ayarlama örneği için [Linux üzerinde çok eşgörunümlü kuyruk yöneticisi yaratılması](#) başlıklı konuya bakın.

2. Temel dosya sistemi davranışını denetleyin.

Tek bir IBM MQ sunucusunda, parametresiz dosya sistemi denetleyicisini çalıştırın.

IBM MQ sunucusu 1 'de:

```
amqmfscck /shared/qmdata
```

3. Her iki IBM MQ sunucusundan aynı dizine aynı anda yazmayı denetleyin.

Her iki IBM MQ sunucusunda, dosya sistemi denetleyicisini `-c` seçeneğiyle aynı anda çalıştırın.

IBM MQ sunucusu 1 'de:

```
amqmfscck -c /shared/qmdata
```

IBM MQ Server 2 'de:

```
amqmfscck -c /shared/qmdata
```

4. Her iki IBM MQ sunucusunda da bekleme ve serbest bırakma kilitlerini denetleyin.

Her iki IBM MQ sunucusunda da dosya sistemi denetleyicisini `-w` seçeneğiyle aynı anda çalıştırın.

IBM MQ sunucusu 1 'de:

```
amqmfscck -w /shared/qmdata
```

IBM MQ Server 2 'de:

```
amqmfscck -w /shared/qmdata
```

5. Veri bütünlüğünü denetleyin.

- a) Test dosyasını biçimlendirin.

Sınanmakta olan dizinde büyük bir dosya yaratın. Dosya, sonraki aşamalar başarıyla tamamlanacak şekilde biçimlendirilir. Dosya, yedek sisteme geçiş benzetimi için ikinci evreyi kesmek için yeterli zaman olacak kadar büyük olmalıdır. 262144 sayfa (1 GB) varsayılan değerini deneyin. Program, yavaş dosya sistemlerinde bu varsayılan değeri otomatik olarak azaltır; böylece biçimleme yaklaşık 60 saniye içinde tamamlanır.

IBM MQ sunucusu 1 'de:

```
amqmfscck -f /shared/qmdata
```

Sunucu aşağıdaki iletilerle yanıt verir:

```
Formatting test file for data integrity test.
```

Test file formatted with 262144 pages of data.

b) Bir hataya neden olurken dosya sistemi denetleyicisini kullanarak verileri test dosyasına yazın.

Sinama programını aynı anda iki sunucuda çalıştırın. Hataya neden olacak sinama programını sunucuda başlatın ve daha sonra, hataya dayanacak sinama programını sunucuda başlatın. Araştırdığınız başarısızlık yüzünden.

İlk sinama programı bir hata iletişiyle durur. İkinci test programı, test dosyasındaki kilidi alır ve ilk test programının kapalı kaldığı yerden başlayarak verileri test dosyasına yazar. İkinci test programının tamamlanmasına izin verin.

<i>Çizelge 10. Veri bütünlüğü denetiminin aynı anda iki sunucuda çalıştırılması</i>	
IBM MQ Sunucu 1	IBM MQ Sunucu 2
amqmfscck -a /shared/qmdata	
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.	amqmfscck -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.

Testin zamanlaması, dosya sisteminin davranışına bağlıdır. Örneğin, bir dosya sisteminin güç kesintisinden sonra ilk program tarafından elde edilen dosya kilitlerini serbest bırakması genellikle 30-90 saniye sürer. İlk test programı dosyayı doldurmadan önce hatayı sunmak için çok az

zamanınız varsa, test dosyasını silmek için **amqmfscck -x** seçeneğini kullanın. Testi en baştan daha büyük bir test dosyasıyla deneyin.

c) Test dosyasındaki verilerin bütünlüğünü doğrulayın.

IBM MQ Server 2 'de:

```
amqmfscck -i /shared/qmdata
```

Sunucu aşağıdaki iletilerle yanıt verir:

```
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. Test dosyalarını silin.

IBM MQ Server 2 'de:

```
amqmfscck -x /shared/qmdata
Test files deleted.
```

Sunucu şu iletiyle yanıt verir:

```
Test files deleted.
```

Sonuçlar

Sinamalar başarıyla tamamlandıysa, program sıfır dışında bir çıkış kodu döndürür.

Örnekler

Üç örnekten oluşan ilk küme, en az çıkış üreten komutu gösterir.

Tek bir sunucuda temel dosya kilitleme sınaması başarılı oldu

```
> amqmfscck /shared/qmdata
The tests on the directory completed successfully.
```

Bir sunucuda temel dosya kilitleme sınaması başarısız oldu

```
> amqmfscck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck' error '2'.
```

İki sunucuda başarıyla kilitleme sınaması

Çizelge 11. İki sunucuda başarıyla kilitleme

IBM MQ Sunucu 1	IBM MQ Sunucu 2
<pre>> amqmfscck -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.</pre>	

Çizelge 11. İki sunucuda başarıyla kilitleme (devamı var)	
IBM MQ Sunucu 1	IBM MQ Sunucu 2
	> amqmfscck -w /shared/qmdata Waiting for lock...
[Return pressed] Lock released.	
	Lock acquired. The tests on the directory completed successfully

Üç örnekten oluşan ikinci küme, ayrıntılı kip kullanılarak aynı komutları gösterir.

Tek bir sunucuda temel dosya kilitleme sınaması başarılı oldu

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.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/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd1 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd1, F_SETLK, F_RDLCK)
System call: fd2 = open("/shared/qmdata/amqmfscck.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.
```

Bir sunucuda temel dosya kilitleme sınaması başarısız oldu

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.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/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_RDLCK)
System call: fdSameFile = open("/shared/qmdata/amqmfscck.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/amqmfscck.lck', errno 2
(Permission denied).
```

İki sunucuda başarıyla kilitleme sınaması

Çizelge 12. İki sunucuda başarıyla kilitleme-ayrıntılı kip

IBM MQ Sunucu 1	IBM MQ Sunucu 2
<pre>> amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.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.</pre>	
	<pre>> amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.lkw", O_EXCL O_CREAT O_RDWR,0666)' Calling 'fd = open("/shared/qmdata/amqmfscck.lkw, O_RDWR, 0666)' Calling 'fcntl(fd, F_SETLK, F_WRLCK) 'Waiting for lock...</pre>
<pre>[Return pressed] Calling 'close(fd)' Lock released.</pre>	
	<pre>Calling 'fcntl(fd, F_SETLK, F_WRLCK)' Lock acquired. The tests on the directory completed successfully</pre>

İlgili başvurular

[Yüksek kullanılabilirlikli örnek programlar](#)

Multi

İleti bütünlüğünü sınamak için amqsfhac çalıştırılıyor

Bir kuyruk yöneticisinin bir hata sırasında ileti bütünlüğünü korumasını göstermek için IBM MQ MQI client örnek programı **amqsfhac amqmfscck** ile paralel olarak çalıştırın.

Başlamadan önce

Bu sınamaya için dört sunucu gereklidir. Çok eşgörünümlü kuyruk yöneticisi için iki sunucu, biri dosya sistemi için ve biri de **amqsfhac** uygulamasını IBM MQ MQI client uygulaması olarak çalıştırmak için.

Çok eşgörünümlü bir kuyruk yöneticisine ilişkin dosya sistemini ayarlamak için "[Çoklu platformlarda paylaşılan dosya sistemi davranışının doğrulanması](#)" sayfa 114 içindeki "1" sayfa 115 adımı izleyin.

Bu görev hakkında

IBM MQ MQI client Örnek program **amqsfhac**, ağ üzerinden depolama kullanan bir kuyruk yöneticisinin bir hatanın ardından veri bütünlüğünü korumasını denetler. Bir kuyruk yöneticisinin bir hata sırasında ileti bütünlüğünü koruduğunu göstermek için **amqsfhac** komutunu **amqmfscck** ile paralel olarak çalıştırın.

Yordam

1. Yordam' daki “1” sayfa 115 . adımda yarattığınız dosya sistemini kullanarak başka bir sunucuda (QM1) çok eşgörsümlü bir kuyruk yöneticisi yaratın.
Bkz. Çok eşgörsümlü kuyruk yöneticisi yaratılması.
2. Kuyruk yöneticisini her iki sunucuda da başlatarak yüksek kullanılabilirlikli kılın.

Sunucu 1 'de:

```
strmqm -x QM1
```

Sunucu 2 'de:

```
strmqm -x QM1
```

3. **amqsfhac** programını çalıştırmak için istemci bağlantısını ayarlayın.
 - a) İşletmenizin bir istemci bağlantısı kurmak için kullandığı platform ya da platformlar için *IBM MQ kuruluşun doğrulanması* başlıklı konudaki yordamı ya da [Uzlaştırılabilir istemci örnekleri](#) içindeki örnek komut dosyalarını kullanın.
 - b) İstemci kanalını, QM1 çalıştıran iki sunucuya karşılık gelen iki IP adresine sahip olacak şekilde değiştirin.
Örnek komut dosyasında şunları değiştirin:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME('LOCALHOST(2345)') QMNAME(QM1) REPLACE
```

Kime:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME('server1(2345),server2(2345)') QMNAME(QM1) REPLACE
```

Burada *server1* ve *server2* , iki sunucunun anasistem adları ve 2345 , kanal dinleyicisinin dinlediği kapıdır. Bu genellikle 1414 varsayılan değerini alır. 1414 ' i varsayılan dinleyici yapılandırmasıyla kullanabilirsiniz.

4. Sınama için QM1 üzerinde iki yerel kuyruk yaratın.
Şu MQSC komut dosyasını çalıştır:

```
DEFINE QLOCAL(TARGETQ) REPLACE  
DEFINE QLOCAL(SIDEQ) REPLACE
```

5. Yapılandırmayı **amqsfhac** ile test edin

```
amqsfhac QM1 TARGETQ SIDEQ 2 2 2
```

6. Dosya sistemi bütünlüğünü test ederken ileti bütünlüğünü test edin.

[“Çoklu platformlarda paylaşılan dosya sistemi davranışının doğrulanması” sayfa 114](#) ile ilgili adım [“5” sayfa 115](#) sırasında **amqsfhac** komutunu çalıştırın.

```
amqsfhac QM1 TARGETQ SIDEQ 10 20 0
```

Etkin kuyruk yöneticisi yönetim ortamını durdurursanız, **amqsfhac** etkin olduktan sonra diğer kuyruk yöneticisi yönetim ortamına yeniden bağlanır. Durdurulan kuyruk yöneticisi eşgörsümünü yeniden başlatın; böylece sonraki sınavınızda hatayı tersine çevirebilirsiniz. Büyük olasılıkla, test programının hata durumunda yedek sisteme geçiş için yeterli süre çalışması için ortamınızdaki deneylere dayalı olarak yineleme sayısını artırmanız gerekir.

Sonuçlar

Aşağıdaki örnekte, “6” sayfa 120 adımında **amqsfhac** komutunu çalıştırma örneği gösterilmektedir. Bu örnekte, test bir başarıdır.

```
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
```

Sinema bir sorun saptadıysa, çıkış hatayı bildirir. Bazı test çalıştırmaları MQRC_CALL_INTERRUPTED "Resolving to backed out" raporunu sunabilir. Sonuç farketmez. Sonuç, diske yazma işleminin, hata oluşmadan önce ya da sonra ağa bağlı dosya depolaması tarafından kesinleştirilip kesinleştirilmemesine bağlıdır.

İlgili başvurular

amqmfsc (dosya sistemi denetimi)

[Yüksek kullanılabilirlikli örnek programlar](#)

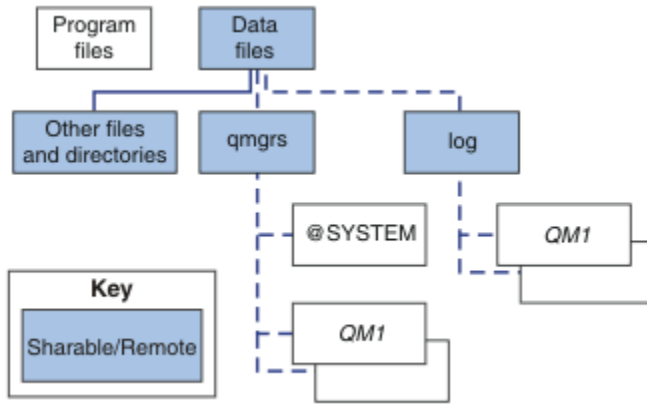
Multi

IBM MQ dosyalarını Multiplatforms üzerinde paylaşma

Bazı IBM MQ dosyalarına yalnızca etkin bir kuyruk yöneticisi tarafından erişilir, diğer dosyalar paylaşılır.

IBM MQ dosyaları, program dosyalarına ve veri dosyalarına bölünür. Program dosyaları genellikle IBM MQ çalıştıran her bir sunucuya yerel olarak kurulur. Kuyruk yöneticileri, varsayılan veri dizinindeki veri dosyalarına ve dizinlerine erişimi paylaşır. Bunlar, Şekil 32 sayfa 122 içinde gösterilen qmgrs ve log dizinlerinin her birinde bulunan kendi kuyruk yöneticisi dizin ağaçlarına özel erişim gerektirir.

Şekil 32 sayfa 122 , IBM MQ dizin yapısının üst düzey bir görünümüdür. Kuyruk yöneticileri arasında paylaşılabilen ve uzak yapılabilen dizinleri gösterir. Ayrıntılar platforma göre değişir. Noktalı çizgiler yapılandırılabilir yolları gösterir.



Şekil 32. IBM MQ dizin yapısının genel görünümü

Program dosyaları

Program dosyaları dizini genellikle varsayılan konumda bırakılır, yereldir ve sunucudaki tüm kuyruk yöneticileri tarafından paylaşılır.

Veri dosyaları

Veri dosyaları dizini genellikle varsayılan konumda yereldir; /var/mqm AIX and Linux sistemlerinde ve Windows üzerinde kurulu yapılandırılabilir. Kuyruk yöneticileri arasında paylaşılır. Varsayılan konumu uzak yapabilirsiniz, ancak farklı IBM MQ kuruluşları arasında paylaşmayın. IBM MQ yapılandırmasındaki DefaultPrefix özniteliği bu yolu gösterir.

qmgrs

Kuyruk yöneticisi verilerinin konumunu belirtmenin iki alternatif yolu vardır.

Prefix özniteliğini kullanma

Prefix özniteliği, qmgrs dizininin konumunu belirtir. IBM MQ , kuyruk yöneticisi adından kuyruk yöneticisi dizini adını oluşturur ve bunu qmgrs dizininin bir alt dizini olarak yaratır.

Prefix özniteliği, qms . ini dosyasının QueueManager kısmında bulunur ve Tüm Kuyruk Yöneticileri bölümü' nün **DefaultPrefix** özniteliğindeki değerden edinilir. Varsayılan olarak, yönetim basitliği için kuyruk yöneticileri genellikle aynı qmgrs dizinini paylaşır.

Herhangi bir kuyruk yöneticisi için qmgrs dizininin konumunu değiştirirseniz, **Prefix** özniteliğinin değerini değiştirmeniz gerekir.

Bir AIX and Linux platformu için Şekil 32 sayfa 122 içindeki QM1 dizinine ilişkin **Prefix** özniteliği şöyledir:

```
Prefix=/var/mqm
```

DataPath özniteliğini kullanma

DataPath özniteliği, kuyruk yöneticisi veri dizininin yerini belirtir.

DataPath özniteliği, kuyruk yöneticisi veri dizininin adını da içeren tam yolu belirtir.

DataPath özniteliği, kuyruk yöneticisi veri dizininin tamamlanmamış bir yolunu belirten **Prefix** özniteliğinden farklıdır.

Belirtildiyse, **DataPath** özniteliği, qms . ini kütüğünün QueueManager kısmında bulunur.

Belirtildiyse, **Prefix** özniteliğindeki herhangi bir değerden önceliklidir.

Herhangi bir kuyruk yöneticisi için kuyruk yöneticisi veri dizininin yerini değiştirirseniz, DataPath özniteliğinin değerini değiştirmeniz gerekir.

Bir Linux ya da AIX platformu için [Şekil 32 sayfa 122](#) içindeki QM1 dizinine ilişkin DataPath özneliği aşağıdaki gibidir:

```
DataPath=/var/mqm/qmgrs/QM1
```

Log

Günlük dizini, kuyruk yöneticisi yapılandırmasındaki [Günlük kısmı](#) içindeki her kuyruk yöneticisi için ayrı olarak belirtilir. Kuyruk yöneticisi yapılandırması `qm.ini` içinde yer almaktadır.

DataPath/QmgrName/@IPCC alt dizinleri

`DataPath/QmgrName/@IPCC` alt dizinleri paylaşılan dizin yolunda bulunur. Bunlar, IPC dosya sistemi nesnelere ilişkin dizin yolunu oluşturmak için kullanılır. Kuyruk yöneticisi sistemler arasında paylaşıldığında kuyruk yöneticisinin ad alanını ayırt etmeleri gerekir.

IPC dosya sistemi nesnelere sistem tarafından ayırt edilmesi gerekir. Kuyruk yöneticisinin çalıştığı her sistem için bir alt dizin, dizin yoluna eklenir (bkz. [Şekil 33 sayfa 123](#)).

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

Şekil 33. Örnek IPC alt dizini

`myHostName`, işletim sistemi tarafından döndürülen anasistem adının ilk 20 karakterine kadar. Bazı sistemlerde, anasistem adı kesilmeden önce en çok 64 karakter uzunluğunda olabilir. Oluşturulan `myHostName` değeri iki nedenden ötürü bir soruna neden olabilir:

1. İlk 20 karakter benzersiz değildir.
2. Anasistem adı, bir sisteme her zaman aynı anasistem adını ayırmayan bir DHCP algoritması tarafından oluşturulur.

Bu durumlarda, `MQS_IPC_HOST` ortam değişkenini kullanarak `myHostName` ayarını tanımlayın; bkz. [Şekil 34 sayfa 123](#).

```
export MQS_IPC_HOST= myHostName
```

Şekil 34. Örnek: MQS_IPC_HOST ayarını belirleme

Diğer dosyalar ve dizinler

İzleme dosyalarını içeren dizin ve genel hata günlüğü gibi diğer dosyalar ve dizinler genellikle yerel dosya sisteminde paylaşılır ve tutulur.

Paylaşılan dosya sistemleri desteğiyle IBM MQ, dosya sistemi kilitlerini kullanarak bu dosyalara özel erişimi yönetir. Dosya sistemi kilidi, belirli bir kuyruk yöneticisinin aynı anda yalnızca bir eşgörünümünün etkin olmasını sağlar.

Belirli bir kuyruk yöneticisinin ilk eşgörünümünü başlattığınızda, kuyruk yöneticisi dizininin sahipliğini alır. İkinci bir yönetim ortamı başlatacaksanız, bu yönetim ortamı ancak ilk yönetim ortamı durduysa sahipliği alabilir. İlk kuyruk yöneticisi çalışmaya devam ediyorsa, ikinci eşgörünüm başlatılamaz ve kuyruk yöneticisinin başka bir yerde çalıştığını bildirir. İlk kuyruk yöneticisi durduysa, ikinci kuyruk yöneticisi kuyruk yöneticisi dosyalarının sahipliğini devralır ve çalışan kuyruk yöneticisi olur.

Birinciden devralan ikinci kuyruk yöneticisinin yordamını otomatikleştirebilirsiniz. İlk kuyruk yöneticisini, başka bir kuyruk yöneticisinin bundan devralmasına izin veren `strmqm -x` seçeneğiyle başlatın. Daha sonra ikinci kuyruk yöneticisi, kuyruk yöneticisi dosyalarının sahipliğini devralmayı ve başlatmayı denemeden önce kuyruk yöneticisi dosyalarının kilidi açılana kadar bekler.

Linux

AIX

AIX and Linux sistemlerinde dizin yapısı

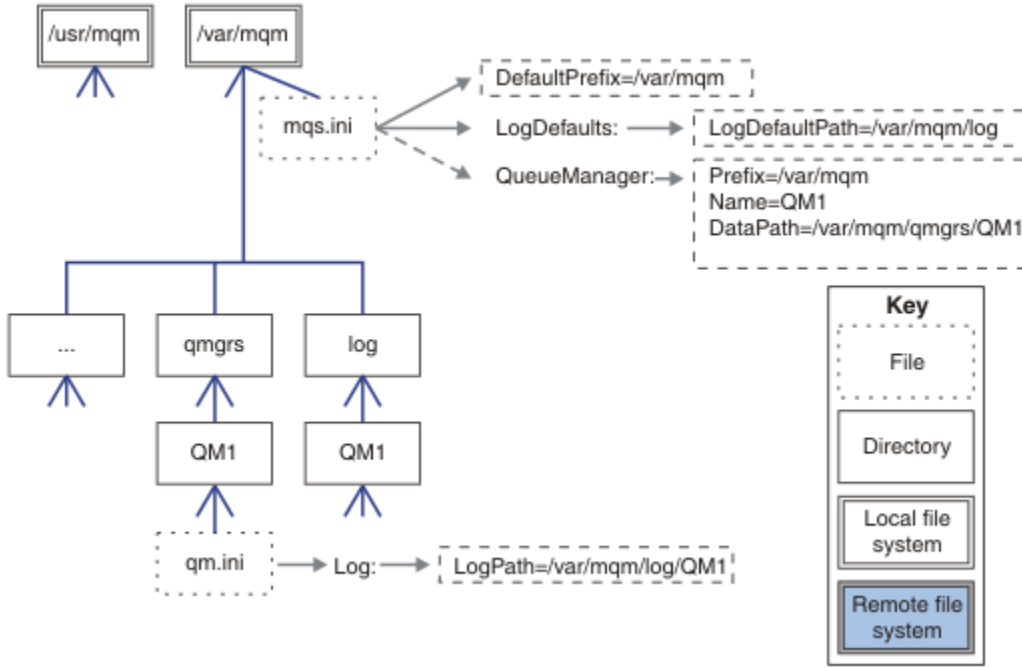
AIX and Linux sistemlerindeki IBM MQ dizin yapısı, daha kolay yönetim, daha iyi performans ve daha iyi güvenilirlik için farklı dosya sistemleriyle eşlenebilir.

Çok eşgörünümlü kuyruk yöneticilerinin çalıştırılması için paylaşılan dosya sistemlerinden yararlanmak üzere IBM MQ ' in esnek dizin yapısını kullanın.

Şekil 35 sayfa 124 içinde gösterilen dizin yapısını yaratmak için **crtmqm QM1** komutunu kullanın; burada R , ürünün yayın düzeyidir. Bu, IBM MQ sisteminde yaratılan bir kuyruk yöneticisi için tipik bir dizin yapısıdır. Bazı dizinler, dosyalar ve .ini özniteliği ayarları netlik için atlanır ve başka bir kuyruk yöneticisi adı, mangling yoluyla değiştirilebilir. Dosya sistemlerinin adları farklı sistemlere göre değişir.

Tipik bir kuruluştta, yarattığınız her kuyruk yöneticisi yerel dosya sistemindeki ortak log ve qmgrs dizinlerini gösterir. Çok eşgörünümlü bir yapılandırmada, log ve qmgrs dizinleri başka bir IBM MQkuruluşuyla paylaşılan bir ağ dosya sisteminde bulunur.

Şekil 35 sayfa 124 içinde AIX üzerinde IBM MQ v7.R için varsayılan yapılandırma gösterilir; burada R , ürünün yayın düzeyidir. Alternatif çok eşgörünümlü yapılandırma örnekleri için bkz. "AIX and Linux sistemlerinde örnek dizin yapılandırmaları" sayfa 129.



Şekil 35. AIX and Linux sistemleri için varsayılan IBM MQ dizin yapısı örneği

Ürün, varsayılan olarak AIX üzerinde /usr/mqm ve diğer sistemlerde /opt/mqm üzerine kurulur. Çalışma dizinleri /var/mqm dizinine kurulur.

Not: /var/mqm dosya sistemini IBM MQ kuruluşundan önce yarattıysanız, mqm kullanıcısının tam dizin izinlerine sahip olduğundan emin olun; örneğin, dosya kipi 755.

Not: Kuyruk yöneticisi tarafından üretilen FFC ' lerin /var/mqm'içeren dosya sistemini doldurmasını önlemek için /var/mqm/errors dizini ayrı bir dosya sistemi olmalıdır.

Ek bilgi için [AIX and Linux sistemlerinde dosya sistemleri yaratılması](#) başlıklı konuya bakın.

Log ve qmgrs dizinleri, mqs . ini dosyasındaki LogDefaultPath ve DefaultPrefix özniteliklerinin varsayılan değerleri tarafından tanımlandığı şekilde varsayılan konumlarında gösterilir. Bir kuyruk yöneticisi yaratıldığında, varsayılan olarak kuyruk yöneticisi veri dizini DefaultPrefix/qmgrsdizinde ve günlük dosyası dizini LogDefaultPath/logiçinde yaratılır. LogDefaultPath ve DefaultPrefix yalnızca kuyruk yöneticilerinin ve günlük dosyalarının varsayılan olarak oluşturulduğu etkileri. Bir kuyruk yöneticisi dizininin gerçek konumu mqs . ini dosyasına kaydedilir ve günlük dosyası dizininin konumu qm . ini dosyasına kaydedilir.

Bir kuyruk yöneticisine ilişkin günlük dosyası dizini, LogPath özniteliğindeki qm . ini dosyasında tanımlanır. Bir kuyruk yöneticisine ilişkin LogPath özniteliğini ayarlamak için **crtmqm** komutunda -ld

seçeneğini kullanın; örneğin, **crtmqm** -ld *LogPath* QM1 .ld parametresini çıkarırsanız, bunun yerine *LogDefaultPath* değeri kullanılır.

Kuyruk yöneticisi veri dizini, *mq.s.ini* dosyasındaki *QueueManager* kısmına ilişkin *DataPath* özniteliğinde tanımlanır. Bir kuyruk yöneticisine ilişkin *DataPath* değerini ayarlamak için **crtmqm** komutundaki -md seçeneğini kullanın; örneğin, **crtmqm** - md *DataPath* QM1 .md parametresini atlırsanız, bunun yerine *DefaultPrefix* ya da *Prefix* özniteliğinin değeri kullanılır. Önek , *DefaultPrefix*' den önceliklidir.

Genellikle, tek bir komutta hem günlük hem de veri dizinlerini belirterek QM1 oluşturun.

```
crtmqm  
-md DataPath -ld  
LogPath QM1
```

Kuyruk yöneticisi durdurulduğunda *qm.ini* dosyasında *DataPath* ve *LogPath* özniteliklerini düzenleyerek, var olan bir kuyruk yöneticisinin günlük ve veri dizinlerinin konumunu değiştirebilirsiniz.

errors dizininin yolu (*/var/mqm* içindeki diğer tüm dizinlerin yolları gibi) değiştirilemez. Ancak, dizinler farklı dosya sistemlerine bağlanabilir ya da sembolik olarak farklı dizinlere bağlanabilir.

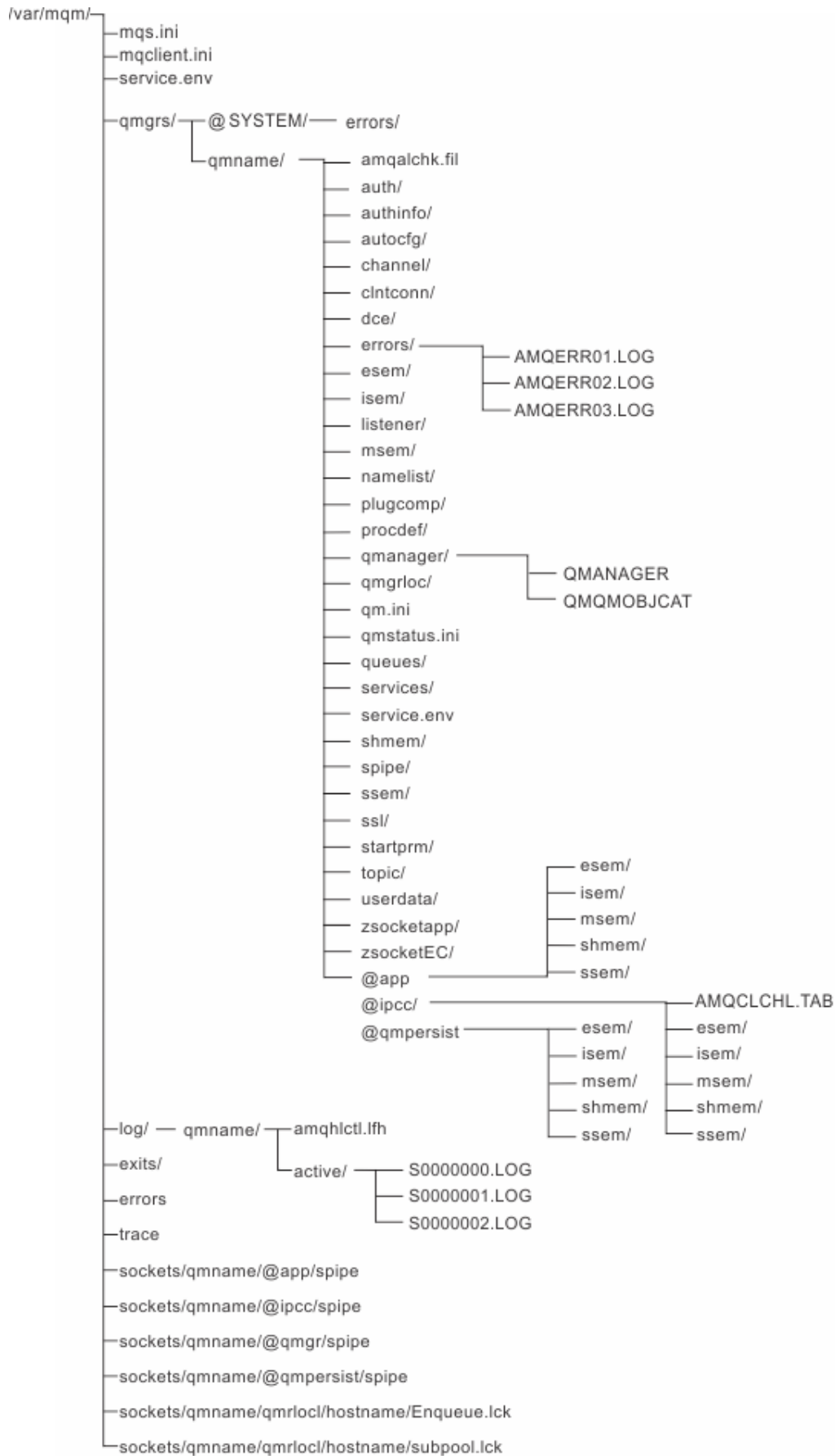
AIX and Linux sistemlerinde izin içeriği

Bir kuyruk yöneticisiyle ilişkili izinlerin içeriği.

Ürün dosyalarının konumu hakkında bilgi için bkz. [Kuruluş yeri seçilmesi](#)

Alternatif izin yapılandırmaları hakkında bilgi için bkz. [“Multiplatforms üzerinde dosya sistemi desteğinin planlanması” sayfa 111.](#)

Aşağıdaki izin yapısı, bir kuyruk yöneticisi bir süredir kullanıldıktan sonra IBM MQ ' in temsilcisidir. Sahip olduğunuz gerçek yapı, kuyruk yöneticisinde gerçekleşen işlemlere bağlıdır.



/var/mqm/

/var/mqm dizini, tek bir kuyruk yöneticisi için değil, bir IBM MQ kuruluşu için geçerli olan yapılandırma dosyalarını ve çıkış dizinlerini içerir.

<i>Çizelge 13. AIX and Linux üzerindeki /var/mqm dizininin belgelenmiş içeriği</i>	
Dizin ya da dosya adı	İçerik
<u>mqs.ini</u>	IBM MQ kuruluş çapında yapılandırma dosyası, bir kuyruk yöneticisi başlatıldığında okunur. AMQ_MQS_INI_LOCATION ortam değişkeni kullanılarak dosya yolu değiştirilebilir. Bunun ayarlandığından ve stirmqm komutunun çalıştırıldığı kabukta dışa aktarıldığından emin olun.
<u>mqclient.ini</u>	IBM MQ MQI client programları tarafından okunan varsayılan istemci yapılandırma dosyası. MQCLNTCF ortam değişkeni kullanılarak dosya yolu değiştirilebilir.
<u>service.env</u>	Bir hizmet sürecine ilişkin makine kapsamı ortam değişkenlerini içerir. Dosya yolu düzeltilmiştir.
<u>hatalar/</u>	Makine kapsamı hata günlükleri ve FFST dosyaları. Dizin yolu düzeltilmiştir. Ayrıca bkz. FFST: IBM MQ for UNIX ve Linux sistemleri .
<u>soketler/</u>	Yalnızca sistem kullanımı için her kuyruk yöneticisine ilişkin bilgileri içerir.
<u>izleme/</u>	İzleme dosyaları. Dizin yolu düzeltilmiştir.
<u>web/</u>	mqweb sunucusu dizini.
<u>çıkışlar/</u>	Kullanıcı kanalı çıkış programlarını içeren varsayılan dizin. mqs.ini dosyasındaki ApiExit stanzas içinde yer değiştirilebilir.
<u>exits64/</u>	

/var/mqm/qmgrs/qmname/

/var/mqm/qmgrs/qmname/ , bir kuyruk yöneticisine ilişkin dizinleri ve dosyaları içerir. Dizin, etkin kuyruk yöneticisi yönetim ortamı tarafından dışlayıcı erişim için kilitlendi. Dizin yolu, mqs . ini kütüğünde ya da **crtmqm** komutunun **md** seçeneği kullanılarak doğrudan değiştirilebilir.

<i>Çizelge 14. AIX and Linux üzerindeki /var/mqm/qmgrs/qmname dizininin belgelenmiş içeriği</i>	
Dizin ya da dosya adı	İçerik
<u>qm.ini</u>	Kuyruk yöneticisi yapılandırma dosyası, bir kuyruk yöneticisi başlatıldığında okunur.
<u>hatalar/</u>	Kuyruk yöneticisi kapsamı hata günlükleri. qmname = @system , bilinmeyen ya da kullanılmayan bir kuyruk yöneticisine ilişkin kanalla ilgili iletileri içerir.

Çizelge 14. AIX and Linux üzerindeki /var/mqm/qmgrs/qmname dizininin belgelenmiş içeriği (devamı var)	
Dizin ya da dosya adı	İçerik
@ipcc/ AMQCLCHL.TAB	IBM MQ sunucusu tarafından yaratılan ve IBM MQ MQI client programları tarafından okunan varsayılan istemci kanal denetimi çizelgesi. MQCHLLIB ve MQCHLTAB ortam değişkenleri kullanılarak dosya yolu değiştirilebilir.
qmanager	Kuyruk yöneticisi nesne dosyası: QMANAGER Kuyruk yöneticisi nesne kataloğu: QMQMOBJCAT
kimlikbilgisi/ kanal/ clntconn/ dinleyici/ namelist/ procdef/ kuyruklar/ hizmetler/ konular/	Kuyruk yöneticisinde tanımlanan her nesne, bu dizinlerdeki bir dosyayla ilişkilendirilir. Dosya adı tanımlama adıyla yaklaşık olarak eşleşiyor; bkz. IBM MQ dosya adlarını anlama .
...	IBM MQ tarafından kullanılan diğer dizinler (örneğin, @ipcc), yalnızca IBM MQ tarafından değiştirilecek.
kullanıcı verileri/	Uygulamaların kalıcı durumunu saklamak için kullanılabilir (kuyruk yöneticilerini farklı düğümlere taşırken RDQM tarafından kullanılabilir-bkz. Kalıcı uygulama durumunu saklama .)
DataPath\autocfg	Otomatik yapılandırma için kullanılır

/var/mqm/log/qmname/

/var/mqm/log/qmname/ , kuyruk yöneticisi günlük dosyalarını içerir. Dizin, etkin kuyruk yöneticisi yönetim ortamı tarafından dışlayıcı erişim için kilitlendi. Dizin yolu, qm.ini kütüğünde ya da **crtmqm** komutunun **ld** seçeneği kullanılarak değiştirilebilir.

Çizelge 15. AIX and Linux üzerindeki /var/mqm/log/qmname dizininin belgelenmiş içeriği	
Dizin ya da dosya adı	İçerik
amqhlctl.lfh	Günlük denetim dosyası.
etkin/	Bu dizin, S0000000.LOG, S0000001.LOG, S0000002.LOG, vb.

/opt/mqm

/opt/mqm varsayılan olarak çoğu platformda kuruluş dizinidir. İşletmenizin kullandığı platformda ya da platformlarda kuruluş dizini için gereken alan miktarı hakkında daha fazla bilgi için bkz. "[Çoklu platformlarda disk alanı gereksinimleri](#)" sayfa 108 .

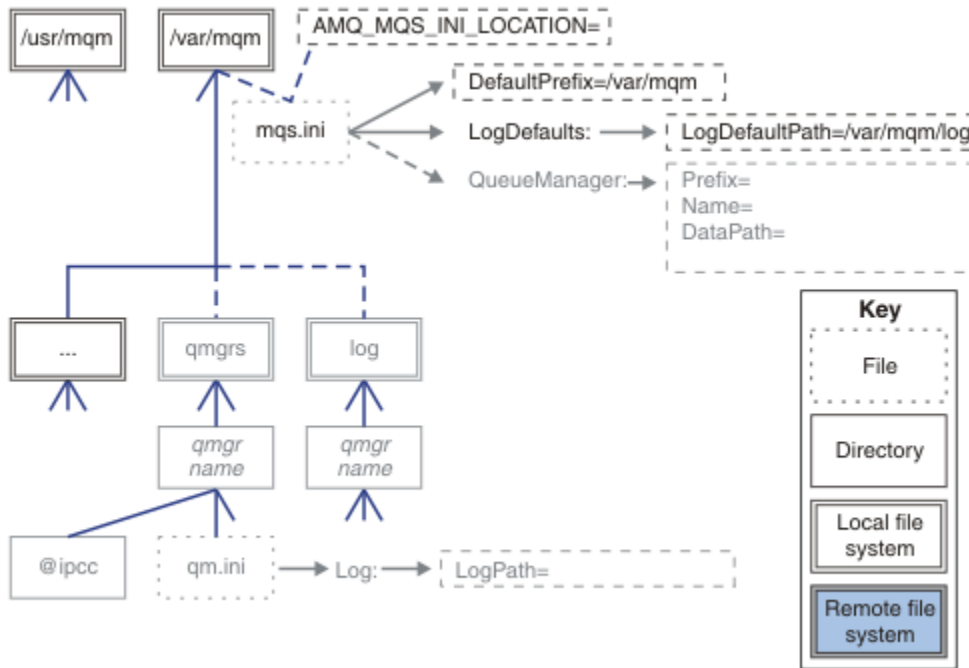
AIX and Linux sistemlerinde alternatif dosya sistemi yapılandırmalarına örnekler.

IBM MQ dizin yapısını çeşitli şekillerde uyarlayarak farklı hedeflere ulaşabilirsiniz.

- Çok eşgörünümlü bir kuyruk yöneticisi yapılandırmak için uzak paylaşılan kütük sistemlerine qmgrs ve log dizinlerini yerleştirin.
 - G/Ç çekişmesini azaltarak performansı artırmak için veri ve günlük dizinleri için ayrı dosya sistemleri kullanın ve dizinleri farklı disklere ayırın.
 - Performans üzerinde daha fazla etkisi olan dizinler için daha hızlı depolama aygıtları kullanın. Fiziksel aygıt gecikme süresi, bir aygıtın yerel olarak mı, yoksa uzaktan mı monte edildiğinden daha kalıcı ileti sistemi performansında sıklıkla daha önemli bir faktördür. Aşağıdaki listede, hangi dizinlerin en çok ve en az başarımlar açısından hassas olduğu gösterilmektedir.
1. log
 2. qmgrs
 3. /usr/mqm da içinde olmak üzere diğer dizinler
- Örneğin, yedek disk dizisi gibi iyi bir esneklikle depolamaya ayrılan dosya sistemlerinde qmgrs ve log dizinlerini oluşturun.
 - Ağ dosya sistemiyle ilgili hatanın günlüğe kaydedilmesi için, genel hata günlüklerini bir ağ dosya sisteminde değil, yerel olarak var/mqm/errors' da saklamanız daha iyi olur.

Şekil 36 sayfa 129 , alternatif IBM MQ dizin yapılarının türetildiği bir şablondur. Şablonda, noktalı çizgiler yapılandırılabilir yolları gösterir. Örneklerde, noktalı çizgiler, AMQ_MQS_INI_LOCATION ortam değişkeninde ve mqs.ini ve qm.ini dosyalarında saklanan yapılandırma bilgilerine karşılık gelen dolu satırlarla değiştirilir.

Not: Yol bilgileri, mqs.ini ya da qm.ini dosyalarında görüldüğü şekilde gösterilir. **crtmqm** komutunda yol parametreleri sağlarsanız, kuyruk yöneticisi dizininin adını atlayın: Kuyruk yöneticisi adı, IBM MQ ile yola eklenir.



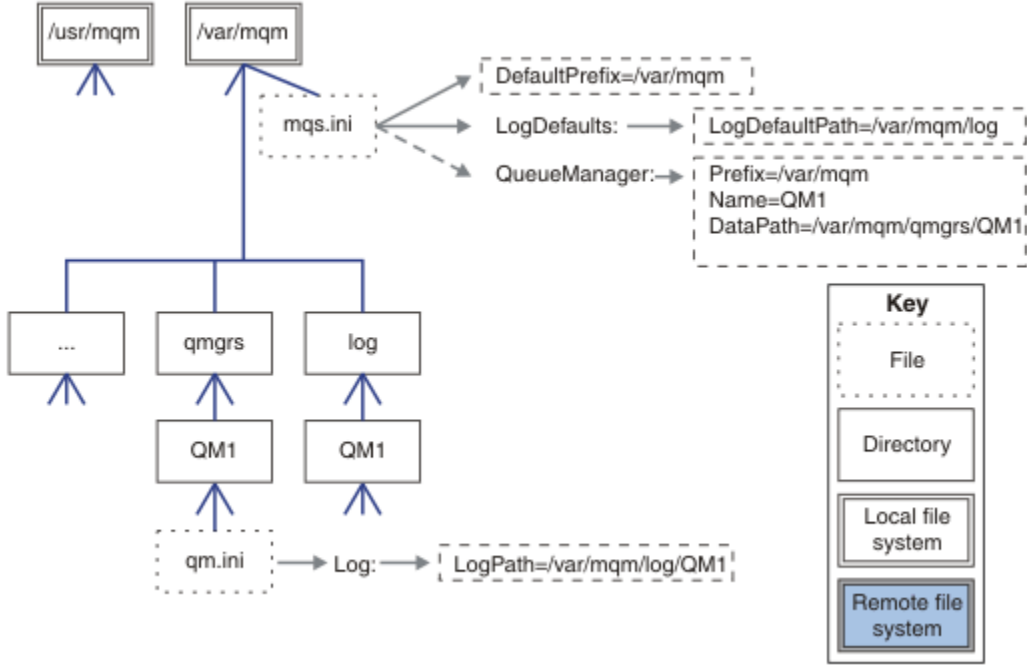
Şekil 36. Dizin yapısı örüntüsü şablonu

IBM MQ için tipik dizin yapısı

Şekil 37 sayfa 130 , IBM MQ içinde **crtmqm QM1** komutu verilerek oluşturulan varsayılan dizin yapısıdır.

mqs.ini dosyası, DefaultPrefix değerine başvurarak oluşturulan QM1 kuyruk yöneticisi için bir bölüme sahiptir. qm.ini dosyasındaki Log (Günlük) kısmı, mqs.ini içindeki LogDefaultPath ögesine başvurarak ayarlanan LogPath için bir değere sahiptir.

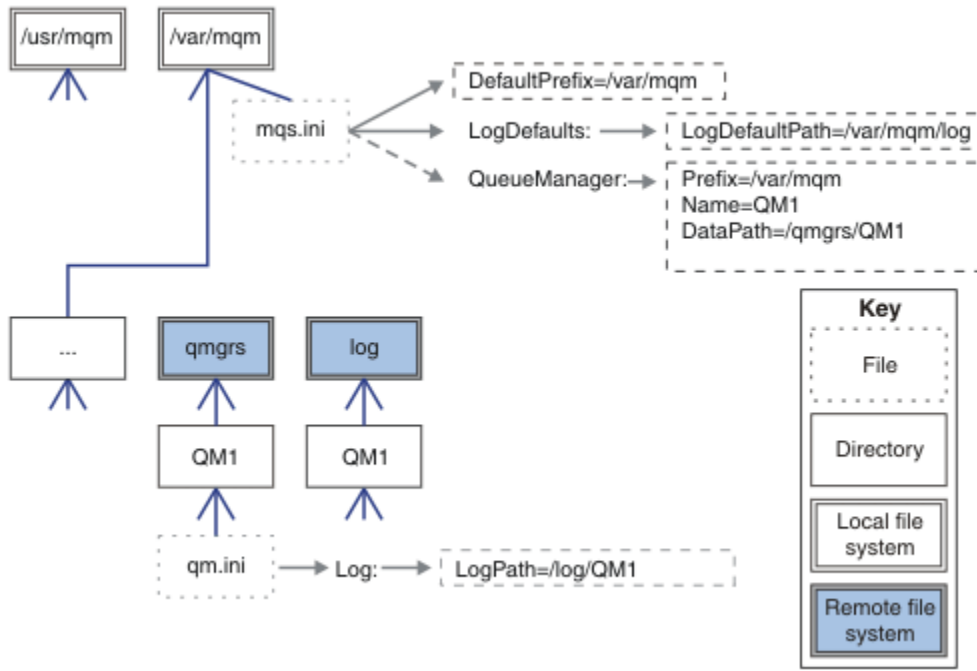
DataPath ve LogPath varsayılan değerlerini geçersiz kılmak için isteğe bağlı **crtmqm** parametrelerini kullanın.



Şekil 37. AIX and Linux sistemleri için varsayılan IBM MQ dizin yapısı örneği

Varsayılan qmgrs ve log dizinlerini paylaş

“Her şeyi paylaş” sayfa 132 için bir alternatif, qmgrs ve log dizinlerini ayrı olarak paylaşmaktır (Şekil 38 sayfa 131). Bu yapılandırmada, yerel /var/mqm dosya sisteminde varsayılan mqs.ini saklandığı için AMQ_MQS_INI_LOCATION ayarlanmasına gerek yoktur. mqclient.ini ve mqserver.ini gibi dosyalar ve dizinler de paylaşılmaz.

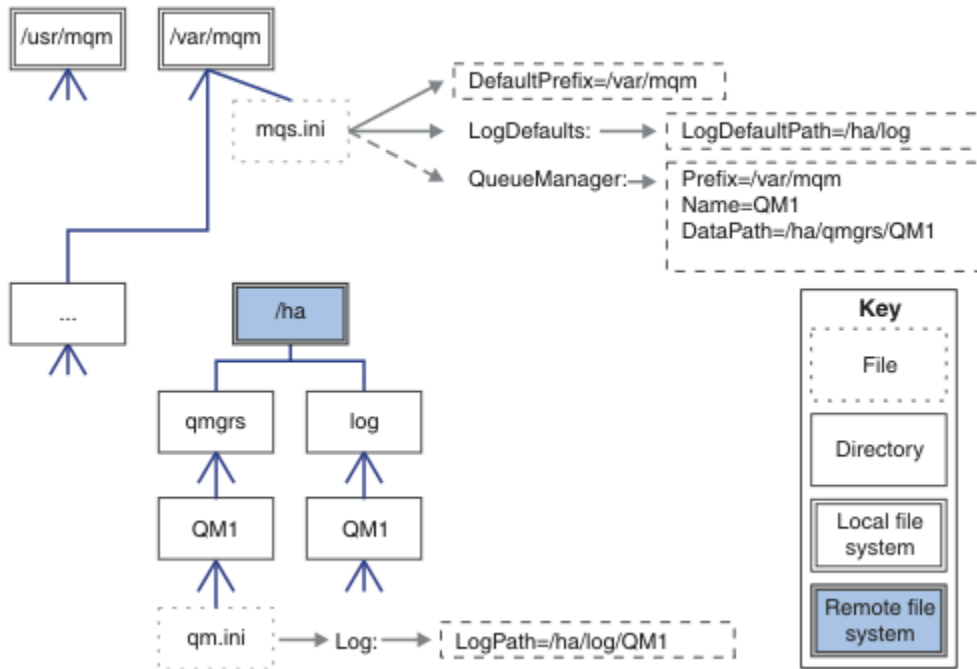


Şekil 38. qmgrs ve log dizinlerini paylaşma

qmgrs ve log dizinlerini paylaşma

Şekil 39 sayfa 131 içindeki yapılandırma, log ve qmgrs dosyasını /ha adlandırılmış ortak bir uzak paylaşılan dosya sistemine yerleştirir. Aynı fiziksel yapılandırma iki farklı şekilde oluşturulabilir.

1. LogDefaultPath=/ha değerini ayarlayın ve **crtmqm - md /ha/qmgrs QM1** komutunu çalıştırın. Sonuç tam olarak Şekil 39 sayfa 131 içinde gösterildiği gibi olur.
2. Varsayılan yolları değiştirmeden bırakın ve **crtmqm - ld /ha/log - md /ha/qmgrs QM1** komutunu çalıştırın.



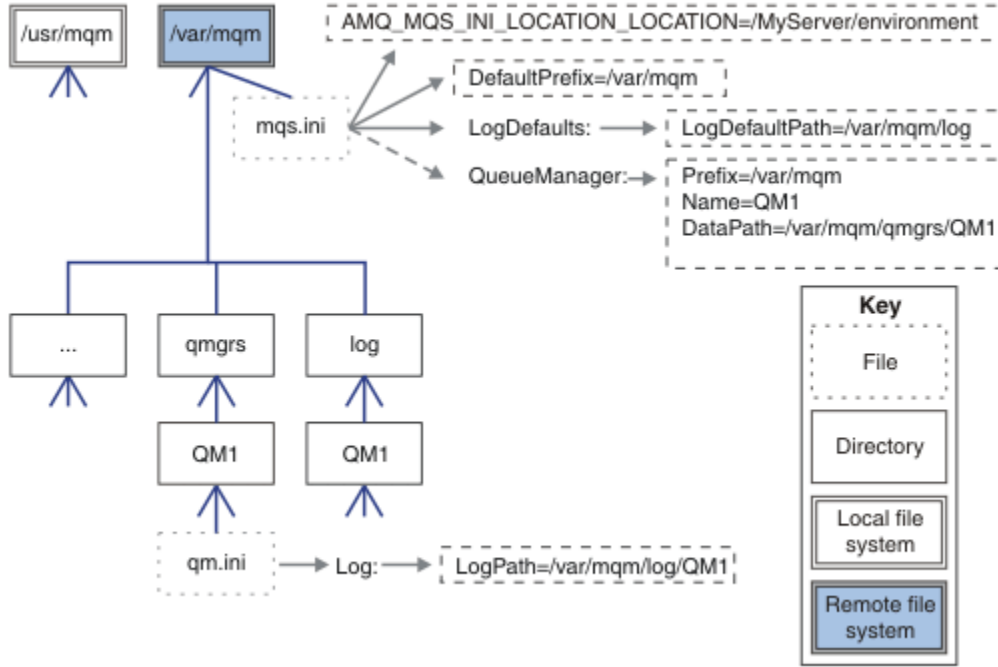
Şekil 39. qmgrs ve log dizinlerini paylaşma

Her şeyi paylaş

Şekil 40 sayfa 132 , hızlı ağ bağlantılı dosya depolamasına sahip sistem için basit bir yapılandırmadır.

Uzak paylaşılan kütük sistemi olarak /var/mqm dosyasını sisteme bağlayın. Varsayılan olarak, QM1'i başlattığınızda, /var/mqm' i arar, paylaşılan dosya sisteminde bulur ve mqs . ini dosyasını /var/mqm içinde okur. Tüm sunucularınızdaki kuyruk yöneticileri için tek /var/mqm/mqs . ini dosyasını kullanmak yerine, her sunucudaki AMQ_MQS_INI_LOCATION ortam değişkenini farklı mqs . ini dosyalarını gösterecek şekilde ayarlayabilirsiniz.

Not: /var/mqm/errors/ içindeki soysal hata dosyasının içeriği, farklı sunuculardaki kuyruk yöneticileri arasında paylaşılır.



Şekil 40. Her şeyi paylaş

Bunu çok eşgörunümlü kuyruk yöneticileri için kullanamayacağınız unutmayın. Bunun nedeni, çok eşgörunümlü bir kuyruk yöneticisindeki her bir anasistemin, semaforlar ve paylaşılan bellek gibi yerel verileri izlemek için kendi yerel /var/mqm kopyasına sahip olması gerekmesi olabilir. Bu varlıklar anasistemler arasında paylaşılabilir.

Windows Windows sistemlerinde dizin yapısı

Windows' da kuyruk yöneticisi yapılanış bilgilerini ve dizinlerini bulma.

IBM MQ for Windows kuruluşu için varsayılan dizinler şunlardır:

Program dizini

C:\Program Files\IBM\MQ

Veri Dizini

C:\ProgramData\IBM\MQ

Önemli: Windows Windows kuruluşlarında, ürünün kayıt girişleri ve/ya da kuyruk yöneticileri içeren önceki bir kuruluşu yoksa, dizinler belirtildiği gibi olur. Bu durumda, yeni kuruluş eski veri dizini yerini kullanır. Daha fazla bilgi için bkz. [Program ve veri dizini konuları](#).

Hangi kuruluş dizininin ve hangi veri dizininin kullanıldığını öğrenmek istiyorsanız, `dspmqr` komutunu çalıştırın.

Kuruluş dizini **InstPath** alanında listelenir ve veri dizini **DataPath** alanında listelenir.

dspmqr komutu çalıştırılırken, örneğin, aşağıdaki bilgiler görüntülenir:

```
>dspmqr
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
```

Çok eşgörunümlü kuyruk yöneticileri

Çok eşgörunümlü bir kuyruk yöneticisini yapılandırmak için, günlük ve veri dizinlerinin, tercihen kuyruk yöneticisinin eşgörunümlerini çalıştıran sunuculardan farklı bir sunucuya yerleştirilmesi gerekir.

Kuyruk yöneticisi verilerinin ve günlük dizinlerinin konumunu daha kolay belirtmek için **crtmqm** komutunda **-md** ve **-ld** olmak üzere iki parametre sağlanır. **-md** değiştirgesini belirtmenin etkisi dört katlıdır:

1. mqs.ini kısmı QueueManager\QmgrName , kuyruk yöneticisi veri dizinini gösteren yeni bir değişken (DataPath) içerir. Prefix (Önek) değişkeninden farklı olarak, yol kuyruk yöneticisi dizininin adını içerir.
2. mqs.ini dosyasında saklanan kuyruk yöneticisi yapılandırma bilgileri Ad, Önek, Dizin ve DataPath olarak azaltılır.

Windows Dizin içeriği

IBM MQ dizinlerinin konumunu ve içeriğini listeler.

Bir IBM MQ yapılandırmasında üç ana dosya ve dizin kümesi vardır:

1. Yürütülür dosya ve yalnızca bakım uygulandığında güncellenen diğer salt okunur dosyalar. Örneğin:
 - Readme (benioku) dosyası
 - IBM MQ Explorer eklentisi ve yardım dosyaları
 - Lisans dosyalarıBu dosyalar [Çizelge 16 sayfa 133](#) içinde açıklanmıştır.
2. Belirli bir kuyruk yöneticisine özgü olmayan, değiştirilebilecek dosyalar ve dizinler. Bu dosyalar ve dizinler [Çizelge 17 sayfa 134](#) içinde açıklanmıştır.
3. Bir sunucudaki her kuyruk yöneticisine özgü dosyalar ve dizinler. Bu dosyalar ve dizinler [Çizelge 18 sayfa 135](#) içinde açıklanmıştır.

Kaynak dizinleri ve dosyalar

Kaynak dizinleri ve dosyalar, bir kuyruk yöneticisini çalıştırmak için gereken tüm yürütülebilir kodları ve kaynakları içerir. Kuruluşa özgü IBM MQ yapılandırma kayıt anahtarındaki *FilePath* değişkeni, kaynak dizinlerinin yolunu içerir.

Çizelge 16. FilePath dizinindeki dizinler ve dosyalar	
Dosya yolu	İçerik
FilePath\bin	Komutlar ve DLL ' ler
FilePath\bin64	Komutlar ve DLL ' ler (64 bit)
FilePath\conv	Veri dönüştürme çizelgeleri

Çizelge 16. <i>FilePath</i> dizinindeki dizinler ve dosyalar (devamı var)	
Dosya yolu	İçerik
<i>FilePath</i> \doc	Sihirbaz yardım dosyaları
<i>FilePath</i> \MQExplorer	Explorer ve Explorer yardımcı Eclipse eklentileri
<i>FilePath</i> \gskit8	Genel güvenlik seti
<i>FilePath</i> \java	JRE de içinde olmak üzere Java kaynakları
<i>FilePath</i> \licenses	Lisans bilgisi
<i>FilePath</i> \Non_IBM_License	Lisans bilgisi
<i>FilePath</i> \properties	Dahili olarak kullanılır
<i>FilePath</i> \Tivoli	
<i>FilePath</i> \tools	Geliştirme kaynakları ve örnekleri
<i>FilePath</i> \web	IBM MQ Console ve REST API düzenlenemeyen dosyalar için kuruluş bileşeni dosya yapısına başlıklı konuda açıklanmıştır.
<i>FilePath</i> \Uninst	Dahili olarak kullanılır
<i>FilePath</i> \README.TXT	Readme dosyası

Bir kuyruk yöneticisine özgü olmayan dizinler

Bazı dizinler, belirli bir kuyruk yöneticisine özgü olmayan izleme dosyaları ve hata günlükleri gibi dosyaları içerir. *DefaultPrefix* değişkeni, bu dizinlerin yolunu içerir. *DefaultPrefix*, *AllQueueManagers* kısmının bir parçasıdır.

Çizelge 17. <i>DefaultPrefix</i> dizinindeki dizinler ve dosyalar	
Dosya yolu	İçerik
<i>DefaultPrefix</i> \config	Dahili olarak kullanılır
<i>DefaultPrefix</i> \conv	ccsid_part2.tbl ve ccid.tbl data dönüştürme denetim dosyası; açıklamalar için bkz. Veri dönüştürme
<i>DefaultPrefix</i> \errors	Kuyruk yöneticisi olmayan hata günlükleri, AMQERR nn.LOG
<i>DefaultPrefix</i> \exits	Kanal çıkış programları
<i>DefaultPrefix</i> \exits64	Kanal çıkış programları (64 bit)
<i>DefaultPrefix</i> \ipc	Kullanılmadı
<i>DefaultPrefix</i> \qmgs	Çizelge 18 sayfa 135 içinde açıklanmıştır
<i>DefaultPrefix</i> \trace	iz dosyaları
<i>DefaultPrefix</i> \web	IBM MQ Console ve REST API kullanıcı tarafından düzenlenebilir dosyalar için kuruluş bileşeni dosya yapısına başlıklı kısımlarda açıklanmıştır.
<i>DefaultPrefix</i> \amqmjpse.txt	Dahili olarak kullanılır

Kuyruk yöneticisi dizinleri

Bir kuyruk yöneticisi yarattığınızda, kuyruk yöneticisine özgü yeni bir dizin kümesi yaratılır.

-**md filepath** deęiřtirgesiyle bir kuyruk yneticisi yaratırsanız, yol mq.s.ini dosyasının kuyruk yneticisi kısmına iliřkin *DataPath* deęiřkeninde saklanır. -**md filepath** deęiřtirgesini ayarlamadan bir kuyruk yneticisi oluřturursanız, kuyruk yneticisi dizinleri *DefaultPrefix*inde saklanan yolda yaratılır ve yol, mq.s.ini dosyasının kuyruk yneticisi kısmına *Prefix* (nek) deęiřkenine kopyalanır.

Çizelge 18. <i>DataPath</i> ve <i>Prefix\qmgrs\QmgrName</i> dizinlerindeki dizinler ve dosyalar	
Dosya yolu	İçerik
<i>DataPath\@ipcc</i>	AMQCLCHL . TABiçin varsayılan konum, istemci baęlantı tablosu.
<i>DataPath\authinfo</i>	Dahili olarak kullanılır.
<i>DataPath\channel</i>	
<i>DataPath\clntconn</i>	
<i>DataPath\errors</i>	Hata gnlkleri, AMQERR <i>nn</i> . LOG
<i>DataPath\listener</i>	Dahili olarak kullanılır.
<i>DataPath\namelist</i>	
<i>DataPath\plugcomp</i>	
<i>DataPath\procdef</i>	
<i>DataPath\qmanager</i>	
<i>DataPath\queues</i>	
<i>DataPath\services</i>	
<i>DataPath\ssl</i>	
<i>DataPath\startprm</i>	
<i>DataPath\topic</i>	
<i>DataPath\active</i>	
<i>DataPath\active.dat</i>	
<i>DataPath\amqalchk.fil</i>	
<i>DataPath\master</i>	
<i>DataPath\master.dat</i>	
<i>DataPath\qm.ini</i>	Kuyruk yneticisi yapısı
<i>DataPath\qmstatus.ini</i>	Kuyruk yneticisi durumu
<i>DataPath\userdata</i>	Uygulamaların kalıcı durumunu saklamak iin kullanılabilir.
<i>Prefix\qmgrs\QmgrName</i>	Dahili olarak kullanılır
<i>Prefix\qmgrs\@SYSTEM</i>	Kullanılmadı
<i>Prefix\qmgrs\@SYSTEM\errors</i>	
<i>DataPath\autocfg</i>	Otomatik yapılandırma iin kullanılır

IBM i **IBM i** üzerinde dizin yapısı

IFS ' nin bir aıklaması verilir ve sunucu, istemci ve Java'iin IBM MQ IFS dizin yapısı aıklanır.

Tümleşik dosya sistemi (IFS), IBM i ' in bir parçasıdır ve sunucuda saklanan tüm bilgiler üzerinde bir bütünleştirme yapısı sağlarken AIX and Linux işletim sistemlerine benzer şekilde akış giriş/çıkışını ve depolama yönetimini destekler.

IBM i dizininde dizin adları, @ (at) karakteri yerine & (ampersand) karakteriyle başlar. Örneğin, IBM i üzerinde @system &system.

IBM MQ sunucusu için IFS kök dosya sistemi

IBM MQ Server for IBM ürününü kurduğunuzda, IFS kök dosya sisteminde aşağıdaki dizinler yaratılır.

ProdData:

Genel Bakış

QIBM

```
'-- ProdData
    '-- mqm
    '-- doc
    '-- inc
    '-- lib
    '-- samp
    '-- licenses
    '-- LicenseDoc
    '-- 5724H72_V8R0M0
```

/QIBM/ProdData/mqm

Bunun altındaki alt dizinler, C++ sınıfları, izleme biçimi dosyaları ve lisans dosyaları gibi tüm ürün verilerini içerir. Bu dizindeki veriler, ürün her kurulduğunda silinir ve değiştirilir.

/QIBM/ProdData/mqm/doc

CL komutlarına ilişkin bir Komut Başvurusu HTML biçiminde sağlanır ve burada kurulur.

/QIBM/ProdData/mqm/inc

C ya da C++ programlarınızı derlemek için kullanılan üstbilgi dosyaları.

/QIBM/ProdData/mqm/lib

MQ tarafından kullanılan yardımcı dosyalar.

/QIBM/ProdData/mqm/samp

Başka örnekler.

/QIBM/ProdData/mqm/licenses

Lisans dosyaları. Her dil için iki dosya LA_ *xx* ve LI_ *xx* gibi adlandırılır; burada *xx* , sağlanan her dil için 2 karakterlik dil tanıtıcısıdır.

Ayrıca, aşağıdaki dizin lisans sözleşmeleri dosyalarını da saklar:

/QIBM/ProdData/LicenseDoc/5724H72_V8R0M0

Lisans dosyaları. Dosyalar 5724H72_V8R0M0_ *xx* şeklinde adlandırılır; burada *xx* , sağlanan her dil için 2 ya da 5 karakterlik dil tanıtıcısıdır.

UserData:

Genel Bakış

QIBM

```
'-- UserData
    '-- mqm
    '-- errors
    '-- trace
    '-- qmgrs
    '-- &system
    '-- qmgrname1
```



```
'-- qmgrname2
'-- and so on
```

/QIBM/UserData/mqm

Bunun altındaki alt dizinler, kuyruk yöneticileriyle ilgili tüm kullanıcı verilerini içerir.

Ürünü kurduğunuzda, /QIBM/UserData/mqm/ dizininde bir mqm.ini dosyası yaratılır (önceki bir kuruluştaki yoksa).

Bir kuyruk yöneticisi yarattığınızda, /QIBM/UserData/mqm/qmgrs/ QMGRNAME /dizininde bir qm.ini dosyası yaratılır (burada QMGRNAME kuyruk yöneticisinin adıdır).

Ürün silindiğinde dizinlerdeki veriler korunur.

IBM MQ MQI client için IFS kök dosya sistemi

IBM MQ MQI client for IBM i' i kurduğunuzda, IFS kök dosya sisteminde oluşturulan aşağıdaki dizinler:

ProdData:

Genel Bakış

QIBM

```
'-- ProdData
'-- mqm
'-- lib
```

/QIBM/ProdData/mqm

Bu dizinin altındaki alt dizinler tüm ürün verilerini içerir. Bu dizindeki veriler, ürün her değiştirildiğinde silinir ve değiştirilir.

UserData:

Genel Bakış

QIBM

```
'-- UserData
'-- mqm
'-- errors
'-- trace
```

/QIBM/UserData/mqm

Bu dizinin altındaki alt dizinler tüm kullanıcı verilerini içerir.

IBM MQ Java için IFS kök dosya sistemi

IBM MQ Java ürününü IBM i üzerine kurduğunuzda, IFS kök dosya sisteminde aşağıdaki dizinler oluşturulur:

ProdData:

Genel Bakış

QIBM

```
'-- ProdData
'-- mqm
'-- java
'-- samples
'-- bin
'-- lib
```

/QIBM/ProdData/mqm/java

Bunun altındaki alt dizinler, Java sınıfları da içinde olmak üzere tüm ürün verilerini içerir. Bu dizindeki veriler, ürün her değiştirildiğinde silinir ve değiştirilir.

/QIBM/ProdData/mqm/java/samples

Bunun altındaki alt dizinler tüm örnek Java sınıflarını ve verilerini içerir.

Sunucu ve istemci kuruluşları tarafından yaratılan kitaplıklar

IBM MQ sunucusunun ya da istemcisinin kuruluşu aşağıdaki kitaplıkları yaratır:

- QMQM
Ürün kitaplığı.
- QMQMSAMP
Örnekler kitaplığı (örnekleri kurmayı seçerseniz).
- QMxxxx
Yalnızca sunucu.

Bir kuyruk yöneticisini her yarattığınızda, IBM MQ otomatik olarak, QMxxxx gibi bir adla ilişkili bir kitaplık yaratır; burada xxxx , kuyruk yöneticisi adından türetilir. Bu kitaplık, günlükler ve ilişkili günlük nesnelere de içinde olmak üzere kuyruk yöneticisine özgü nesnelere içerir. Varsayılan olarak, bu kitaplığın adı kuyruk yöneticisinin adının başında QM karakterlerinin bulunduğu addan türetilir. Örneğin, TEST adlı bir kuyruk yöneticisi için, kitaplığa QMTEST adı verilir.

Not: Bir kuyruk yöneticisi yarattığınızda, isterseniz, bu yöneticinin kitaplığının adını belirtebilirsiniz. Örneğin:

```
CRTMQM MQMNAME(TEST) MQMLIB(TESTLIB)
```

IBM MQ for IBM i tarafından yaratılan tüm kitaplıkları listelemek için WRKLIB komutunu kullanabilirsiniz. Kuyruk yöneticisi kitaplıklarında QMGR: QMGRNAMEmetnini görürsünüz. Komutun biçimi şöyledir:

```
WRKLIB LIB(QM*)
```

Bu kuyruk yöneticisiyle ilişkili kitaplıklar, ürün silindiğinde alıkonur.

Multi Çoklu Platformlar üzerinde MFT için planlama dosya sistemi desteği

IBM MQ Managed File Transfer MFT araçları, bir dosya sistemindeki dosyalara/dosyalardan veri aktarmak için kullanılabilir. Buna ek olarak, bir aracı içinde çalışan kaynak izleyicilerinin konfigürasyonu bir dosya sistemindeki dosyaları izleyecek şekilde tanımlanabilir.

MFT ' in bu dosyaların kilitlemeyi destekleyen bir dosya sisteminde saklanması gereksinimi vardır. Bunun iki nedeni vardır:

- Aracı, bir dosyadan veri okumaya ya da dosyaya veri yazmaya başladıktan sonra dosyanın değişmediğinden emin olmak için dosyayı kilitlet.
- Kaynak izleyicileri, başka bir sürecin bunları kullanmadığını denetlemek için dosyaları kilitlet.

Araçlar ve kaynak izleyicileri kilitleme gerçekleştirmek için Java yöntemi **FileChannel.tryLock()** kullanır ve dosya sistemi bu çağrıyı kullanarak dosyaları kilitleyebilmelidir.

Önemli: MFT' in teknik gereksinimlerini karşılamadıkları için aşağıdaki dosya sistemleri desteklenmez:

- GlusterFS
- NFS sürüm 3

Multiplatforms üzerinde dairesel ya da doğrusal günlük kaydının seçilmesi

IBM MQÇiçinde dairesel ya da doğrusal günlük kaydını seçebilirsiniz. Aşağıdaki bilgiler, her iki tipe ilişkin bir genel bakış sağlar.

Dairesel günlük kaydının avantajları

Dairesel günlüğe kaydetmenin başlıca avantajları şudur:

- Daha kolay yönetilebilir.

İş yükünüz için döngüsel günlük kaydını doğru şekilde yapılandırdıktan sonra, başka bir yönetim gerekmez. Oysa, doğrusal günlük kaydı için, artık gerekli olmayan ortam görüntülerinin kaydedilmesi ve günlük kapsamlarının arşivlenmesi veya silinmesi gerekir.

- Daha iyi performans

Çevrimsel günlük kaydı, önceden biçimlendirilmiş günlük kapsamlarını yeniden kullanabildiğinden, çevrimsel günlük kaydı doğrusal günlük kaydından daha iyi çalışır. Oysa doğrusal günlük kaydı, yeni günlük kapsamları ayırmalı ve bunları biçimlendirmeli.

Daha fazla bilgi için [Günlüklerin yönetilmesi](#) başlıklı konuya bakın.

Doğrusal günlük kaydının avantajları

Doğrusal günlük kaydının temel avantajı, doğrusal günlük kaydının daha fazla hataya karşı koruma sağlamasıdır.

Çevrimsel ya da doğrusal günlük kaydı, bozuk ya da silinmiş bir günlüğe ya da uygulamalar ya da denetimci tarafından silinmiş iletilere ya da kuyruklara karşı korunmaz.

Doğrusal günlüğe kaydetme (dairesele değil), hasarlı nesnelere kurtarılmasını sağlar. Bu nedenle, doğrusal günlük kaydı, bu hasarlı kuyruklar doğrusal bir günlükten kurtarılabildiğinden, kuyruk dosyalarının bozulmasına ya da silinmesine karşı koruma sağlar.

Recovering from power loss or communication failure(Güç kaybından ya da iletişim hatalarından kurtarma) başlıklı konuda açıklandığı gibi, güç kaybına ve iletişim arızalarına karşı hem dairesel hem de doğrusal koruma.

Diğer önemli noktalar

Doğrusal ya da dairesel seçip seçmediğiniz, ne kadar yedeklilik gerektirdiğinize bağlıdır.

Performans maliyeti ve yönetim maliyetinden kaynaklanan, doğrusal günlük kaydı olan daha fazla yedekliliği seçmenin bir maliyeti vardır.

Ek bilgi için [Günlük kaydı tipleri](#) konusuna bakın.

AIX üzerinde paylaşılan bellek

Bazı uygulama tipleri AIX bellek sınırlaması nedeniyle bağlanamazsa, çoğu durumda bu, EXTSHM=ON ortam değişkenini ayarlanarak çözülebilir.

AIX üzerindeki bazı 32 bit işlemler, IBM MQ kuyruk yöneticilerine bağlanma yeteneklerini etkileyen bir işletim sistemi sınırlamasından geçebilir. IBM MQ ile kurulan her standart bağlantı paylaşılan bellek kullanır, ancak diğer UNIX platformlarından farklı olarak AIX , 32 bitlik işlemlerin yalnızca 11 paylaşılan bellek kümesi bağlanmasına izin verir.

32 bitlik işlemlerin çoğu bu sınırla karşılaşmaz, ancak yüksek bellek gereksinimi olan uygulamalar IBM MQ ile bağlantı kuramayabilir. Neden kodu 2102: MQRC_RESOURCE_PROBLEM. Aşağıdaki uygulama tipleri bu hatayı görebilir:

- 32 bit Java sanal makinelerinde çalışan programlar

- Büyük ya da çok büyük bellek modellerini kullanan programlar
- Birçok kuyruk yöneticilerine ya da veritabanına bağlanan programlar
- Paylaşılan bellek kümelerine kendi başlarına bağlanan programlar

AIX , 32 bit işlemler için daha fazla paylaşılan bellek bağlamalarına olanak sağlayan genişletilmiş bir paylaşılan bellek özelliği sunar. Bir uygulamayı bu özellikte çalıştırmak için, kuyruk yöneticilerinizi ve programınızı başlatmadan önce EXTSHM=ON ortam değişkenini dışa aktarın. EXTSHM=ON özelliği çoğu durumda bu hatayı önler, ancak shmctl işlevinin SHM_SIZE seçeneğini kullanan programlarla uyumsuz.

IBM MQ MQI client uygulamaları ve tüm 64 bitlik süreçler bu sınırlamadan etkilenmez. EXTSHM ' nin ayarlanıp ayarlanmadığına bakılmaksızın IBM MQ kuyruk yöneticilerine bağlanabilirler.

Linux

AIX

IBM MQ ve UNIX System V IPC kaynakları

Kuyruk yöneticisi bazı IPC kaynaklarını kullanır. Hangi kaynakların kullanıldığını öğrenmek için **ipcs -a** kullanın.

Bu bilgiler, yalnızca AIX and Linux sistemlerinde çalışan IBM MQ için geçerlidir.

IBM MQ , System V işlemler arası iletişim (IPC) kaynaklarını kullanır (*semaforlar* ve *paylaşılan bellek kesimleri*) sistem bileşenleri arasında veri depolamak ve aktarmak için. Bu kaynaklar, kuyruk yöneticisine bağlanan kuyruk yöneticisi işlemleri ve uygulamaları tarafından kullanılır. IBM MQ MQI clients , IBM MQ izleme denetimi dışında IPC kaynaklarını kullanmayın. Makinede kullanılmakta olan IPC kaynaklarının sayısı ve boyutu hakkında tam bilgi almak için UNIX komutu **ipcs -a** kullanın.

Linux

AIX

IBM MQ ve UNIX Süreç Önceliği

Süreç önceliği *nice* değerleri ayarlanırken iyi uygulamalar.

Bu bilgiler, yalnızca AIX and Linux sistemlerinde çalışan IBM MQ için geçerlidir.

Bir işlemi arka planda çalıştırırsanız, bu işleme çağırın kabuk tarafından daha yüksek bir *hoş* değer (ve dolayısıyla daha düşük öncelik) verilebilir. Bu, genel IBM MQ performans çıkarımları olabilir. Yüksek stresli durumlarda, daha yüksek öncelikte ve daha düşük öncelikte çalıştırılmaya hazır birçok iş parçacığı varsa, işletim sistemi zamanlama özellikleri, düşük öncelikli iş parçacıklarını işlemci zamanından mahrum edebilir.

runmq1s gibi kuyruk yöneticileriyle ilişkili bağımsız olarak başlatılan süreçlerin, ilişkilendirildikleri kuyruk yöneticisiyle aynı *güzel* değerlere sahip olması iyi bir uygulamadır. Kabuğun bu arka plan işlemlerine daha yüksek bir *nice* değeri atamadığından emin olun. Örneğin, ksh 'de, ksh 'nin arka plan işlemlerinin *nice* değerini yükseltmesini durdurmak için "set +o bgnice" ayarını kullanın. "ps -efl" listesinin *NI* kolonunu inceleyerek, çalışan işlemlerin *güzel* değerlerini doğrulayabilirsiniz.

Ayrıca, IBM MQ uygulama işlemlerini kuyruk yöneticisiyle aynı *güzel* değerle başlatın. Farklı *güzel* değerlerle çalışırlarsa, bir uygulama iş parçacığı bir kuyruk yöneticisi iş parçacığını engelleyebilir ya da tam tersi, performansın düşmesine neden olabilir.

z/OS

z/OS üzerinde IBM MQ ortamının planlanması

IBM MQ ortamınızı planlarken, veri kümeleri, sayfa kümeleri, Db2, Coupling Facilities ve günlüğe kaydetme ve yedekleme olanaklarına ilişkin kaynak gereksinimlerini göz önünde bulundurmanız gerekir. IBM MQ ' in çalıştığı ortamı planlamak için bu konuyu kullanın.

IBM MQ mimarinizi planlamadan önce, temel IBM MQ for z/OS kavramlarını tanıyın ve [IBM MQ for z/OS kavramları](#) içindeki konulara bakın.

Kuyruk yöneticinizi planlarken, kuruluşunuzdaki farklı kişilerle çalışmanız gerekebilir. Değişiklik kontrol prosedürleri uzun sürebileceği için bu insanları erken dahil etmek genellikle iyi bir fikirdir. Ayrıca, IBM MQ for z/OS' yi yapılandırmanız için gereken parametreleri de size söyleyebilirler.

Örneğin, aşağıdakilerle çalışmanız gerekebilir:

- Depolama yöneticisi, kuyruk yöneticisi veri kümelerinin üst düzey niteleyicisini belirler ve kuyruk yöneticisi veri kümeleri için yeterli alan ayırır.
- z/OS sistem programcısı, IBM MQ altsistemini z/OS ve APF ' yi tanımlamak için IBM MQ for z/OS kitaplıklarını yetkilendirir.
- IBM MQ for z/OS için hangi TCP/IP yığınının ve kapılarının kullanılması gerektiğini belirlemek için ağ yöneticisi.
- Kuyruk yöneticisi veri kümelerine, IBM MQ for z/OS kaynakları için güvenlik profillerine ve TLS sertifikalarına erişim ayarlamak için güvenlik yöneticisi.
- Db2 yöneticisi, bir kuyruk paylaşım grubunu yapılandırırken Db2 çizelgelerini ayarlayabilir.

İlgili kavramlar

[IBM MQ Teknik genel bakış](#)

İlgili görevler

[“IBM MQ mimarisinin planlanması” sayfa 5](#)

IBM MQ ortamınızı planlarken, IBM MQ ' un tek ve birden çok kuyruk yöneticisi mimarisi ve noktadan noktaya iletişim ve yayınlama/abone olma ileti sistemi stilleri için sağladığı desteği göz önünde bulundurun. Ayrıca, kaynak gereksinimlerinizi ve günlük kaydı ve yedekleme olanaklarını kullanımınızı da planlayın.

[yapılandırmaz/OS](#)

[YönetmeIBM MQ for z/OS](#)

► z/OS

Kuyruk yöneticinizin planlanması

Bir kuyruk yöneticisi ayarlarken, kuyruk yöneticisinin işletmenizin gereksinimlerini karşılaması için planlamanız kuyruk yöneticisinin büyümesine izin vermelidir.

Bir kuyruk yöneticisini yapılandırmanın en iyi yolu aşağıdaki adımlarda bulunur:

1. Temel kuyruk yöneticisini yapılandır
2. Kuyruk yöneticisini kuyruk yöneticisi iletişimini ve uzak istemci uygulaması iletişimini yapan kanal başlatıcısını yapılandır
3. İletileri şifrelemek ve korumak istiyorsanız, [Advanced Message Security](#) ürününü yapılandırın
4. IBM MQ üzerinden Dosya Aktarma olanağını kullanmak istiyorsanız, [z/OS için Managed File Transfer](#) dosyasını yapılandırın.
5. Yönetim ya da ileti sistemini REST API ya da IBM MQ ' yi bir web tarayıcısından yönetmek için IBM MQ Console kullanmak istiyorsanız, [mqweb](#) sunucusunu yapılandırın.

Bazı işletmeler, kendi ortamlarında yüz binlerce kuyruk yöneticilerine sahip. Şimdi ve beş yıl içinde IBM MQ ağınızı göz önünde bulundurmanız gerekir.

z/OS üzerinde, bazı kuyruk yöneticileri saniyede binlerce iletiyi işlerler ve saniyede 100 MB ' yi aşarak oturum açerler. Çok yüksek birimler bekliyorsanız, birden çok kuyruk yöneticisine sahip olmayı göz önünde bulundurmanız gerekebilir.

z/OS işletim sisteminde IBM MQ , iletilerin Bağlama Olanığı 'nda saklandığı bir kuyruk paylaşım grubunun (QSG) bir parçası olarak çalışabilir ve kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisi iletilere erişebilir. Bir kuyruk paylaşım grubunda çalışmak istiyorsanız, kaç kuyruk yöneticilerine gereksinim duyacağınızı göz önünde bulundurmanız gerekir. Genellikle, her LPAR için bir kuyruk yöneticisi vardır. CF yapılarını düzenli olarak yedeklemek için bir kuyruk yöneticiniz de olabilir.

Yeni bir kuyruk tanımlama gibi, yapılandırmada yapılan bazı değişikliklerin yapılması kolaydır. Günlüklerin ve sayfa kümelerinin daha büyük olması gibi bazıları daha zordur ve kuyruk yöneticisinin adı ya da kuyruk paylaşım grubu adı gibi bazı yapılandırmada değişiklik yapılamaz.

[MP16 performance SupportPac](#) içinde performans ve ayarlama bilgileri vardır.

Adlandırma kuralları

Kuyruk yöneticisi veri kümeleri için bir adlandırma kuralınız olmalıdır.

Birçok kuruluş, yükleme kitaplıkları adında yayın numarasını kullanır ve bu şekilde devam eder. Şu anda kullanımda olan sürümü (örneğin, MQM.V930.SCSQAUTH) gösteren bir MQM.SCSQAUTH diğer adına sahip olmayı düşünebilirsiniz; bu nedenle, yeni bir IBM MQsürümüne geçiş yaparken CICS, Batch ve IMS JCL ' yi değiştirmeniz gerekmez.

Kullanılmakta olan IBM MQ sürümüne ilişkin kuruluş dizinine başvurmak için z/OS UNIX System Services içindeki bir simgesel bağlantıyı kullanabilirsiniz.

Kuyruk yöneticisi tarafından kullanılan veri kümeleri (günlükler, sayfa kümeleri, JCL kitaplıkları), güvenlik profillerinin oluşturulmasını basitleştirmek için bir adlandırma kuralı ve veri kümelerinin, veri kümelerinin diske yerleştirildiği yeri ve sahip oldukları öznitelikleri denetleyen SMS depolama sınıflarıyla eşlenmesi gerekir.

IBM MQ sürümünün sayfa kümelerinin ya da günlüklerinin adına yerleştirilmesinin iyi bir fikir olmadığını unutmayın. Bir gün yeni bir sürüme geçebilirsiniz ve veri kümesi "yanlış" adlara sahip olur.

Uygulamalar

IBM MQuygulamasını yapılandırmanın en iyi yolunu ve iş uygulamalarını anlamanız gerekir. Örneğin, uygulamaların kurtarma ve yineleme yeteneği sağlama mantığı varsa, kalıcı olmayan iletiler yeterli olabilir. IBM MQ ' in kurtarmayı işlemlerini istiyorsanız, kalıcı iletileri kullanmanız ve iletileri eşitleme noktasına yerleştirmeniz ve almanız gerekir.

Kuyrukları farklı iş hareketlerinden yalıtmanız gerekir. Bir iş uygulamasına ilişkin kuyruk dolduruluyorsa, bunun diğer iş uygulamalarını etkilemesini istemezsiniz. Mümkünse, farklı sayfa kümelerindeki ve arabellek havuzlarındaki ya da yapılarındaki kuyrukları yalıtın.

İletilerin profilini anlamanız gerekir. Birçok uygulama için kuyrukların yalnızca birkaç ileti vardır. Diğer uygulamalarda gün boyunca kuyruklar oluşturulabilir ve bir gecede işlenebilir. Normalde üzerinde yalnızca birkaç ileti bulunan bir kuyruğun, bir sorun varsa ve iletiler işlenmezse, saatlerce ileti tutması gerekebilir. Beklenen en yüksek kapasiteye izin vermek için CF yapılarını ve sayfa kümelerini boyutlandırmanız gerekir.

Yapılandırma sonrası

Kuyruk yöneticinizi (ve bileşenlerini) yapılandırdıktan sonra aşağıdakileri planlamanız gerekir:

- Sayfa kümeleri yedekleniyor.
- Nesnelerin tanımlamaları yedekleniyor.
- CF yapılarının yedeklenmesi otomatikleştiriliyor.
- IBM MQ iletilerinin izlenmesi ve bir sorun saptandığında işlem yapılması.
- IBM MQ istatistik verileri toplanıyor.
- Sanal depolama gibi kaynak kullanımının ve saat başına günlüğe kaydedilen veri miktarının izlenmesi. Bununla, kaynak kullanımında artış olup olmadığını ve yeni bir kuyruk yöneticisi ayarlama gibi işlemler gerçekleştirmeniz gerekir gerekolmadığını görebilirsiniz.

z/OS üzerinde depolama ve performans gereksinimlerinizin planlanması

IBM MQ sisteminiz için gerçekçi ve ulaşılabilir depolama ve performans hedefleri belirlemelisiniz. Depolamayı ve performansını etkileyen etkenleri anlamanıza yardımcı olmak için bu konuyu kullanın.

Bu konuda, IBM MQ for z/OSiçin depolama ve performans gereksinimlerine ilişkin bilgiler yer alır. Aşağıdaki bölümleri içerir:

- [z/OS IBM MQiçin performans seçenekleri](#)

- [z/OS İş yükü yönetimi öneminin ve hız hedeflerinin belirlenmesi](#)
- [“Kitaplık depolama” sayfa 143](#)
- [“Sistem LX kullanımı” sayfa 143](#)
- [“Depolama yapılandırması” sayfa 144](#)
- [“Disk depolaması” sayfa 149](#)

Daha fazla bilgi için bkz. [“Depolama ve performans gereksinimleri hakkında daha fazla bilgi nereden edinebilir?” sayfa 150](#) .

z/OS performans seçenekleri IBM MQ

İş yükü yönetimiyle, performans hedeflerini tanımlar ve her hedefe bir iş önemi atarsınız. İşe ilişkin hedefleri iş açısından tanımlarsınız ve sistem, amacına ulaşmak için işe ne kadar kaynak (işlemci ve depolama gibi) verilmesi gerektiğine karar verir. İş yükü yönetimi, sağladığınız hedeflere dayalı olarak dağıtma önceliğini denetler. İş yükü yönetimi, belirtilen hedefe ulaşmak için gerektiği şekilde önceliği yükseltir ya da düşürür. Bu nedenle, sistemdeki her iş parçasının önceliklerini tam olarak ayarlamanıza gerek yoktur ve bunun yerine iş hedeflerine odaklanabilirsiniz.

Üç tür hedef vardır:

Yanıt süresi

İşin ne kadar hızlı işlenmesini istiyorsunuz

Yürütme hızı

İşlemci, depolama, G/Ç erişimi ve kuyruk gecikmesi için gecikmeden işin hazır olduğunda ne kadar hızlı çalışması gerektiği

kullanıcı tarafından oluşturulan

Performans hedefleri olmayan düşük öncelikli iş kategorisi

Yanıt süresi hedefleri, son kullanıcı uygulamaları için uygundur. Örneğin, CICS kullanıcıları iş yükü hedeflerini yanıt süresi hedefleri olarak belirleyebilir. IBM MQ adres alanları için hız hedefleri daha uygundur. Kuyruk yöneticisinde yapılan işlerin küçük bir miktarı bu hız hedefine doğru sayılır, ancak bu iş performans açısından kritik önem taşır. Kuyruk yöneticisi tarafından yapılan işlerin çoğu, son kullanıcı uygulamasının performans hedefine doğru sayılır. Kanal başlatıcısı adres alanı tarafından yapılan işlerin çoğu kendi hız hedefine doğru sayılır. Kanal başlatıcısının başardığı IBM MQ iletilerinin alınması ve gönderilmesi, genellikle bunları kullanan iş uygulamalarının başarımı için önemlidir.

z/OS iş yükü yönetimi önem derecesini ve hız hedeflerini belirleme

Ek bilgi için bkz. [“z/OS iş yükü yönetimi öneminin belirlenmesi” sayfa 144](#) .

Kitaplık depolama

Ürün kitaplıkları için disk saklama alanı ayırmanız gerekir. Tam şekiller yapılandırmanıza bağlıdır ve SMP/E kitaplıklarının yanı sıra hedef ve dağıtım kitaplıklarını da içermelidir.

IBM MQ for z/OS tarafından kullanılan hedef kitaplıklar PDSE biçimlerini kullanır. Herhangi bir PDSE hedef kitaplıklarının bir sistem şebekesi dışında paylaşılmadığından emin olun. Gerekli kitaplıklar ve bunların boyutları ve gerekli biçimi hakkında daha fazla bilgi için Program Dizinine bakın. Program Dizinlerine ilişkin yükleme bağlantıları için bkz. [IBM MQ for z/OS Program Dizini PDF dosyaları](#).

Sistem LX kullanımı

Tanımlanan her IBM MQ altsistemi, IPL sırasında bir sistem bağ dizinini (LX) ve kuyruk yöneticisi başlatıldığında sistem dışı bağ dizinlerini ayırır. Kuyruk yöneticisi durdurulup yeniden başlatıldığında sistem bağ dizini yeniden kullanılır. Benzer şekilde, dağıtılmış kuyruğa alma, sistem dışı bir bağ dizinini ayırıyor. z/OS sisteminizde yeterli sistem LX 'i tanımlanmamış olması durumunda, bu ayrılmış sistem LX 'leri göz önünde bulundurmanız gerekebilir.

Gerekirse, SYS1.PARMLIB üyesi IEASYSxx içinde NSYSLX parametresi ayarlanarak sistem LX sayısı artırılabilir.

z/OS **z/OS iş yükü yönetimi öneminin belirlenmesi**

Hizmet tanımı aracılığıyla iş yükü yönetimi ve hedeflerin tanımlanması hakkında tam bilgi için bkz.z/OS ürün belgeleri.

Bu konuda, z/OS iş yükü yönetimi öneminin ve hız hedeflerinin sisteminizdeki diğer önemli çalışmalara göre nasıl ayarlandığı açıklanmaktadır. Daha fazla bilgi için bkz. [z/OS MVS Planning: Workload Management](#) .

Kuyruk yöneticisi adres alanının, altsistem hizmetleri sağladığı için yüksek öncelikli olarak tanımlanması gerekir. Kanal başlatıcı bir uygulama adresi alanıdır, ancak uzak kuyruk yöneticisine gönderilen iletilerin gecikmemesini sağlamak için genellikle yüksek öncelik verilir. Advanced Message Security (AMS) ayrıca altsistem hizmetleri sağlar ve yüksek öncelikli olarak tanımlanması gerekir.

Aşağıdaki hizmet sınıflarını kullanın:

Varsayılan SYSSTC hizmet sınıfı

- VTAM ve TCP/IP adres alanları
- IRLM adres alanı (IRLMPROC)

Not: VTAM, TCP/IP ve IRLM adres alanları, tüm DBMS adres alanlarından, bağlı adres alanlarından ve bunların alt adres alanlarından daha yüksek bir dağıtma önceliğine sahip olmalıdır. İş yükü yönetiminin VTAM, TCP/IP ya da IRLM ' nin önceliğini diğer DBMS adres alanlarındaki önceliğe (ya da daha altına) düşürmesine izin vermeyin

PRODREGN gibi tanımladığınız bir ada sahip bir hizmet sınıfı için yüksek hızlı hedef ve önem düzeyi:

- IBM MQ kuyruk yöneticisi, kanal başlatıcısı ve AMS adres alanları
- Db2 (Db2tarafından oluşturulan saklanmış yordamlar adres alanı dışında tüm adres alanları)
- CICS (tüm bölge tipleri)
- IMS (BMP dışındaki tüm bölge tipleri)

Yüksek hızlı bir hedef, tüm bu adres alanları için girişimler ve yeniden başlatmaların mümkün olan en kısa sürede gerçekleştirilmesini sağlamak için iyi bir hedeftir.

CICS ve IMS bölgelerine ilişkin hız hedefleri yalnızca başlatma ya da yeniden başlatma sırasında önemlidir. İşlemler çalışmaya başladıktan sonra, iş yükü yönetimi CICS ya da IMS hız hedeflerini yoksayar ve bölgelerde çalışan işlemlerin yanıt süresi hedeflerine göre öncelikleri atar. Bu işlem hedefleri, uyguladıkları iş uygulamalarının görece önceliğini yansıtmalıdır. Genellikle 2 önem değerine sahip olabilirler. IBM MQ kullanan her toplu iş uygulaması, uyguladıkları iş uygulamalarının görece önceliğini yansıtan hız hedeflerine ve önemine sahip olmalıdır. Genellikle önem ve hız hedefleri PRODREGN için olandan daha az olacaktır.

z/OS **Depolama yapılandırması**

64 bit adres alanında, 2GB adresini işaretleyen "çubuk" adlı sanal bir hat vardır. Çubuk, 2GB adresinin altındaki depolamayı "çubuğun altında"olarak adlandırılan depolamadan 2GB adresinin üstündeki depolamadan ayırır; bu depolama alanı, çubuğun üstünde "olarak adlandırılır". Çubuğun altındaki depolama 31 bit adreslenebilirlik özelliğini kullanır, çubuğun üstündeki depolama 64 bit adreslenebilirlik özelliğini kullanır.

JCL REGION parametresini kullanarak 31 bit saklama alanı sınırını ve MEMLIMIT parametresini kullanarak 64 bit saklama alanı sınırını belirtebilirsiniz. Belirtilen bu değerler z/OS çıkışları tarafından geçersiz kılınabilir.

Önerilen depolama yapılandırması

Aşağıdaki çizelge, kuyruk yöneticisi, kanal başlatıcısı ve AMS adres alanları için önerilen **REGION** ve **MEMLIMIT** değerlerini göstermektedir. Bu öneriler başlangıç noktası olarak kullanılmalı ve aşağıdaki bilgiler kullanılarak ayarlanmalıdır:

- “Kuyruk yöneticisi depolama yapılandırması” sayfa 145
- “Kanal başlatıcı depolama yapılandırması: IBM MQ 9.3” sayfa 147
- **V9.3.1** “IBM MQ 9.3.1 ' den kanal başlatıcı depolama yapılandırması” sayfa 148

Çizelge 19. REGION ve MEMLIMIT için önerilen tanımlar	
Adres alanı	Depolama yapılandırması
Kuyruk yöneticisi	REGION=0M, MEMLIMIT=3G
Kanal başlatıcı: IBM MQ 9.3.0	REGION=0M, MEMLIMIT=256M
V9.3.1 IBM MQ 9.3.1 ' den kanal başlatıcı	REGION=0M, MEMLIMIT=2G
AMS adres alanı	REGION=0M

MEMLIMIT ve REGION boyutunun yönetilmesi

Diğer düzenekler (örneğin, SYS1.PARMLIB ' un SMFPRMxx üyesindeki **MEMLIMIT** parametresi ya da IEFUSI çıkışı, z/OS adres alanları için çubuğun üzerinde varsayılan bir sanal saklama alanı miktarı sağlamak üzere kuruluşunuzda kullanılabilir. Çubuğun üzerindeki depolamayı sınırlamaya ilişkin tam ayrıntılar için bkz. [Çubuğun üstündeki bellek yönetimi](#) .

z/OS Kuyruk yöneticisi depolama yapılandırması

Kuyruk yöneticisi adres alanı, bir IBM MQ kuruluşunda 64 bit saklama alanının ana kullanıcısı olabilir. Kuyruk yöneticisine yönelik her bağlantı, aşağıdaki metinde açıklandığı gibi ortak saklama alanının ayrılmasını gerektirir. 64 bit saklama alanına ek olarak, kuyruk yöneticisinin JCL kuyruk yöneticisinde REGION=0M belirterek kullanılabilir tüm 31 bit saklama alanını kullanmasına izin vermeniz gerekir.

Ortak depolama

Her bir IBM MQ for z/OS altsistemi, aşağıdaki yaklaşık depolama gereksinimlerine sahiptir:

- CSA 4KB
- ECSA 800KBve CSQ6SYSP sistem değiştirgesi makrosunun **TRACTBL** değiştirgesinde belirtilen izleme çizelgesinin büyüklüğü. Ek bilgi için [CSQ6SYSP' nin Kullanılması](#) başlıklı konuya bakın.

Ayrıca, kuyruk yöneticisine her eşzamanlı mantıksal bağlantı için yaklaşık 5 KB ECSA gerekir. Bir görev sona erdiğinde, diğer IBM MQ görevleri bu depolamayı yeniden kullanabilir.

IBM MQ , kuyruk yöneticisi kapatılıncaya kadar saklama alanını serbest bırakmaz; bu nedenle, koşut zamanlı bağlantı sayısı üst sınırını 5KBile çarparak gereken ECSA miktarı üst sınırını hesaplayabilirsiniz. Koşut zamanlı mantıksal bağlantı sayısı, aşağıdaki sayıların toplamıdır:

- IBM MQ' e bağlı olan, ancak bağlantısı kesilmiş olmayan Toplu İş, TSO, z/OS UNIX System Services, IMSve Db2 saklanmış yordam adres alanı (SPAS) bölgelerindeki görevler (TCB).
- Bir IBM MQ isteği yayınlayan, ancak sonlandırılmamış CICS hareketleri
- JMS Bağlantılar, Oturumlar, TopicSessions ya da QueueSessions (kuyruk oturumları bağlantısı için) yaratıldı, ancak henüz yok edilmedi ya da atık toplanmadı.
- Etkin IBM MQ kanalları

ACELIM yapılanış deęiřtirgesiyle, kuyruk yöneticisine mantıksal baęlantılar tarafından kullanılan ortak saklama alanı için bir sınır ayarlayabilirsiniz. **ACELIM** denetimi, öncelikle Db2 saklanmış yordamlarının IBM MQ kuyruklarında işlemlere neden olduęu siteleri ilgilendiriyor.

Saklanmış bir yordamdan yönlendirildiğinde, her IBM MQ işlemi kuyruk yöneticisine yeni bir mantıksal baęlantıyla sonuçlanabilir. Büyük Db2 iş birimleri (örneğin, tablo yükü nedeniyle), ortak depolama için aşırı taleple sonuçlanabilir.

ACELIM, sistemdeki baęlantı sayısını sınırlayarak ortak depolama kullanımını sınırlamak ve z/OS sistemini korumak için tasarlanmıştır. **ACELIM** deęerini yalnızca, aşırı miktarda ECSA saklama alanı kullandığı belirtilen kuyruk yöneticilerine ayarlamamız gerekir. Ek bilgi için *CSQ6SYSP* 'nin Kullanılması başlıklı konudaki **ACELIM** bölümüne bakın.

ACELIM için bir deęer ayarlamak üzere öncelikle **ACELIM** deęeri tarafından denetlenen alt havuzda bulunan depolama miktarını belirleyin. Bu bilgiler, istatistik CLASS (3) izlemesi tarafından üretilen SMF 115 alt tip 5 kayıtlarında yer alır.

IBM MQ SMF verileri SupportPac MP1B kullanılarak biçimlendirilebilir. **ACELIM** tarafından denetlenen alt havuzda kullanılan bayt sayısı, *ACE/PEB* başlıklı satırda STGPOOL DD 'de görüntülenir.

SMF 115 istatistik kayıtlarıyla ilgili ek bilgi için IBM MQ for z/OS performans istatistiklerini yorumlamabařlıklı konuya bakın.

Büyüme ve iş yükü artışları için alan sağlamak üzere normal deęeri yeterli bir marjla artırın. **ACELIM** yapılandırmasında kullanılmak üzere KB cinsinden depolama boyutu üst sınırını elde etmek için yeni deęeri 1024 'e bölün.

Özel depolama

Kuyruk yöneticisi adres alanı, birçok iç denetim bloęu için 64 bitlik depolama alanı kullanır. Kuyruk yöneticisi JCL 'nin **MEMLIMIT** parametresi, kullanılabilir 64 bit saklama alanı miktarı üst sınırını tanımlar. 3GB depolama alanı, **MEMLIMIT=3G**, kullanmanız gereken en düşük depolama alanıdır, ancak yapılandırmanıza baęlı olarak çok daha fazla depolama gerekebilir.

Olası sorunları önlemek için **MEMLIMIT=NOLIMIT** yerine belirli bir **MEMLIMIT** deęeri belirtmelisiniz. **NOLIMIT** deęerini ya da çok büyük bir deęeri belirlerseniz, sisteminizde sayfalanmaya neden olan kullanılabilir z/OS sanal saklama alanının tümünü kullanma potansiyeli vardır. **MEMLIMIT** deęerini artırırken, z/OS sistem programcınızla yeni ayarı tartışmanız gerekir. Bunun için, sistem genelinde kullanılabilir depolama alanı miktarına iliřkin bir sınırlama olması gerekir.

MEMLIMIT için büyük bir deęeriniz varsa, bir dökümde daha fazla veri yakalandıkça döküm veri kümelerinizin boyutunu artırmanız gerekebilir.

Adres alanı depolama kullanımını, kullanılmakta olan 31 ve 64 bitlik özel saklama alanı miktarını ve kalan boş miktarı gösteren CSQY220I iletisinden izleyebilirsiniz.

Arabellek Havuzları

Arabellek havuzları, kuyruk yöneticisi adres alanındaki özel saklama alanının önemli bir kullanıcıdır. Her arabellek havuzu büyüklüęü kuyruk yöneticisi kullanıma hazırlama sırasında belirlenir ve bu arabellek havuzunu kullanan bir sayfa kümesi baęlandığında arabellek havuzu için saklama alanı ayrılır. **LOCATION (ABOVE|BELOW)** deęiřtirgesi, arabelleklerin nereye ayrıldığını belirtmek için kullanılır. Arabellek havuzlarının büyüklüęünü devingen olarak deęiřtirmek için ALTER BUFFPOOL komutunu kullanabilirsiniz.

MEMLIMIT için bir deęer hesaplarırken, **LOCATION (ABOVE)** ile yapılandırıldıysa arabellek havuzu boyutlarını göz önünde bulundurmanız önemlidir. Hesaplamayı ařaęıdaki gibi gerçekleřtirmeniz gerekir.

MEMLIMIT deęerini 2GB olarak ve **LOCATION (ABOVE)** ile yapılandırılmış arabellek havuzlarının boyutunu, en yakın GB 'ye yuvarlanmış olarak hesaplayın. **MEMLIMIT** deęerini en az 3GB olarak ayarlayın ve arabellek havuzlarınızın boyutunu artırmanız gerektiğinde bunu gereken řekilde artırın.

Örneğin, **LOCATION (ABOVE)** ile yapılandırılmış üç arabellek havuzu için, bir arabellek havuzunun 10.000 arabelleği ve iki ve üç arabellek havuzunun her birinin 50.000 arabelleği vardır. Çubuğun üzerindeki bellek kullanımı 110.000 'e eşittir (toplam arabellek sayısı) * 4096 = 450,560,000 bayt = 430MB.

LOCATION ' den bağımsız olarak tüm arabellek havuzları, denetim yapıları için 64 bit saklama alanından yararlanır. Bu havuzlardaki arabellek havuzlarının sayısı ve arabellek sayısı arttıkça bu önemli hale gelebilir. Her arabellek için yaklaşık 200 baytlık 64 bitlik depolama alanı gerekir. Gereken önceki yapılandırma için: 200 * 110,000 = 22.000.000 bayt = 21MB.

Bu nedenle, bu senaryoda 3GB , büyüme kapsamına izin veren **MEMLIMIT** için kullanılabilir: 3GB' ye yuvarlanan 21MB + 430MB + 2GB .

Bazı yapılandırmalarda, arabellekleri gerçek saklama alanı tarafından kalıcı olarak desteklenen arabellek havuzlarını kullanmanın önemli başarımı olabilir. Bunu, arabellek havuzunun **PAGECLAS** özneliği için **FIXED4KB** değerini belirterek yapabilirsiniz. Ancak, bunu yalnızca LPAR ' da yeterli gerçek saklama alanı varsa yapmalısınız; tersi durumda, diğer adres alanları etkilenebilir. **PAGECLAS** için **FIXED4KB** değerini ne zaman kullanmanız gerektiğine ilişkin bilgi için bkz. IBM MQ Support Pac [MP16: IBM MQ for z/OS -Capacity planning & tuning](#).

Arabellek havuzlarının MVS sayfalaması olacak kadar büyük olması başarımı olumsuz etkileyebilir. IBM MQ ' in iletiyi sayfa kümesine/sayfa kümesinden taşınmasıyla sayfa oluşturmayan daha küçük bir arabellek havuzu kullanmayı düşünebilirsiniz.

V 9.3.1 CFSTRUCT 'U KURTARIN

IBM MQ 9.3.1 ' den **RECOVER CFSTRUCT** komutu, 64 bitlik depolamayı daha iyi kullanır. Çoğu durumda kullanılabilir yedek 64 bit depolama alanı olması gerekir; bu nedenle, komutun kullanılması **MEMLIMIT** değerinin artırılmasını gerektirmez. Ancak, birkaç milyondan fazla ileti içeren büyük yapı yedekleriniz olması muhtemelse, **RECOVER CFSTRUCT** komutunu 500MB ile işleyebilecek tüm kuyruk yöneticileri için **MEMLIMIT** artırmanız gerekir.

Örneğin, **MEMLIMIT=3G** önceden varsa, **MEMLIMIT** parametresi ondalık noktalara izin vermediği için **MEMLIMIT=4G** parametresini kullanmayı düşünmelisiniz.

Paylaşılan İleti Veri Kümesi (SMDS) arabellekleri ve MEMLIMIT

İleti alışverişi iş yüklerini paylaşılan ileti veri kümelerini kullanarak çalıştırırken, **DSBUFS** ve **DSBLOCK** özneliklerini ayarlayarak elde edilebilecek iki optimizasyon düzeyi vardır.

SMDS arabelleği tarafından kullanılan çubuk kuyruk yöneticisi saklama alanı miktarı **DSBUFS x DSBLOCK** ' tur. Bu, kuyruk yöneticisindeki her **CFLEVEL (5)** yapısı için varsayılan olarak 100 x 256KB (25MB) kullanıldığı anlamına gelir.

Bu değer çok yüksek olmasa da, işletmenizin ya da işletmenizin çok sayıda **CFSTRUCT** ' si varsa, bunların bazıları arabellek havuzları için **MEMLIMIT** yüksek değeri ayırabilir ve bazen derin dizinli kuyrukları olabilir, bu nedenle toplam olarak çubuğun üzerinde depolama alanı tükenebilir.

z/OS Kanal başlatıcı depolama yapılandırması: IBM MQ 9.3

Kanal başlatıcı genellikle kuyruk yöneticisinden çok daha az 64 bit depolama kullanır. 64 bit saklama alanına ek olarak, kuyruk yöneticisi **JCL** ' de **REGION=0M** belirterek kanal başlatıcısının kullanılabilir tüm 31 bit saklama alanını kullanmasına izin vermeniz gerekir.

Ortak depolama

Kanal başlatıcı genellikle **ECSA** ' nın en çok 160KB kullanılması gerektirir.

31 bitlik özel depolama

Kanal başlatıcısının kullanabileceği 31 bitlik saklama alanı, **CHINIT** ' nin sahip olabileceği eşzamanlı bağlantı sayısını sınırlar.

Her kanal, kanal başlatıcı adres alanında yaklaşık 170KB genişletilmiş özel bölge kullanır. İleti kanalları (örneğin, gönderen ya da alıcı kanalları) için, 32KB ' den büyük iletiler iletilildiğinde ileti boyutuna göre depolama alanı artar. Bu artırılmış depolama alanı aşağıdaki durumda serbest bırakılır:

- Bir gönderen ya da istemci kanalı, ardışık 10 ileti için geçerli arabellek büyüklüğünün yarısından daha azını gerektirir.
- Bir sağlıklı işletim bildirimi gönderilir ya da alınır.

Depolama alanı Dil Ortamı içinde yeniden kullanılmak üzere serbest bırakılır, ancak depolama alanı z/OS sanal depolama yöneticisi tarafından boş olarak görülmez. Bu, kanal sayısının üst sınırının ileti boyutu ve geliş kalıplarına ve genişletilmiş özel bölge boyutundaki tek tek kullanıcı sistemlerinin sınırlamalarına bağlı olduğu anlamına gelir.

Genişletilmiş bölge boyutunun 1.6GB' yi aşma olasılığı düşük olduğundan, kanal sayısının üst sınırı birçok sistemde yaklaşık 9000 olabilir. 32KB ' den büyük ileti boyutlarının kullanılması, sistemdeki kanal sayısı üst sınırını azaltır. Örneğin, 100MB uzunluklu iletiler iletilir ve 1.6GB genişletilmiş bölge boyutu varsayılır, kanal sayısı üst sınırı 15 'tir.

Kanal başlatıcı izlemesi bir veri alanına yazılır. Veri alanı depolamanın boyutu, **TRAXTBL** parametresiyle denetlenir. Bkz. [ALTER QMGR](#).

64 bit özel depolama

Kanal başlatıcısı JCL ' nin MEMLIMIT parametresi, kullanılabilir 64 bitlik saklama alanı miktarı üst sınırını tanımlar. 256MB depolama alanı, MEMLIMIT=256M, kullanmanız gereken değer alt sınırdır. Yapılandırmanıza bağlı olarak önemli ölçüde daha fazla gereksinim olabilir.


Olası sorunları önlemek için, MEMLIMIT = NOLIMIT yerine duyarlı bir MEMLIMIT değeri belirtmelisiniz. NOLIMIT ya da çok büyük bir değer belirlerseniz, sisteminizde sayfalanmaya yol açan tüm kullanılabilir z/OS sanal saklama alanını kullanma potansiyeli vardır. MEMLIMIT değerini artırırken, z/OS sistem programcınızla yeni ayarı tartışmanız gerekir; bu durumda, sistem genelinde kullanılabilir depolama alanı miktarı sınırı vardır.

MEMLIMIT için büyük bir değerin varsa, bir dökümden daha fazla veri yakalandıkça döküm veri kümelerinizin boyutunu artırmanız gerekebilir.

Kanal başlatıcısında tek bir 64 bitlik depolama kullanıcısı vardır: SMF.

SMF

Etkinleştirilirse, SMF sınıf 4 muhasebesi ya da istatistikleri 64 bit depolama gerektirir. En az 256MB depolama alanı gerekir. Yeterli saklama alanı yoksa, kanal başlatıcısı [CSQX124E](#) iletisini yayınlar ve sınıf 4 muhasebe ve istatistikler kullanılamaz.

 *IBM MQ 9.3.1 ' den kanal başlatıcı depolama yapılandırması*

Kanal başlatıcı genellikle kuyruk yöneticisinden çok daha az 64 bit depolama kullanır. Ancak, IBM MQ 9.3.1 ' den kullanım arttı. 64 bit saklama alanına ek olarak, kuyruk yöneticisi JCL ' de REGION=0M belirterek kanal başlatıcısının kullanılabilir tüm 31 bit saklama alanını kullanmasına izin vermeniz gerekir.

Ortak depolama

Kanal başlatıcı genellikle ECSA ' nın en çok 160KB kullanılması gerektirir.

31 bitlik özel depolama

Kanal başlatıcısının kullanabileceği 31 bitlik saklama alanı, CHINIT ' nin sahip olabileceği eşzamanlı bağlantı sayısını sınırlar.

Her kanal, kanal başlatıcı adres alanında yaklaşık 170KB genişletilmiş özel bölge kullanır. İleti kanalları (örneğin, gönderen ya da alıcı kanalları) için, 32KB ' den büyük iletiler iletilildiğinde depolama alanı ileti boyutuna göre artar. Bu artırılmış depolama alanı aşağıdaki durumda serbest bırakılır:

- Bir gönderen ya da istemci kanalı, ardışık 10 ileti için geçerli arabellek büyüklüğünün yarısından daha azını gerektirir.
- Bir sağlıklı işletim bildirim gönderilir ya da alınır.

Depolama alanı Dil Ortamı içinde yeniden kullanılmak üzere serbest bırakılır, ancak depolama alanı z/OS sanal depolama yöneticisi tarafından boş olarak görülmez. Bu, kanal sayısının üst sınırının ileti boyutu ve geliş kalıplarına ve genişletilmiş özel bölge boyutundaki tek tek kullanıcı sistemlerinin sınırlamalarına bağlı olduğu anlamına gelir.

Genişletilmiş bölge boyutunun 1.6GB' yi aşma olasılığı düşük olduğundan, kanal sayısının üst sınırı birçok sistemde yaklaşık 9000 olabilir.

Kanal başlatıcı izlemesi bir veri alanına yazılır. Veri alanı depolamanın boyutu, **TRAXTBL** parametresiyle denetlenir. Bkz. [ALTER QMGR](#).

64 bit özel depolama

Kanal başlatıcısı JCL ' nin MEMLIMIT parametresi, kullanılabilir 64 bitlik saklama alanı miktarı üst sınırını tanımlar. 2 GB depolama alanı, MEMLIMIT=2 GB, kullanmanız gereken minimum değerdir. Yapılandırmanıza bağlı olarak önemli ölçüde daha fazla gereksinim olabilir.

Olası sorunları önlemek için, MEMLIMIT = NOLIMIT yerine duyarlı bir MEMLIMIT değeri belirtmelisiniz. NOLIMIT ya da çok büyük bir değer belirlerseniz, sisteminizde sayfalanmaya yol açan tüm kullanılabilir z/OS sanal saklama alanını kullanma potansiyeli vardır. MEMLIMIT değerini artırırken, z/OS sistem programcınızla yeni ayarı tartışmanız gerekir; bu durumda, sistem genelinde kullanılabilir depolama alanı miktarı sınırı vardır.

MEMLIMIT için büyük bir değerin varsa, bir dökümden daha fazla veri yakalandıkça döküm veri kümelerinizin boyutunu artırmamız gerekebilir.

Kanal başlatıcısında 64 bit depolama alanı olan iki kullanıcı vardır: SMF ve sunucu bağlantısı kanalları.

SMF

Etkinleştirilirse, SMF sınıf 4 muhasebesi ya da istatistikleri 64 bit depolama gerektirir. En az 256MB depolama alanı gerekir. Yeterli saklama alanı yoksa, kanal başlatıcısı [CSQX124E](#) iletisini yayınlar ve sınıf 4 muhasebe ve istatistikler kullanılamaz.

Sunucu bağlantısı kanalları

IBM MQ 9.3.1 sunucu bağlantısı kanallarından, 32 KB ' den büyük iletileri aktarıyorlarsa, ileti arabelleklerini 64 bitlik depolama alanına ayırır.

Kanallar, ardışık 10 ileti için geçerli arabellek boyutunun yarısından daha azını gerektiriyorsa ya da bir sağlıklı işletim bildirim gönderildiyse ya da alındıysa, bu arabellekler serbest bırakılır.

MEMLIMIT değeri, koşut zamanlı sunucu bağlantısı kanallarının çalışabileceği üst sınırı belirler. MEMLIMIT=2G alt sınır değerini kullanarak, IBM MQ 9.3.1' un önceki sürümlerinde olduğu gibi aynı sayıda kanalın çalışabilmesini sağlamanın yanı sıra, büyüme için de bir kapasite sağlamanız gerekir.

MEMLIMIT için yaklaşık bir değeri, eş zamanlı etkin sunucu bağlantısı kanallarının maksimum sayısını hesaplayarak ve bu kanallar için aktarmalarını beklediğiniz ileti boyutu üst sınırını hesaplayabilirsiniz. Başlangıç noktası olarak MEMLIMIT=2GB kullanılmalı ve yuvarlamalısınız.

Örneğin, eş zamanlı sunucu bağlantısı kanalı sayısı üst sınırını 2.000 olarak ayarlarsanız ve her kanalda 1MB ileti boyutu üst sınırı olacak şekilde ayarlanırsa, sunucu bağlantısı kanalları en fazla 2GB 64 bit depolama alanı kullanır. Bu, 2GB ' ye çok yakın olduğundan, MEMLIMIT=3G değerine yuvarlamanız gerekir.

Disk depolaması

Günlük veri kümeleri, Db2 depolama, bağlaşım olanağı depolama ve sayfa veri kümeleri için disk depolama gereksinimlerinizi planlarken bu konuyu kullanın.

Kuyruk yöneticisi veri kümelerinin nereye konacağını belirlemek için depolama yöneticinizle birlikte çalışın. Örneğin, depolama yöneticiniz size farklı veri kümesi tipleri için belirli DASD birimleri ya da SMS depolama sınıfları, veri sınıfları ve yönetim sınıfları verebilir.

- Günlük veri kümeleri DASD üzerinde olmalıdır. Bu günlükler, küçük bir yanıt süresine sahip yüksek G/Ç etkinliğine sahip olabilir ve yedeklenmeleri gerekmez.
- Arşiv günlükleri DASD ya da manyetik bantta olabilir. Bunlar yaratıldıktan sonra, bir yedekten sayfa kümesi kurtarılması gibi olağandışı bir durum dışında bir daha okunmayabilir. Uzun bir alıkoyma tarihleri olmalı.
- Sayfa kümelerinde düşük ila orta boy etkinlik olabilir ve düzenli olarak yedeklenmelidir. Yüksek kullanım sisteminde, günde iki kez yedeklenmeleri gerekir.
- BSDS veri kümeleri günlük olarak yedeklenmelidir; bunlar yüksek G/Ç etkinliğine sahip değildir.

Tüm veri kümeleri Db2 tarafından kullanılanlara benzer ve benzer bakım yordamları IBM MQ için kullanılabilir.

Veri depolamasını nasıl planlayacağınıza ilişkin ayrıntılar için aşağıdaki bölümlere bakın:

• **Günlükleri ve arşiv depolaması**

“Arşiv günlüklerini ne kadar süre tutmam gerekiyor?” sayfa 167 içinde, IBM MQ sisteminizin işlediği iletilerin hacmine ve etkin günlüklerin arşiv veri kümelerinize ne sıklıkta boşaltıldığına bağlı olarak etkin günlük ve arşiv veri kümelerinizin ne kadar depolama gerektirdiğini nasıl belirleyeceğinizi açıklar.

• **Db2 depolama**

“Db2 depolama” sayfa 184 , IBM MQ verileri için Db2 depolamanın ne kadar gerektirdiğini nasıl belirleyeceğinizi açıklar.

• **bağlaşım olanağı depolaması**

“Bağlaşım olanağı kaynaklarının tanımlanması” sayfa 174 içinde, bağlaşım olanağı yapılarınızı ne kadar büyük yapacağınızı nasıl belirleyeceğinizi açıklar.

• **Sayfa kümesi ve ileti depolama**

“Sayfa kümelerinizin ve arabellek havuzlarınızın planlanması” sayfa 151 içinde, uygulamalarınızın değiş tokuş ettiği iletilerin boyutlarına, bu iletilerin sayısına ve bunların oluşturulma ya da değiş tokuş hızına bağlı olarak sayfa veri kümelerinizin ne kadar depolama gerektirdiğini nasıl belirleyeceğinizi açıklar.

z/OS Depolama ve performans gereksinimleri hakkında daha fazla bilgi nereden edinebilir?

Depolama ve performans gereksinimlerine ilişkin daha fazla bilgi edinmek için bu konuyu başvuru olarak kullanın.

Aşağıdaki kaynaklardan daha fazla bilgi bulabilirsiniz:

Çizelge 20. Depolama gereksinimlerine ilişkin daha fazla bilgi nerede bulunur?	
Konu	Nereye bakıyorsunuz?
Sistem değiştiricileri	CSQ6SYSP ve Kuyruk Yöneticilerinin Uyarlanması portal uygulamalarının kullanılması
IBM MQ ürününü kurmak için gerekli depolama alanı	Program Dizini. Program Dizinine ilişkin yükleme bağlantıları için bkz. IBM MQ for z/OS Program Dizini PDF dosyaları .
IEALIMIT ve IEFUSI çıktıları	z/OS:MVS Installation Exits belgesinde IEALIMIT ve IEFUSI bölümüne bakın.
En son bilgiler	IBM MQ SupportPac Web sitesi IBM MQ ve diğer proje alanları için SupportPacs .

Çizelge 20. Depolama gereksinimlerine ilişkin daha fazla bilgi nerede bulunur? (devamı var)

Konu	Nereye bakıyorsunuz?
Hizmet tanımı aracılığıyla iş yükü yönetimi ve hedeflerin tanımlanması	z/OS MVS Planlama: İş Yükü Yönetimi

z/OS Sayfa kümelerinizin ve arabellek havuzlarınızın planlanması

Sayfa veri kümelerinizin ve arabellek havuzlarınızın ilk sayısını ve boyutlarını planlamanıza yardımcı olacak bilgiler.

Bu konu aşağıdaki kısımları içerir:

- [“Sayfa kümelerinizi planlayın” sayfa 151](#)
 - [Sayfa kümesi kullanımı](#)
 - [Sayfa kümesi sayısı](#)
 - [Sayfa kümelerinin boyutu](#)
 - [z/OS veri kümesi şifrelemesinin planlanması](#)
- [“Sayfa kümelerinizin boyutunu hesaplayın” sayfa 152](#)
 - [Sayfa kümesi sıfır](#)
 - [Sayfa kümesi 01-99](#)
 - [İletilere ilişkin depolama gereksiniminin hesaplanması](#)
- [“Dinamik sayfa kümesi genişletmesinin etkinleştirilmesi” sayfa 154](#)
- [“Arabellek havuzlarınızın tanımlanması” sayfa 155](#)

Sayfa kümelerinizi planlayın

Sayfa kümesi kullanımı

Kısa ömürlü iletiler için, normalde sayfa kümesinde birkaç sayfa kullanılır ve başlatma, denetim noktası ya da kapatma dışında veri kümelerinde çok az G/Ç bulunur ya da yoktur.

Uzun ömürlü iletiler için, ileti içeren sayfalar normalde diske yazılır. Bu işlem, yeniden başlatma süresini kısaltmak için kuyruk yöneticisi tarafından gerçekleştirilir.

Kısa ömürlü iletileri farklı sayfa kümelerine ve farklı arabellek havuzlarına yerleştirerek uzun ömürlü iletilerden ayırın.

Sayfa kümesi sayısı

Birkaç büyük sayfa kümesinin kullanılması IBM MQ yöneticisinin rolünü kolaylaştırabilir; bu, kuyrukların sayfa kümeleriyle eşlenmesini daha kolay hale getirerek daha az sayfa kümesine gereksinim duyacağınız anlamına gelir.

Birden çok, daha küçük sayfa kümelerinin kullanılması bir dizi avantaja sahiptir. Örneğin, bunların yedeklenmesi daha az zaman alır ve G/Ç, yedekleme ve yeniden başlatma sırasında paralel olarak yürütülebilir. Ancak bunun, her bir kuyruğu çok daha fazla sayıda sayfa kümesinde eşlemek için gerekli olan IBM MQ yöneticisinin rolüne önemli bir performans maliyeti eklediğini göz önünde bulundurun.

Aşağıdaki gibi en az beş sayfa kümesi tanımlayın:

- Nesne tanımlamaları için ayrılmış bir sayfa kümesi (sayfa kümesi sıfır)
- Sistemle ilgili iletiler için bir sayfa kümesi
- Performans açısından kritik uzun ömürlü iletiler için bir sayfa kümesi
- Performans açısından kritik kısa ömürlü iletiler için bir sayfa kümesi

- Diğer tüm iletiler için ayarlanmış bir sayfa

“Arabellek havuzlarınızın tanımlanması” sayfa 155 , iletilerinizi bu şekilde sayfa kümelerine dağıtmanın performans avantajlarını açıklar.

Sayfa kümelerinin boyutu

Beklenen en yüksek ileti kapasitesi için sayfa kümelerinizdeki yeterli alanı tanımlayın. Örneğin, bir kuyruk sunucusu programı çalışmadığı için bir ileti oluşturmanın gelişmesi gibi beklenmedik en yüksek kapasiteyi göz önünde bulundurun. Bunu, sayfa kümesini ikincil kapsamlarla ayırarak ya da dinamik sayfa kümesi genişletmesini etkinleştirerek yapabilirsiniz. Daha fazla bilgi için bkz “[Dinamik sayfa kümesi genişletmesinin etkinleştirilmesi](#)” sayfa 154. Bir sayfa kümesini küçültmek zordur, bu nedenle genellikle daha küçük bir sayfa kümesi ayırmak ve gerektiğinde genişlemesine izin vermek daha iyidir.

Sayfa kümesi boyutlarını planlarken, uygulama dışı ileti verileri de içinde olmak üzere, üretilebilecek tüm iletileri göz önünde bulundurun. Örneğin, tetikleyici iletiler, olay iletileri ve uygulamanızın istediği rapor iletileri.

Büyük bir sayfa kümesinin geri yüklenmesi daha uzun sürdüğü için, bir yedekten geri yükleme sırasında sayfa kümesinin kurtarılması için geçen süreyi sayfa kümesinin boyutu belirler.

Not: Bir sayfa kümesinin kurtarılması, yedekleme işleminden bu yana yazılan günlük kayıtlarını işlemek için kuyruk yöneticisinin aldığı süreye de bağlıdır; bu süre, yedekleme sıklığına göre belirlenir. Daha fazla bilgi için bkz “[Yedekleme ve kurtarma planlaması](#)” sayfa 186.

Not: 4 GB ' den büyük sayfa kümeleri için SMS genişletilmiş adreslenirliğinin kullanılması gerekir.

z/OS veri kümesi şifrelemesinin planlanması

z/OS veri kümesi şifreleme özelliğini, IBM MQ for z/OS 9.1.4 ya da daha sonra çalışan kuyruk yöneticileri için sayfa kümelerine uygulayabilirsiniz.

Bu sayfa kümelerini EXTENDED öznitelikleriyle ve verilerin AES şifreli olmasını sağlayan bir veri kümesi anahtar etiketiyle ayırmanız gerekir.

[veri kümesi şifrelemesiyle IBM MQ for z/OS üzerinde atıl durumdaki veriler için gizlilik](#) başlıklı bölüme bakın. ek bilgi için.

Sayfa kümelerinizin boyutunu hesaplayın

Kuyruk yöneticisi nesne tanımlamaları (örneğin, kuyruklar ve işlemler) için, bu nesnelere sabit büyüklükte ve kalıcı olduğu için depolama gereksinimini hesaplamak kolaydır. Bununla birlikte, iletiler için hesaplama aşağıdaki nedenlerden ötürü daha karmaşıktır:

- İletilerin boyutu değişir.
- Mesajlar geçici.
- Alınan iletilerin kapladığı alan, zamanuysuz bir işlem tarafından belirli aralıklarla geri istenir.

Ağ durursa iletiler için ek kapasite sağlayan 4 GB ' den büyük sayfa kümeleri, gerekirse oluşturulabilir. Var olan sayfa kümeleri değiştirilemez. Bunun yerine, genişletilmiş adreslenebilirlik ve genişletilmiş biçim özniteliklerine sahip yeni sayfa kümeleri oluşturulmalıdır. Yeni sayfa kümeleri eskilerinkiyle aynı fiziksel boyutta olmalıdır ve eski sayfa kümeleri yeni sayfa kümelerine kopyalanmalıdır. Geriye doğru geçiş gerekiyorsa, sayfa kümesi sıfır değiştirilmemelidir. 4 GB ' den küçük sayfa kümeleri yeterliyse, herhangi bir işlem yapılması gerekmez.

Sayfa kümesi sıfır

Sayfa kümesi sıfır, nesne tanımlamaları için ayrılmıştır.

Sayfa kümesi sıfır için gereken depolama alanı:


```
(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)
```

Sayfa kümesi veri kümesine ilişkin kümede belirlenecek kayıt sayısını belirlemek için bu değeri 4096 'ya bölün.

Paylaşılan havuzda saklanan nesnelere izin vermeniz gerekmez; ancak, sayfa kümesi sıfıra (GROUP ya da QMGR) saklanan ya da kopyalanan nesnelere izin vermeniz gerekir.

Oluşturabileceğiniz toplam nesne sayısı, sayfa kümesi sıfır kapasitesiyle sınırlıdır. Tanımlayabileceğiniz yerel kuyruk sayısı 524 287 ile sınırlıdır.

Sayfa kümeleri 01-99

Sayfa kümeleri 01-99 için, her sayfa kümesi için gereken depolama alanı, o sayfa kümesinde saklanan iletilerin sayısına ve boyutuna göre belirlenir. (Paylaşılan kuyruklardaki iletiler sayfa kümelerinde saklanmaz.)

Sayfa kümesi veri kümesine ilişkin kümede belirlenecek kayıt sayısını belirlemek için bu değeri 4096 'ya bölün.

İletiler için depolama gereksiniminin hesaplanması

Bu bölümde, iletilerin sayfalarda nasıl saklandığı açıklanmaktadır. Bunu anlamak, iletilerinizi için ne kadar sayfa kümesi depolaması tanımlamanız gerektiğini hesaplamana yardımcı olabilir. Bir sayfa kümesindeki tüm iletiler için gereken yaklaşık alanı hesaplamak için, sayfa kümesiyle eşleyen tüm kuyrukların kuyruk derinliği üst sınırını ve bu kuyruklardaki iletilerin ortalama boyutunu göz önünde bulundurmanız gerekir.

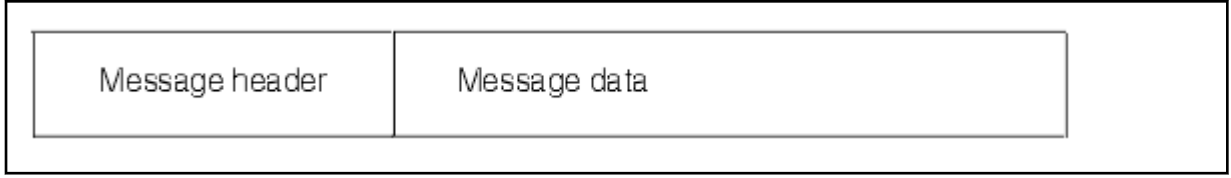
Not: Bu bölümde verilen yapıların ve denetim bilgilerinin boyutları, ana yayınlar arasında değişebilir. IBM MQ yayınına özgü ayrıntılar için bkz. [SupportPac MP16 - z/OS Kapasite planlama ve ayarlama için IBM MQ ve IBM MQ Ürün Ailesi-Performans Raporları](#)

IBM MQ denetimi dışındaki nedenlerle (örneğin, iletişim protokolünüzle ilgili bir sorun nedeniyle) "alma" iletilerinin gecikmesine izin vermeniz gerekir. Bu durumda, iletilerin "koyma" oranı "alma" oranını çok aşabilir. Bu, sayfa kümelerinde depolanan ileti sayısında büyük bir artışa ve talep edilen depolama boyutunda sonuçta ortaya çıkan bir artışa neden olabilir.

Sayfa kümesindeki her sayfa 4096 bayt uzunluğundadır. Sabit üstbilgi bilgilerine izin veren her sayfada, iletilerin saklanması için kullanılabilir 4057 bayt yer vardır.

Her ileti için gereken alanı hesaplarken, göz önünde bulundurmanız gereken ilk şey, iletinin bir sayfaya mı (kısa bir ileti) sığması, yoksa iki ya da daha fazla sayfaya mı (uzun bir ileti) bölünmesi gerekir. İletiler bu şekilde bölündüğünde, alan hesaplamalarınızda ek denetim bilgilerine izin vermeniz gerekir.

Alan hesaplaması amacıyla, bir ileti aşağıdaki gibi gösterilebilir:



İleti üstbilgisi bölümü, ileti tanımlayıcısı ve diğer denetim bilgilerini içerir; bu bilgilerin boyutu iletinin boyutuna bağlı olarak değişir. İleti verileri bölümü, tüm gerçek ileti verilerini ve diğer üstbilgileri (örneğin, iletim üstbilgisi ya da IMS köprüsü üstbilgisi) içerir.

Genellikle iletiler için gereken toplam alanın %1 'inden az olan sayfa kümesi denetim bilgileri için en az iki sayfa gerekir.

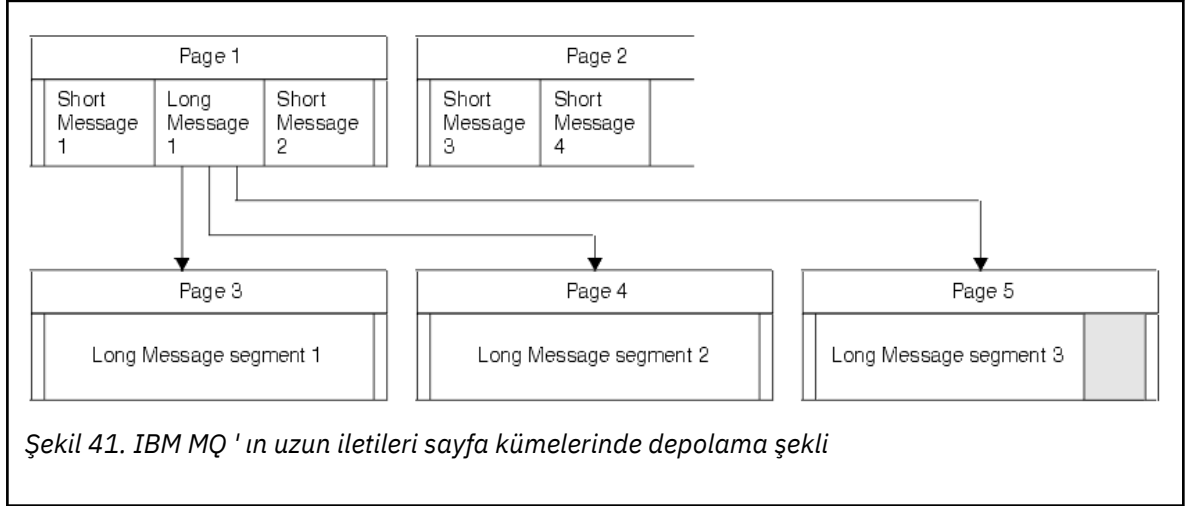
Kısa iletiler

Kısa bir ileti, bir sayfaya sığan bir ileti olarak tanımlanır.

Küçük iletiler her sayfada bir tane saklanır.

Uzun iletiler

İleti verilerinin boyutu 3596 bayttan büyükse, ancak 4 MB ' den büyük değilse, ileti uzun bir ileti olarak sınıflandırılır. Uzun bir ileti sunulduğunda, IBM MQ iletiyi bir sayfa dizisinde saklar ve bu sayfaları gösteren denetim bilgilerini, kısa bir iletiyi saklayacağı şekilde saklar. Bu, [Şekil 41 sayfa 154](#) içinde gösterilir:



Çok uzun iletiler

Çok uzun iletiler, boyutu 4 MB ' den büyük olan iletilerdir. Bunlar, her 4 MB ' nin 1037 sayfa kullanması için depolanır. Kalan her şey, yukarıda açıklandığı gibi, uzun bir iletiyle aynı şekilde saklanır.

z/OS Dinamik sayfa kümesi genişletmesinin etkinleştirilmesi

Kuyruk yöneticisi çalışırken sayfa kümeleri dinamik olarak genişletilebilir. Bir sayfa kümesi 123 kapsama sahip olabilir ve birden çok disk birimine dağıtılabilir.

Bir sayfa kümesi her genişlediğinde, yeni bir veri kümesi kapsamı kullanılır. Kuyruk yöneticisi, kapsam sayısı üst sınırına ulaşıncaya ya da uygun birimlerde ayırma için kullanılabilir depolama alanı kalmayınca kadar, gerektiğinde bir sayfa kümesini genişletmeye devam eder.

Sayfa kümesi genişletme işlemi yukarıdaki nedenlerden biri nedeniyle başarısız olduğunda, kuyruk yöneticisi sayfa kümesini başka genişletme girişimi olmayacak şekilde işaretler. Bu işaret, sayfa kümesi EXPAND (SYSTEM) olarak değiştirilerek ilk durumuna getirilir.

Sayfa kümesi genişletmesi, sayfa kümesindeki var olan alanın %90 ' ı ayrıldığında, diğer tüm sayfa kümesi etkinliğinde zamanuyumsuz olarak gerçekleşir.

Sayfa kümesi genişletme işlemi, yeni ayrılan kapsamı biçimlendirir ve kuyruk yöneticisi tarafından kullanılabilir hale getirir. Ancak, tüm kapsam biçimlendirilinceye kadar alanın hiçbiri kullanılamaz. Bu, genişlemenin büyük olasılıkla biraz zaman alacağı ve uygulamaların genişleme tamamlanmadan önce sayfa kümesinin kalan %10 'unu doldurmaları durumunda 'engellebileceği' anlamına gelir.

Örnek thlqual.SCSQPROC(CSQ4PAGE), ikincil kapsamların nasıl tanımlanacağını gösterir.

Yeni kapsamların boyutunu denetlemek için, DEFINE PSID ve ALTER PSID komutlarının EXPAND anahtar sözcüğünün aşağıdaki seçeneklerinden birini kullanabilirsiniz:

- KULLANICI
- SYSTEM
- YOK

KULLANICI

Sayfa kümesi ayrıldığında belirlenen ikincil kapsam boyutunu kullanır. Bir değer belirtilmediyse ya da sıfır değeri belirtildiyse, dinamik sayfa kümesi genişletmesi oluşmaz.

Sayfa kümesi genişletmesi, sayfadaki alan %90 kullanıldığında ve diğer sayfa kümesi etkinliğiyle zamanuyumsuz olarak gerçekleştirildiğinde oluşur.

Bu, bir defada tek bir kapsamdan daha fazla genişlemeye neden olabilir.

Aşağıdaki örneği göz önünde bulundurun: Birincil kapsamı 100.000 sayfa ve ikincil kapsamı 5000 sayfa olan bir sayfa kümesi ayırın. 9999 sayfa gerektiren bir ileti konmuştur. Sayfa kümesi zaten 85.000 sayfa kullanıyorsa, iletinin yazılması %90 tam sınırı (90,000 sayfa) aşmaktadır. Bu noktada, sayfa kümesi boyutu 105.000 sayfaya çıkarılacak şekilde, 100.000 sayfalık birincil kapsama ek bir ikincil kapsam ayrılır. İletinin kalan 4999 sayfası yazılmaya devam ediyor. Kullanılan sayfa alanı, 105.000 sayfalık güncellenmiş sayfa kümesi boyutunun %90 'ı olan 94.500 sayfaya ulaştığında, 5000 sayfa kapsamı ayrılır ve sayfa kümesi boyutu 110.000 sayfaya çıkarılır. MQPUT ' nin sonunda sayfa kümesi iki kez genişletildi ve 94.500 sayfa kullanıldı. Ayrılmalarına rağmen, ikinci sayfa kümesi genişletmesindeki sayfaların hiçbiri kullanılmadı.

Yeniden başlatma sırasında, önceden kullanılan bir sayfa kümesi daha küçük bir veri kümesiyle değiştirildiyse, daha önce kullanılan veri kümesinin boyutuna ulaşıncaya kadar genişletilir. Bu boyuta ulaşmak için yalnızca bir kapsam gereklidir.

SYSTEM

Sayfa kümesi tanımlandığında belirtilen ikincil kapsam boyutunu yoksayar. Kuyruk yöneticisi, yürürlükteki sayfa kümesi büyüklüğünün yaklaşık %10 'u olan bir değer belirler. Değer, en yakın DASD silindirene yuvarlanır.

Bir değer belirtilmediyse ya da sıfır değeri belirtildiyse, dinamik sayfa kümesi genişletmesi devam edebilir. Kuyruk yöneticisi, yürürlükteki sayfa kümesi büyüklüğünün yaklaşık %10 'u olan bir değer ayarlar. Yeni değer, DASD ' nin özelliklerine bağlı olarak yuvarlanır.

Sayfa kümesi genişletmesi, sayfa kümesindeki alan yaklaşık %90 kullanıldığında ve diğer sayfa kümesi etkinliğiyle zamanuyumsuz olarak gerçekleştirildiğinde oluşur.

Yeniden başlatma sırasında, önceden kullanılan bir sayfa kümesi daha küçük bir veri kümesiyle değiştirildiyse, daha önce kullanılan veri kümesinin boyutuna ulaşıncaya kadar genişletilir.

YOK

Başka bir sayfa kümesi genişletmesi gerçekleşmeyecek.

İlgili başvurular

[ALTER PSID \(PSID\)](#)

[PSID TANIMLAYIN](#)

[YERLEŞİM YERİ](#)

Arabellek havuzlarınızın tanımlanması

Tanımlamanız gereken arabellek havuzu sayısını ve ayarlarını planlamak için bu konuyu kullanın.

Bu konu ařađıdaki bölümlere ayrılmıřtır:

1. [“Tanımlanacak arabellek havuzu sayısına karar verin” sayfa 156](#)
2. [“Her arabellek havuzu için ayarlara karar verin” sayfa 157](#)
3. [“Beklenen yük altında arabellek havuzlarının başarımını izleyin” sayfa 157](#)
4. [“Arabellek havuzu özelliklerini ayarla” sayfa 157](#)

Tanımlanacak arabellek havuzu sayısına karar verin

Başlangıçta dört arabellek havuzu tanımlamanız gerekir:

Arabellek havuzu 0

Nesne tanımlamaları (sayfa kümesi sıfır) ve başarım açısından kritik, sistemle ilgili ileti kuyrukları (örneğin, SYSTEM.CHANNEL.SYNCO kuyruđu ve SYSTEM.CLUSTER.COMMAND.QUEUE ve SYSTEM.CLUSTER.REPOSITORY.QUEUE KUYRUKLARI).

Ancak çok sayıda kanal ya da kümeleme kullanılacaksa, *Arabellek havuzu özelliklerini ayarla* seçeneđindeki [“7” sayfa 158](#) noktasının dikkate alınması önemlidir.

Kullanıcı iletileri için geri kalan üç arabellek havuzunu kullanın.

Arabellek havuzu 1

Uzun ömürlü önemli iletiler için kullanın.

Uzun ömürlü iletiler, sistemde iki denetim noktasından daha uzun süre kalır ve bu süre içinde sayfa kümesine yazılır. Uzun ömürlü birçok iletiniz varsa, bu arabellek havuzu nispeten küçük olmalıdır; böylece sayfa kümesi G/Ç 'si eşit olarak dağıtılır (eski iletiler, arabellek havuzu %85 dolu olduđunda DASD ' ye yazılır).

Arabellek havuzu çok büyükse ve arabellek havuzu hiçbir zaman %85 doluysa, sayfa kümesi G/Ç işlemi denetim noktası işleninceye kadar geciktirilir. Bu, sistem genelinde yanıt sürelerini etkileyebilir.

Yalnızca birkaç uzun ömürlü ileti bekliyorsanız, bu arabellek havuzunu, tüm bu iletileri tutacak kadar büyük olacak şekilde tanımlayın.

Arabellek havuzu 2

Performans açısından kritik, kısa ömürlü iletiler için kullanın.

Normalde çok az arabellek kullanan, yüksek düzeyde bir arabellek yeniden kullanımı vardır. Ancak, bu arabellek havuzunu beklenmeyen ileti birikimine izin verecek şekilde (örneğin, bir sunucu uygulaması başarısız olduđunda) geniş yapmalısınız.

Arabellek havuzu 3

Diđer tüm (genellikle performans açısından kritik olmayan) iletiler için kullanın.

Gönderilmeyen iletiler kuyruđu SYSTEM.COMMAND.* KOMUTU kuyruklar ve SYSTEM.ADMIN.* kuyruklar arabellek havuzu 3 ile de eşlenebilir.

Sanal saklama alanı kısıtlamaları varsa ve arabellek havuzlarının daha küçük olması gerekiyorsa, arabellek havuzu 3, büyüklük azaltma için ilk adaydır.

Ařađıdaki durumlarda ek arabellek havuzları tanımlamanız gerekebilir:

- Belirli bir kuyruğun tecrit edilmesi gerektiđi biliniyorsa, çeřitli zamanlarda farklı davranıřlar sergilediđi için olabilir.
 - Böyle bir kuyruk, deđiřen kořullar altında mümkün olan en iyi başarımı gerektirebilir ya da bir arabellek havuzundaki diđer kuyrukları olumsuz etkilememesi için yalıtılması gerekebilir.
 - Bu kuyruğun her biri kendi arabellek havuzunda ve sayfa kümesinde yalıtılabilir.
- Hizmet sınıfı nedenleriyle farklı kuyruk kümelerini birbirinden yalıtma istiyorsunuz.
 - Daha sonra her kuyruk kümesi, [Arabellek havuzu ayarlarına iliřkin önerilen tanımlarkısmında](#) açıklandığı gibi, 1 ya da 2 numaralı arabellek havuzlarından birini ya da her ikisini birden gerektirebilir ve belirli tipte birkaç arabellek havuzunun oluřturulmasını gerektirir.

Her arabellek havuzu için ayarlara karar verin

“Tanımlanacak arabellek havuzu sayısına karar verin” sayfa 156’inde açıklanan dört arabellek havuzu kullanıyorsanız, Arabellek havuzu ayarları için önerilen tanımlar arabellek havuzlarının büyüklüğü için iki değer kümesi verir.

İlk set bir test sistemi için uygundur, diğeri ise bir üretim sistemi veya sonunda bir üretim sistemi haline gelecek bir sistem için. Her durumda, arabellek havuzlarınızı **LOCATION(ABOVE)** özniteliğiyle tanımlayın

Tanımlama ayarı	Test sistemi	Üretim sistemi
BUFFPOOL 0	1050 arabellekler	50 000 arabellekler
BUFFPOOL 1.	1050 arabellekler	20 000 arabellekler
BUFFPOOL 2	1050 arabellekler	50 000 arabellekler
BUFFPOOL 3.	1050 arabellekler	20 000 arabellekler

Önerilen dört arabellek havuzundan fazlasına gereksinim duyarsanız, arabellek havuzundaki kuyrukların beklenen davranışını en doğru şekilde açıklayan arabellek havuzunu (1 ya da 2) seçin ve Arabellek havuzu ayarları için önerilen tanımlar içindeki bilgileri kullanarak bu havuzu boyutlandırın.

MEMLIMIT değerinin, tüm arabellek havuzlarının çubuğun üzerinde bulunabilmesi için yeterli yüksek olmasına dikkat edin.

Beklenen yük altında arabellek havuzlarının başarımını izleyin

Arabellek havuzu başarım istatistiklerini çözümlenerek arabellek havuzlarının kullanımını izleyebilirsiniz. Özellikle, arabellek havuzlarının QPSTSOS, QPSTSTLA ve QPSTDMC değerlerinin sıfır olarak kalabilmesi için yeterli büyüklükte olduğundan emin olmanız gerekir.

Daha fazla bilgi için bkz. Arabellek yöneticisi veri kayıtları.

Arabellek havuzu özelliklerini ayarla

Gerekirse, arabellek havuzu ayarlarını “Her arabellek havuzu için ayarlara karar verin” sayfa 157’ den ayarlamak için aşağıdaki noktaları kullanın.

“Beklenen yük altında arabellek havuzlarının başarımını izleyin” sayfa 157 ' in performans istatistiklerini kılavuz olarak kullanın.

1. IBM MQ' un önceki bir sürümünden geçiş yapıyorsanız, yalnızca daha fazla gerçek depolama alanı varsa, var olan ayarlarınızı değiştirin.
2. Genel olarak, daha büyük arabellek havuzları başarım için daha iyidir ve arabellek havuzları çubuğun üstündeyse çok daha büyük olabilir.

Ancak, arabellek havuzlarının gerçek saklama alanında yerleşik olması için her zaman yeterli gerçek saklama alanı olmalıdır. Sayfalamaya neden olmayan daha küçük arabellek havuzlarının olması, büyük havuzlara göre daha iyidir.

Ayrıca, onu kullanan sayfa kümelerinin toplam boyutundan daha büyük bir arabellek havuzunun olmasının bir anlamı yoktur, ancak oluşma olasılığı varsa sayfa kümesi genişletmesini göz önünde bulundurmanız gerekir.

3. Daha iyi uygulama yalıtımı sağladığı için arabellek havuzu başına bir sayfa kümesi hedefleyin.
4. Yeterli gerçek saklama alanınız varsa, işletim sistemi arabellek havuzlarınızı hiçbir zaman sayfalanmayacak şekilde, arabellek havuzunuzda sayfa değişmez arabellekler kullanabilirsiniz.

Bu özellikle, arabellek havuzunun G/Ç 'den önce arabellekleri sayfa düzeltmeye ve daha sonra bunların sayfa düzeltmeye ilişkin CPU maliyetini kaydettiği için, arabellek havuzunun çok fazla G/Ç' ye maruz kalma olasılığı varsa önemlidir.

5. Çubuğun altına sığacak kadar küçük olsalar da, arabellek havuzlarını çubuğun üzerinde bulmanın çeşitli yararları vardır. Bunlar:
- 31 bit sanal depolama kısıtlaması-örneğin, ortak depolama için daha fazla alan.
 - Bir arabellek havuzunun büyüklüğünün, yoğun bir şekilde kullanıldığında beklenmedik bir şekilde artırılması gerekiyorsa, kuyruk yöneticisi ve iş yükü için, zaten çubuğun üzerinde olan bir arabellek havuzuna daha fazla arabellek eklenerek, arabellek havuzunun çubuğun üstüne taşınmasından ve daha sonra daha fazla arabellek eklenmesinden daha az etki ve risk olur.
6. Arabellek havuzu sıfır ve kısa ömürlü iletiler (arabellek havuzu 2) için arabellek havuzunu ayarlayın; böylece %15 boş eşik hiçbir zaman aşılmaz (yani, QPSTCBSL bölü QPSTNBUF her zaman %15 'ten büyüktür). Arabelleklerin %15 'inden fazlası boş kalırsa, bu arabellek havuzlarını kullanan sayfa kümelerindeki G/Ç ' ler normal çalışma sırasında büyük ölçüde önlenebilir, ancak iki denetim noktasından daha eski iletiler sayfa kümelerine yazılır.



Uyarı: Bu parametrelere ilişkin en iyi değer, tek tek sistemin özelliklerine bağlıdır. Belirtilen değerler yalnızca bir kılavuz olarak amaçlanır ve sisteminiz için uygun olmayabilir.

7. **SISTEM.*** kuyruklar çok derinleşiyor, örneğin **SYSTEM.CHANNEL.SYNCO**, yeterli saklama alanı varsa, kendi arabellek havuzuna yerleştirilmekten yararlanabilir.

IBM MQ SupportPac [MP16 - z/OS Kapasite planlama ve ayarlama için IBM MQ](#) , arabellek havuzlarının ayarlanmasıyla ilgili daha fazla bilgi sağlar.

z/OS **Günlük kaydı ortamınızın planlanması**

Günlüklerin sayısını, boyutunu ve yerleşimini ve IBM MQ tarafından kullanılan günlük arşivlerini planlamak için bu konuyu kullanın.

Günlükler aşağıdakiler için kullanılır:

- Kalıcı iletilere ilişkin kurtarma bilgilerini yaz
- Kalıcı iletileri kullanarak iş birimlerine ilişkin bilgileri kaydet
- Kuyruk tanımlama gibi nesnelere yapılan değişikliklerle ilgili bilgileri kaydet
- Yedek CF yapıları

ve diğer iç bilgiler için.

IBM MQ günlük kaydı ortamı, tek ya da çift etkin günlüklerin olup olmaması, arşiv günlüğü birimleri için hangi ortamın kullanılacağı ve kaç günlük arabelleği bulunduracağı gibi seçenekleri belirtmek için sistem parametre makroları kullanılarak oluşturulur.

Bu makrolar, [Create the bootstrap and log data sets](#) (Önyükleme ve günlük veri kümelerinin yaratılması) ve [Tailor to system parameter module](#) (Sistem parametre modülünüzü ayarlayın) başlıklı konuda açıklanmıştır.

Not: Kuyruk paylaşım grupları kullanıyorsanız, önyükleme ve günlük veri kümelerini **SHAREOPTIONS (2 3)** ile tanımladığınızdan emin olun.

Bu bölüm, aşağıdaki konularla ilgili bilgi içerir:

z/OS **Günlük veri kümesi tanımlamaları**

Günlük veri kümelerinize ilişkin en uygun yapılandırmaya karar vermek için bu konuyu kullanın.

Bu konu, aşağıdaki soruları yanıtlamanıza yardımcı olacak bilgiler içerir:

- [Kuruluşunuz tek ya da çift günlük kaydı kullanmalı mı?](#)
- [Kaç etkin günlük veri kümesine ihtiyacınız var?](#)
- [“Etkin günlükler ne kadar büyük olmalıdır?” sayfa 160](#)
- [Etkin günlük yerleşimi](#)
- [“z/OS veri kümesi şifrelemesi ile etkin günlük şifrelemesi” sayfa 161](#)

Kuruluşunuz tek ya da ikili günlük kaydı kullanmalı mı?

Genel olarak, veri kaybı riskini en aza indirmek için üretim için ikili günlük kaydı kullanmalısınız. Test sisteminizin üretimi yansıtmasını istiyorsanız, her ikisi de ikili günlük kaydı kullanılmalıdır; tersi durumda test sistemleriniz tek günlük kaydı kullanabilir.

Tek günlük kaydı verileriyle tek bir günlük veri kümesi kümesine yazılır. İkili günlük kaydı verileri iki günlük veri kümesine yazılır; bu nedenle, bir günlük veri kümesiyle ilgili bir sorun olması durumunda (örneğin, yanlışlıkla silinen veri kümesi), diğer günlük kümesindeki eşdeğer veri kümesi, verileri kurtarmak için kullanılabilir.

İkili günlük kaydı ile tek günlük kaydınıza göre iki kat daha fazla DASD gerekir.

İkili günlük kaydı kullanıyorsanız, veri kurtarma için yeterli yetkilendirmeyi sağlamak üzere ikili BSDS'lerden ve ikili arşivlemeden de yararlanabilirsiniz.

İkili etkin günlük kaydı, küçük bir performans maliyeti ekler.



Uyarı: Metro Mirrorgibi disk ikizleme teknolojilerinin kullanımı, ikili günlük kaydı ve ikili BSDS için bir yedek olmak zorunda değildir. İkizlenmiş bir veri kümesi yanlışlıkla silinirse, her iki kopya da kaybolur.

Kalıcı iletiler kullanırsanız, tek günlük kaydı kapasite üst sınırını %10-30 artırabilir ve yanıt sürelerini de artırabilir.

Tek günlük kaydı 2-310 etkin günlük veri kümesi kullanırken, çift günlük kaydı aynı sayıda etkin günlük sağlamak için 4-620 etkin günlük veri kümesi kullanır. Bu nedenle, tek günlük kaydı, günlüğe kaydedilen veri miktarını azaltır; bu, kuruluşunuz G/Ç kısıtlaması durumunda önemli olabilir.

Kaç etkin günlük veri kümesine ihtiyacınız var?

Günlük sayısı, kuyruk yöneticinizin etkinliklerine bağlıdır. Düşük verimi olan bir test sistemi için üç etkin günlük veri kümesi uygun olabilir. Yüksek verim üretim sistemi için kullanılabilir günlük sayısı üst sınırını isteyebilirsiniz; bu nedenle, günlüklerin boşaltılmasıyla ilgili bir sorun varsa, sorunları çözmek için daha fazla zamanınız olur.

En az üç etkin günlük veri kümeniz olmalıdır, ancak daha fazla tanımlama tercih edilir. Örneğin, bir günlüğü doldurmak için geçen süre, en yoğun yükleme sırasında bir günlüğü arşivlemek için geçen süreye yaklaşıyorsa, daha fazla günlük tanımlayın.

Not: Sayfa kümeleri ve etkin günlük veri kümeleri, genişletilmiş adres birimlerinin (EAV) genişletilmiş adresleme alanı (EAS) bölümünde bulunmaya uygundur ve bir arşiv günlüğü veri kümesi EAS içinde de yer alabilir.

Günlük arşivlemede olası gecikmeleri dengelemek için daha fazla günlük de tanımlamanız gerekir. Manyetik bantta arşiv günlüklerini kullanıyorsanız, manyetik bantı takmak için gerekli süreyi verin.

Sistemin DASD eksikliği nedeniyle ya da manyetik bantta yazamaması nedeniyle arşivlenememesi durumunda, bir günlük verileri tutmak için yeterli etkin günlük alanına sahip olmayı göz önünde bulundurun. Tüm etkin günlükler doldurulursa, IBM MQ kalıcı iletileri ya da işlemleri işleyemez. Yeterli etkin günlük alanına sahip olmak çok önemlidir.

Arşiv gecikmelerinin ya da sorunlarının etkisini en aza indirmenin bir yolu olarak yeni etkin günlük veri kümeleri dinamik olarak tanımlanabilir. Yeni veri kümeleri, etkin günlükte yer olmaması nedeniyle kuyruk yöneticisinin 'durmasını' önlemek için **DEFINE LOG** komutu kullanılarak hızla çevrimiçi duruma getirilebiliyor.

31 'den fazla etkin günlük veri kümesi tanımlamak istiyorsanız, günlük kaydı ortamınızı sürüm 2 biçimi BSDS kullanacak şekilde yapılandırmanız gerekir. Sürüm 2 biçimi BSDS kullanıldıktan sonra, her günlük kopyalama halkası için en çok 310 etkin günlük veri kümesi tanımlanabilir. BSDS sürüm 2 biçimine nasıl dönüştüreceğinizi öğrenmek için bkz. [“Adreslenebilir günlük aralığı üst sınırını artırmayı planlama” sayfa 168](#).

Kuyruk yöneticinizin günlük eşleme yardımcı programını (CSQJU004) çalıştırarak ya da kuyruk yöneticisini kullanıma hazırlama sırasında yayınlanan CSQJ034I iletisini kullanarak sürüm 2 ya da daha yüksek bir BSDS kullanıp kullanmadığını anlayabilirsiniz. CSQJ034I iletisindeki FFFFFFFFFFFFFFFFFFgünlük RBA aralığının sonu, BSDS biçiminin sürüm 2 ya da üstü olduğunu gösterir. CSQJ034I iletisindeki 0000FFFFFFFFFFFFFFFFgünlük RBA aralığının sonu, BSDS sürüm 1 biçimindeki bir BSDS 'nin kullanımda olduğunu gösterir.

Bir kuyruk yöneticisi BSDS biçimini kullanan bir sürüm 2 ya da üstü kullanıyorsa, günlük kopyalama halkasına dinamik olarak 31 'den fazla etkin günlük veri kümesi eklemek için **DEFINE LOG** komutunu kullanabilirsiniz.

Etkin günlükler ne kadar büyük olmalıdır?

IBM MQ 8.0' den diske arşivlenirken desteklenen etkin günlük boyutu üst sınırı 4 GB 'dir. Ürünün önceki yayınlarında, diske arşivleme sırasında desteklenen etkin günlük boyutu üst sınırı 3 GB 'dir.

Manyetik banda arşivleme sırasında etkin günlük boyutu üst sınırı 4 GB 'dir.

Üretim ve test sistemleri için en az 1 GB boyutunda etkin günlükler yaratmanız gerekir.

Önemli: Veri kümelerini ayırırken dikkatli olmanız gerekir; IDCAMS, ayırdığınız boyutu yukarı yuvarlar.

3 GB ' lik bir günlük ayırmak için aşağıdaki seçeneklerden birini belirleyin:

- Silindirler (4369)
- Megabayt (3071)
- İZLER (65535)
- KAYIT (786420)

Bunlardan herhangi biri 2.99995 GB ayırır.

4GB günlüğü ayırmak için aşağıdaki seçeneklerden birini belirtin:

- Silindirler (5825)
- Megabayt (4095)
- İZLER (87375)
- KAYIT (1048500)

Bu ürünlerden herhangi biri 3.9997 GB ayırır.

Veri kümesinin birden çok birime dağıtıldığı şeritli veri kümeleri kullanılırken, paylaşırma için kullanılan her DASD biriminde belirtilen boyut değeri ayrılır. Bu nedenle, paylaşırma için 4 GB günlük ve dört birim kullanmak istiyorsanız şunları belirtmeniz gerekir:

- CYLinders (1456)
- Megabayt (1023)

Bu özniteliklerin belirlenmesi $4 * 1456 = 5824$ Silindir ya da $4 * 1023 = 4092$ Megabayt ayırır.

Not: Genişletilmiş biçim veri kümeleri kullanılırken veri paylaşırma desteklenir. Bu genellikle depolama yöneticisi tarafından ayarlanır.

Yordamın uygulanmasına ilişkin bilgi için [Etkin günlüğün boyutunun artırılması](#) başlıklı konuya bakın.

Etkin günlük yerleşimi

Kuyruk yöneticileri için depolama havuzları ayarlamak üzere depolama yönetimi ekibinizle birlikte çalışmanız gerekir. Aşağıdakileri göz önünde bulundurmanız gerekir:

- Kuyruk yöneticilerinin doğru SMS tanımlarını kullanması için bir adlandırma kuralı.

- Etkin ve arşiv günlükleri için gereken alan. Depolama havuzunuzda, tüm bir günün etkin günlükleri için yeterli alan olmalıdır.
- Başarımı ve başarısızlıklara dayanıklılığı.

Performans nedenleriyle, etkin günlük veri kümelerinizi paylaşmayı düşünmeniz gerekir. G/Ç, birden çok birime dağıtılır ve G/Ç yanıt sürelerini azaltarak daha yüksek verim sağlar. Paylaşırma işlemini kullanırken etkin günlüklerin boyutunu ayırmaya ilişkin bilgi için önceki metne bakın.

RMF ya da benzer bir üründen gelen raporları kullanarak G/Ç istatistiklerini gözden geçirmeniz gerekir. Veri kümelerinin konumu nedeniyle gecikme olmadığından emin olmak için IBM MQ veri kümeleri için bu istatistiklerin aylık (ya da daha sık) gözden geçirilmesini gerçekleştirin.

Bazı durumlarda, çok fazla IBM MQ sayfa kümesi G/Ç olabilir ve bu, aynı DASD ' de bulunuyorsa IBM MQ günlük başarımını etkileyebilir.

İkili günlük kaydı kullanıyorsanız, her bir etkin ve arşiv günlüğü kümesinin ayrı tutulduğundan emin olun. Örneğin, bunları ayrı DASD altsistemlerine ya da farklı aygıtlara ayırın.

Bu, birimlerden biri bozulursa ya da yok edilirse ikisinin de kaybolma riskini azaltır. Günlüğün her iki kopyası da kaybolursa, veri kaybı olasılığı yüksektir.

Yeni bir etkin günlük verisi yarattığınızda, bunu [CSQJUFMT](#) kullanarak önceden biçimlendirmeniz gerekir. Günlük önceden biçimlendirilmediyse, kuyruk yöneticisi günlüğü ilk kez kullanıldığında biçimlendirir ve bu da performansı etkiler.

Büyük dönen disklere sahip eski DASD ile, en iyi performansı elde etmek için hangi birimlerin kullanıldığına dikkat etmeniz gerekiyordu.

Verilerin birçok PC boyutlu diske dağıtıldığı modern DASD ile, hangi birimlerin kullanıldığı konusunda bu kadar endişelenmenize gerek yoktur.

Depolama yöneticiniz, performans sorunlarını gözden geçirmek ve çözmek için kurumsal DASD ' yi denetliyor olmalıdır. Kullanılabilirlik için, bir DASD altsisteminde bir günlük kümesi ve farklı bir DASD altsisteminde ikili günlük kullanmak isteyebilirsiniz.

z/OS veri kümesi şifrelemesi ile etkin günlük şifrelemesi

z/OS veri kümesi şifreleme özelliğini, IBM MQ for z/OS 9.1.4 ya da daha sonraki bir yayın düzeylerinde çalışan kuyruk yöneticileri için etkin günlük veri kümelerine uygulayabilirsiniz.

Bu etkin günlük veri kümelerini EXTENDED öznitelikleriyle ayırmanız ve verilerin AES şifrelenmesini sağlayan bir veri kümesi anahtar etiketi ayırmanız gerekir.

[veri kümesi şifrelemesiyle IBM MQ for z/OS üzerinde atıl durumdaki veriler için gizlilik](#), başlıklı bölüme bakın. ek bilgi için.

z/OS IBM MQ ile MetroMirror özelliğinin kullanılması

IBM Metro Mirror, önceden Eşler Arası Zamanuyumlu Eşdüzey Uzaktan Kopyalama (PPRC) olarak bilinir, iki depolama altsistemi arasında zamanuyumlu bir eşleme çözümdür. Burada yazma işlemi tamamlanmadan önce hem birincil hem de ikincil birimlerde yazma işlemleri tamamlanır. Metro Mirror , depolama altsistemi arızası durumunda veri kaybı gerektirmeyen ortamlarda kullanılabilir.

Desteklenen veri kümesi tipleri

Aşağıdaki IBM MQ veri kümesi tiplerinin tümü Metro Mirror kullanılarak eşlenebilir. Ancak, tam olarak hangilerinin eşlenmesi, kuruluşunuzun kullanılabilirlik gereksinimlerine bağlıdır:

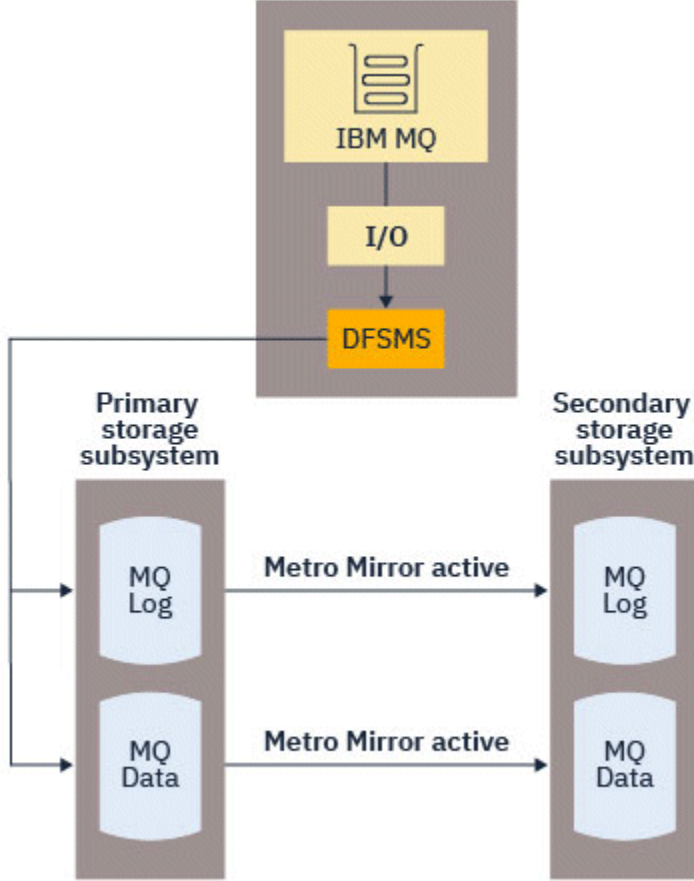
- Etkin günlükler
- Günlükleri arşivle
- Bootstrap veri kümesi (BSDS)
- Sayfa kümeleri
- Paylaşılan ileti veri kümesi (SMDS)

- Yapılandırma için kullanılan veri kümeleri; örneğin, MSTR JCL ' deki CSQINP* DD kartlarında

zHyperYazma özelliğinin IBM MQ etkin günlüklerle kullanılması

Metro Mirror kullanılarak eşlenen bir veri kümesine yazma işlemi yapıldığında, yazma işlemi önce birincil birime yapılır ve sonra ikincil birime kopyalanır. Bu eşleme, depolama altsistemi tarafından yapılır ve yazma işlemini yayınlayan uygulama için saydamdır; örneğin, IBM MQ.

Bu işlem aşağıdaki çizgede gösterilmiştir.

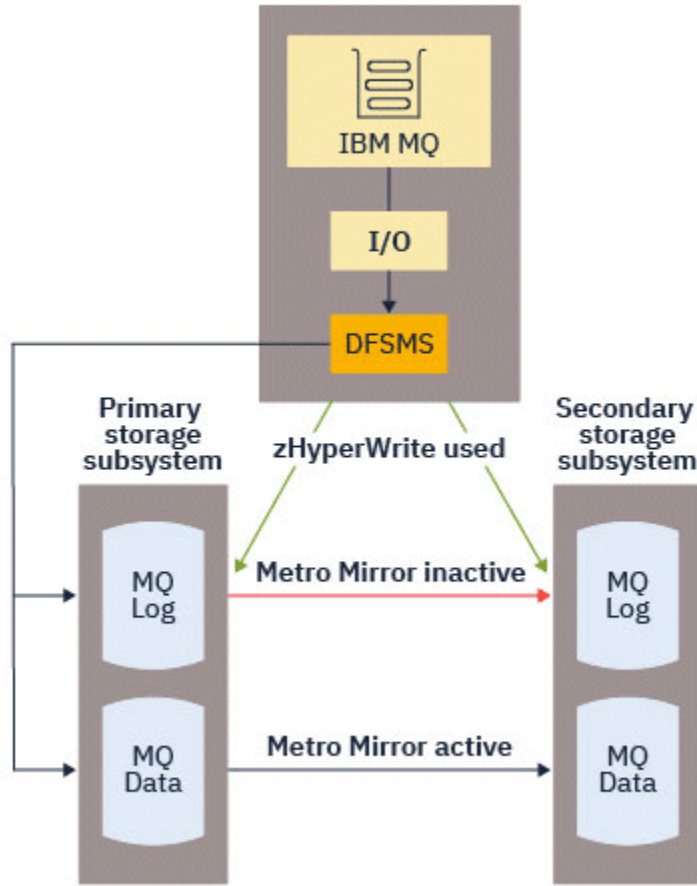


Hem birincil hem de ikincil depolama altsistemlerine yazma işleminin IBM MQ' e geri dönmeden önce tamamlanması gerektiğinden, Metro Mirror kullanımı performans üzerinde bir etki yaratabilir. Bu performans etkisini, Metro Mirror' ı kullanmanın kullanılabilirlik avantajlarına karşı dengelemeniz gerekir.

IBM MQ etkin günlükleri, Metro Mirror işlevinin kullanılmasının başarım üzerindeki etkisine en duyarlıdır. IBM MQ , bu performans etkisini azaltmaya yardımcı olmak için etkin günlüklerle birlikte zHyper yazılmasını sağlar.

zHyperWrite, Metro Mirror kullanılarak eşlenen veri kümelerine yapılan yazma işlemlerinin performans etkisini azaltmak için z/OS ile birlikte çalışan bir depolama altsistemi teknolojisidir. zHyperWrite (Yazma) kullanıldığında, birincil ve ikincil birimlere yazma, depolama altsistemi düzeyinde sıralı olarak değil, Data Facility Storage Management Subsystem (DFSMS) düzeyinde paralel olarak yayınlanır ve böylece performans etkisini azaltır.

Aşağıdaki şekilde, etkin günlükler için kullanılan zHyperYazma ve diğer IBM MQ veri kümesi tipleri için kullanılan Metro Mirror gösterilmektedir. zHyperYazma başarısız olursa, DFSMS ' in Metro Mirror kullanılarak yazma işlemini saydam bir şekilde yeniden yayınladığını unutmayın.



zHyperYazma IBM MQ, yalnızca etkin günlük veri kümelerinde desteklenir.

zHyper' i etkin günlüklerle yazmak için aşağıdakileri yapmanız gerekir:

- IBM MQ ' i zHyperYazma özelliğini kullanacak şekilde yapılandırın ve
- Etkin günlüklerin zHyperYazma yeteneğine sahip birimlerde olması gerekir

IBM MQ ' i aşağıdaki yöntemlerden birini kullanarak zHyperYazma özelliğini kullanacak şekilde yapılandırabilirsiniz:

- Sistem değiştirgesi modülünde ZHYWRITE(YES) değerini belirtin.
- SET LOG ZHYWRITE(YES) komutunu verin.

Etkin günlük veri kümelerinin zHyperYazma yeteneğine sahip birimlerde olması için aşağıdaki koşulları ayarlayın:

- Metro Mirrorbirimlerini etkinleştirin ve birimler zHyperYazma işlemini destekler
- Birimlerin HyperSwap özelliğinin etkinleştirildiğinden emin olun
- IECIOSxx parametresinde HYPERWRITE=YES değerini belirtin

V 9.3.5 IBM MQ 9.3.5öncesinde, önceki tüm koşullar karşılandıysa, zHyperYazma için etkin günlüklere yazma etkinleştirilir. Bu koşullardan biri ya da daha fazlası karşılanmazsa, IBM MQ etkin günlüklere olağan şekilde yazar ve Metro Mirror , yapılandırıldıysa yazma işlemlerini eşler.

V 9.3.5 IBM MQ 9.3.5' den ZHYWRITE (YES) belirtilirse, günlüklerin zHyperYazma yetenekli birimlerde olup olmadığına bakılmaksızın, IBM MQ her zaman etkin günlüklere yazarken zHyperkomutunu kullanmayı dener. Günlükler zHyperYazma yeteneğine sahip birimlerde değilse, Metro Mirror , yapılandırıldıysa yazma işlemlerini eşler. Günlükler zHyperYazma yetenekli birimlerde değilse, zHyper yazma işleminin olumsuz etkileri yoktur.

Notlar:

- IBM MQ , tüm etkin günlük veri kümelerinin zHyperYazma yeteneğine sahip birimlerde olmasını gerektirmez.

IBM MQ , bazı etkin günlük veri kümelerinin zHyperYazma yetenekli birimlerde olduğunu saptarsa ve diğerleri değilse, CSQJ166E iletisini yayınlar ve işleme devam eder.

- IBM MQ , veri kümeleri ilk açıldığında etkin günlük veri kümelerinin zHyperYazma yeteneğine sahip olup olmadığını denetler.

Günlük veri kümeleri, kuyruk yöneticisi başlatılırken ya da DEFINE LOG komutu kullanılarak dinamik olarak eklenirken açılır. Bir kuyruk yöneticisi açık durumdayken günlük veri kümeleri zHyperYazma yeteneğine sahipse, kuyruk yöneticisi yeniden başlatılıncaya kadar bunu algılayamaz.

Yürürlükteki etkin günlük veri kümelerinin zHyperYazma yeteneğine sahip olup olmadığını belirtmek için `DISPLAY LOG` komutunun çıkışını kullanabilirsiniz. Aşağıdaki örnek, her iki veri kümesinin de zHyperYazma yeteneğine sahip olduğunu göstermektedir. Kuyruk yöneticisi ZHYWRITE (YES) ile yapılandırıldıysa, zHyperYazma için bu günlüklere yazma etkinleştirilir:

```
Copy %Full zHyperWrite DSName
1     4  CAPABLE      MQTST.SUBSYS.MQDL.LOGCOPY1.DS001
2     4  CAPABLE      MQTST.SUBSYS.MQDL.LOGCOPY2.DS001
```

Günlük arşivi depolamanın planlanması

Arşiv günlüğü veri kümelerinizi korumanın farklı yollarını anlamak için bu konuyu kullanın.

Arşiv günlüğü veri kümelerini standart etiketli manyetik bantlara ya da DASD ' ye yerleştirebilirsiniz ve bunları veri olanağı sıradüzensel depolama yöneticisine (DFHSM) göre yönetebilirsiniz. Arşiv günlüğü veri kümesindeki her z/OS mantıksal kaydı, etkin günlük veri kümesinden bir VSAM denetim aralığıdır. Blok boyutu 4 KB ' nin bir katı.

Arşiv günlüğü veri kümeleri, IBM MQ tarafından seçilen adlarla dinamik olarak ayrılır. Bu tür ayırmalar için gereken veri kümesi adı öneki, blok boyutu, birim adı ve DASD boyutları sistem parametre modülünde belirtilir. Kuruluş sırasında, IBM MQ ' in arşiv günlüğü veri kümesi adına bir tarih ve saat eklemesini de seçebilirsiniz.

Yeni arşiv günlükleri için IBM MQ ile belirli birimler belirtilemez, ancak bunu yönetmek için Storage Management yordamlarını kullanabilirsiniz. Ayırma hataları oluşursa, boşaltma işlemi bir sonraki boşaltma tetikleninceye kadar ertelenir.

Kuruluş sırasında ikili arşiv günlükleri belirtirseniz, etkin günlükten alınan her günlük denetim aralığı iki arşiv günlüğü veri kümesine yazılır. Arşiv günlüğü veri kümesi çiftinde bulunan günlük kayıtları aynıdır, ancak birim sonu noktaları çoklu birim veri kümeleri için eşitlenmez.

Arşiv günlükleriniz manyetik bantta mı yoksa DASD ' de mi bulunmalı?

Arşiv günlükleriniz için manyetik bant ya da DASD kullanılıp kullanılmayacağına karar verirken göz önünde bulundurmanız gereken bazı etkenler vardır:

- Manyetik bant ya da disk hakkında karar vermeden önce işletim yordamlarınızı gözden geçirin. Örneğin, manyetik bantta arşivlemeyi seçerseniz, gerektiğinde yeterli manyetik bant sürücüsü olması gerekir. Bir olağanüstü durumdan sonra, tüm altsistemler manyetik bant sürücülerini isteyebilir ve beklediğiniz kadar boş manyetik bant sürücüsüne sahip olabilirsiniz.
- Kurtarma sırasında, manyetik bant takıldığı anda manyetik banttaki arşiv günlükleri kullanılabilir. DASD arşivleri kullanıldıysa ve veri kümeleri hiyerarşik depolama yöneticisi (HSM) kullanılarak manyetik banda geçirildiyse, HSM her veri kümesini diske geri çağırırken bir gecikme olur. Arşiv günlüğü kullanılmadan önce veri kümelerini geri çağırabilirsiniz. Ancak, gerekli oldukları doğru sırayı tahmin etmek her zaman mümkün değildir.

- DAsD ' de arşiv günlüklerini kullanırken, çok sayıda günlük gerekiyorsa (yedekten geri yüklendikten sonra bir sayfa kümesi kurtarılırken bu durum olabilir), tüm arşiv günlüklerini tutmak için önemli miktarda DAsD gerekebilir.
- Düşük kullanışlı bir sistemde ya da sına ma sisteminde, manyetik bant bağlama gereksinimini ortadan kaldırmak için DAsD ' de arşiv günlüklerinin olması daha uygun olabilir.
- Her ikisi de RECOVER CFSTRUCT komutu vererek ve kalıcı bir iş birimini geri çekerek, günlüğün geriye doğru okunmasıyla sonuçlanır. Donanım sıkıştırmasına sahip manyetik bant sürücüler i, geriye doğru okuyan işlemlerde kötü performans sergiler. Manyetik bantt an geriye doğru okumayı önlemek için DAsD üzerinde yeterli günlük verisi planlayın.

DAsD ' ye arşivleme, daha hızlı kurtarılabilirlik sunar, ancak manyetik bant da arşivlenmekten daha pahalıdır. İkili günlük kaydı kullanırsanız, arşiv günlüğünün birincil kopyasının DAsD ' ye, ikincil kopyanın da manyetik bant da gitmesini belirtebilirsiniz. Bu, DAsD kullanmadan kurtarma hızını artırır ve manyetik bantı yedekleme olarak kullanabilirsiniz.

Günlüklerinizi manyetik bantt an DAsD ' ye nasıl arşivlediğine ve ters işle mi nasıl gerçekleştirdiğine ilişkin ayrıntılar için bkz. [“Arşiv günlükleri için depolama ortamının değiştirilmesi” sayfa 166](#) .

Manyetik bant da arşivleme

Bir manyetik bant aygıtını arşivlemeyi seçerseniz, IBM MQ en çok 20 birim genişletebilir.

Kümenin bir manyetik bant birimine sığması için etkin günlük veri kümesinin boyutunu değiştirmeyi düşünüyorsanız, BSDS ' nin bir kopyasının etkin günlük veri kümesinin kopyasıyla aynı manyetik bant birimine yerleştirildiğini unutmayın. Manyetik bant birimindeki BSDS için gereken alanı dengelemek için etkin günlük veri kümesinin boyutunu aşağı doğru ayarlayın.

Manyetik bantta ikili arşiv günlükleri kullanıyorsanız, bir kopyanın yerel olarak tutulması ve diğer kopyanın olağanüstü durumdan kurtarmada kullanılmak üzere iş yeri dışında tutulması tipik bir durumdur.

DAsD birimlerine arşivleme

IBM MQ , manyetik bant dışı aygıtlarda (DAsD) ayrılmış tüm arşiv günlüğü veri kümelerini kataloğa getirmenizi gerektirir. DAsD ' de arşivlemeyi seçerseniz, CSQ6ARVP makrosunun CATALOG parametresi YES olmalıdır. Bu parametre NO ise ve DAsD ' ye arşiv günlüğü veri kümeleri yerleştirmeye karar verirsiniz, IBM MQ veri kümesini kataloğa almaya devam etmesine rağmen, her arşiv günlüğü veri kümesi ayrıldığında CSQJ072E iletisi alırsınız.

Arşiv günlüğü veri kümesi DAsD üzerinde tutulursa, arşiv günlüğü veri kümeleri başka bir birime genişletilebilir; çoklu birim desteklenir.

DAsD 'yi kullanmayı seçerseniz, birincil alan ayırmasının (hem miktar hem de blok boyutu) etkin günlük veri kümesinden gelen verileri ya da ilgili BSDS' den (hangisi daha büyükse) içerecek kadar büyük olduğundan emin olun.

Bu, boşaltma işle mi sırasında istenmeyen z/OS X ' B37 ' ya da X ' E37 ' olağ andışı bitiş kodları olasılığını en aza indirir. Birincil alan ayırma, CSQ6ARVP makrosunun PRIQTY (birincil miktar) parametresiyle ayarlanır.

Arşiv günlüğü veri kümeleri büyük ya da genişletilmiş biçimli sıralı veri kümelerinde var olabilir. SMS ACS yordamları artık DSNTYPE (LARGE) ya da DSNTYPE (EXT) kullanıyor.

IBM MQ , arşiv günlüklerinin genişletilmiş biçim veri kümeleri olarak ayrılmasını destekler. Genişletilmiş biçim kullanıldığında, arşiv günlüğü büyüklüğü üst sınırı 65535 izden 4GBetkin günlük büyüklüğü üst sınırına yükseltilir. Arşiv günlükleri, genişletilmiş adres birimlerinin (EAV) genişletilmiş adresleme alanında (EAS) ayrılmaya hak kazanır.

Gerekli donanım ve yazılım düzeyleri varsa, zEDC kullanılarak COMPACTION ile tanımlanan bir veri sınıfına arşiv günlükleri ayrılması, arşiv günlüklerini tutmak için gereken disk saklama alanını azaltabilir. Daha fazla bilgi için bkz. IBM MQ for z/OS: [IBM zEnterprise Data Compression \(zEDC\) ve zEnterprise Data Compression \(zEDC\) ile depolama doluluğu azaltılıyor](#).

z/OS veri kümesi şifreleme özelliği, IBM MQüzerinde çalışan kuyruk yöneticileri için arşiv günlüklerine uygulanabilir. Bu arşiv günlükleri, Otomatik Sınıf Seçimi (ACS) yordamları aracılığıyla, EXTENDED öznitelikleriyle tanımlanmış bir veri sınıfına ve verilerin AES şifreli olmasını sağlayan bir veri kümesi anahtarı etiketine ayrılmalıdır.

Arşiv günlüğü veri kümeleriyle SMS kullanılması

Sisteminizde MVS/DFP depolama yönetimi altsistemi (DFSMS) kuruluysa, arşiv günlüğü veri kümeleriniz için bir Otomatik Sınıf Seçimi (ACS) kullanıcı çıkışı süzgeci yazabilirsiniz; bu süzgeç, bunları SMS ortamı için dönüştürmenize yardımcı olur.

Örneğin, böyle bir süzgeç, çıkışınızı DFSMS ' nin yönetebileceği bir DASD veri kümesine yönlendirebilir. Bu şekilde bir ACS süzgeci kullanıyorsanız dikkatli olmanız gerekir. SMS, DASD veri kümelerinin kataloğa alınmasını gerektirdiğinden, CSQ6ARVP makrosunun CATALOG DATA alanının YES değerini içerdiğini doğrulayın. Yoksa, CSQJ072E iletisi döndürülür; ancak, veri kümesi IBM MQtarafından kataloğa alınmaktadır.

ACS süzgeçleriyle ilgili daha fazla bilgi için DFSMSHsm 'in toplu yedekleme işlemi sırasında dinamik olarak ayırdığı veri kümeleribaşlıklı konuya bakın.

z/OS Arşiv günlükleri için depolama ortamının değiştirilmesi

Arşiv günlükleri tarafından kullanılan depolama ortamını değiştirme yordamı.

Bu görev hakkında

Bu kısımda, arşiv günlükleri için kullanılan depolama ortamının nasıl değiştirileceği açıklanır; örneğin, arşivlemeden manyetik banda arşivlemeye, DASD ' ye geçiş.

Değişiklikleri nasıl yapacağınıza karar vermeniz gerekir:

1. Değişiklikleri yalnızca CSQ6ARVP makrosunu kullanarak yapın; böylece, bu makrolar kuyruk yöneticisi yeniden başlatıldığında uygulanır.
2. CSQ6ARVP makrosunu kullanarak değişiklikleri yapın ve SET ARCHIVE komutunu kullanarak devingen olarak değişiklik yapın. Bu, kuyruk yöneticisinin bir günlük dosyasını bir sonraki arşivlemesinden sonra değişikliklerin geçerli olduğu ve kuyruk yöneticisi yeniden başlatıldıktan sonra kalıcı olduğu anlamına gelir.

Yordam

1. Bu nedenle, arşiv günlükleri manyetik bant yerine DASD ' de saklanır:
 - a) “DASD birimlerine arşivleme” sayfa 165 bölümünü okuyun ve CSQ6ARVP parametrelerini gözden geçirin.
 - b) CSQ6ARVP ' de aşağıdaki parametrelerde değişiklik yapın
 - UNIT ve gerekirse, UNIT2 parametrelerini güncelleyin.
 - DASD için en uygun ayar manyetik banttaki farklı olduğundan BLKSIZE parametresini güncelleyin.
 - PRIQTY ve SECQTY değiştirgelerini, etkin günlüğün ya da BSDS ' nin en büyüğünün tutulması için yeterli büyüklükte olacak şekilde ayarlayın.
 - CATALOG değiştirgesini YES olarak ayarlayın.
 - İstedığınız ALCUNIT ayarının bu olduğunu onaylayın. BLK, aygıt tipinden bağımsız olduğu için kullanılmalıdır.
 - Önceden değilse, ARCWTOR değiştirgesini NO olarak ayarlayın.
2. DASD yerine, arşiv günlüklerinin manyetik bantta saklanacağı şekilde değiştiriliyor:
 - a) “Manyetik banda arşivleme” sayfa 165bölümünü okuyun ve CSQ6ARVP parametrelerini gözden geçirin.
 - b) CSQ6ARVP: içinde aşağıdaki parametrelerde değişiklik yapın:

- UNIT ve gerekirse, UNIT2 parametrelerini güncelleyin.
- En uygun manyetik bant ayarı DASD ' den farklı olduğundan BLKSIZE parametresini güncelleyin.
- İstedığınız ALCUNIT ayarının bu olduğunu onaylayın. BLK, aygıt tipinden bağımsız olduğu için kullanılmalıdır.
- ARCWTOR değiştirgesinin ayarını gözden geçirin.

z/OS Arşiv günlüklerini ne kadar süre tutmam gerekiyor?

Yedekleme stratejinizi planlamanıza yardımcı olması için bu bölümdeki bilgileri kullanın.

USING CSQ6ARVP ya da SET SYSTEM komutundaki ARCRETN değiştirgesini kullanarak, arşiv günlüklerinin gün olarak ne kadar süreyle alıkonacağını belirtirsiniz. Bu dönemden sonra veri kümeleri z/OS'tan silinebilir.

Artık gerekli olmadığı anda arşiv günlüğü veri kümelerini el ile silebilirsiniz.

- Kuyruk yöneticisinin kurtarma için arşiv günlüklerine ihtiyacı olabilir.

Kuyruk yöneticisi, yalnızca BSDS 'deki en son 1000 arşivi tutabilir; arşiv günlükleri BSDS' de değilse, bunlar kurtarma için kullanılamaz ve yalnızca denetim, çözümleme ya da yeniden yürütme amaçlarıyla kullanılır.

- Günlüklerden bilgi alabilmek için arşiv günlüklerini tutmak isteyebilirsiniz. Örneğin, günlükten iletileri alma ve iletiyi hangi kullanıcı kimliğinin koyduğunu ya da aldığı gözden geçirme.

BSDS, günlüklere ve diğer kurtarma bilgilerine ilişkin bilgiler içerir. Bu veri kümesi sabit bir boyuttur. Arşiv günlüğü sayısı CSQ6LOGPiçindeki MAXARCH değerine ulaştığında ya da BSDS dolduğunda, en eski arşiv günlüğü bilgilerinin üzerine yazılır.

BSDS ' den arşiv günlüğü girişlerini kaldırmak için yardımcı programlar vardır, ancak genel olarak BSDS, en eski arşiv günlüğü kaydını kaydırır ve bunların üzerine yazılır.

Arşiv günlüğü gerekli olduğunda

Sayfa kümelerinizi düzenli olarak yedeklemeniz gerekir. Yedekleme sıklığı, bir sayfa kümesinin kaybedilmesi durumunda hangi arşiv günlüklerinin gerekli olduğunu belirler.

CF yapılarınızı düzenli olarak yedeklemeniz gerekir. Yedekleme sıklığı, CF yapısında veri kaybı durumunda hangi arşiv günlüklerinin gerekli olduğunu belirler.

Kurtarma için arşiv günlüğü gerekebilir. Aşağıdaki bilgiler, arşiv günlüğünün ne zaman gerekli olabileceğini ve farklı IBM MQ kaynaklarıyla ilgili sorunların nerede olduğunu açıklar.

Sayfa kümesi kaybı

Sisteminizi yedekten kurtarmanız ve kuyruk yöneticisini yeniden başlatmanız gerekir.

Yedeklemenin alındığı zamanki günlüklerin yanı sıra, yedeklemeden önce en çok üç günlük veri kümesi gerekir.

Tüm LPAR ' lar bir CF yapısına bağlılığı kaybeder ya da yapı kullanılamaz.

Yapıyı kurtarmak için RECOVER CFSTRUCT komutunu kullanın.

Yapı kurtarma işlemi, son yedeklemeden (yedekleme zamanına kadar) sonra yapıya erişen tüm kuyruk yöneticilerinden alınan günlüklerin yanı sıra, yedeklemeyi alan kuyruk yöneticisinin günlüğünde yapının kendisini yedeklemesini gerektirir.

CF yapılarını sık sık yedeklediyseniz, veriler etkin günlüklerde olmalı ve arşiv günlüklerine gerek duymamalısınız.

CF yapısının yeni bir yedeği yoksa, arşiv günlüklerine gerek duyabilirsiniz.

Not: Kalıcı olmayan tüm iletiler kaybedilir; tüm kalıcı iletiler aşağıdaki görevler gerçekleştirilerek yeniden yaratılır:

1. Günlükten son CF yapısı yedeğinin okunması
2. Yapıyı kullanan tüm kuyruk yöneticilerinden günlükler okunuyor

3. Yedeklemeden bu yana güncelleştirmeler birleştiriliyor

Yönetim yapısını yeniden oluşturma

Yönetim yapısını yeniden oluşturmanız gerekirse, bilgiler QSG ' deki her kuyruk yöneticisi için günlüğün son denetim noktasından okunur.

Bir kuyruk yöneticisi etkin değilse, QSG ' deki başka bir kuyruk yöneticisi günlüğü okur.

Arşiv günlüklerine gerek duymamalısınız.

SMDS veri kümesi kaybı

Bir SMDS veri kümesini kaybederseniz ya da veri kümesi bozulursa, veri kümesi kullanılamaz duruma gelir ve durumu FAILED olarak ayarlanır. CF yapısı değişmez.

SMDS veri kümesini geri yüklemek için aşağıdakileri yapmanız gerekir:

1. SMDS veri kümesini yeniden tanımlayın ve
2. [RECOVER CFSTRUCT](#) komutunu vererek CF yapısını kurtarın.

Not: CF yapısındaki tüm kalıcı olmayan iletiler kaybolacak; tüm kalıcı iletiler geri yüklenecek.

Kuyruk yöneticisi günlüklerine ilişkin gereksinme, kullanılmayan bir yapıdan kurtarma gereksinimiyle aynıdır.

z/OS Adreslenebilir günlük aralığı üst sınırını artırmayı planlama

Kuyruk yöneticinizi daha büyük bir günlük görelî bayt adresi (RBA) kullanacak şekilde yapılandırarak, adreslenebilir günlük aralığı üst sınırını artırabilirsiniz.

Günlük RBA boyutu IBM MQ for z/OS 8.0 boyutundan artırıldı. Bu değişikliğe genel bakış için bkz. [Daha Büyük günlük Görelî Bayt Adresi](#).

V 9.3.0 IBM MQ 9.3.0 ya da daha sonraki bir sürümde oluşturulan kuyruk yöneticilerinin varsayılan olarak 8 baytlık günlük RBA ' ları etkindir ve bu nedenle dönüştürme gerektirmez.

Kuyruk yöneticilerinizi istediğiniz zaman 8 baytlık günlük RBA değerlerini kullanacak şekilde dönüştürebilirsiniz. Bir kuyruk paylaşım grubu, 8 baytlık günlük RBA etkinleştirilmiş bazı kuyruk yöneticilerini ve 6 baytlık günlük RBA etkinleştirilmiş bazı kuyruk yöneticilerini içerebilir.

Değişiklik geri alınmıyor

Değişiklik geri alınamaz.

Ne kadar sürer?

Değişiklik için kuyruk yöneticisinin yeniden başlatılması gerekiyor. Yeni veri kümeleri yaratmak için kuyruk yöneticisini durdurun, CSQJUCNV yardımcı programını önyükleme veri kümesi (BSDS) ya da veri kümeleri için çalıştırın, bu önyükleme veri kümelerini yeniden adlandırın ve kuyruk yöneticisini yeniden başlatın. CSQJUCNV yardımcı programının çalışması genellikle birkaç saniye sürer.

Bunun ne gibi bir etkisi var?

- 8 baytlık günlük RBA kullanımıyla, günlük veri kümelerine her veri yazımında ek baytlar vardır. Bu nedenle, kalıcı iletilerden oluşan bir iş yükü için, günlüklere yazılan veri miktarında küçük bir artış vardır.
- Bir sayfa kümesine ya da bağlaşım olanağı (CF) yapısına yazılan veriler bundan etkilenmez.

İlgili görevler

[Daha büyük günlük Görelî Bayt Adresi Uygulanıyor](#)

z/OS Kanal başlatıcısının planlanması

Kanal başlatıcı, kuyruk yöneticileri arasında iletişim sağlar ve kendi adres alanında çalışır.

İki tip bağlantı vardır:

1. Bir ağ üzerinden bir kuyruk yöneticisine uygulama bağlantıları. Bunlar müşteri kanalları olarak bilinir.
2. Kuyruk yöneticisinden kuyruk yöneticisi bağlantılarına. Bunlar MCA kanalları olarak bilinir.

Dinleyiciler

Kanal dinleyici programı, gelen ağ isteklerini dinler ve kanal gerektiğinde uygun kanalı başlatır. Gelen bağlantıları işlemek için kanal başlatıcının en az bir IBM MQ dinleyici görevinin yapılandırılması gerekir. Bir dinleyici TCP dinleyicisi ya da LU 6.2 dinleyicisi olabilir.

Her dinleyici bir TCP kapısı ya da LU adı gerektirir.

Her kanal başlatıcısı için birden çok dinleyicinizin olabileceğine dikkat edin.

TCP/IP

Bir kanal başlatıcı, aynı z/OS görüntüsünde birden çok TCP yığını ile çalışabilir. Örneğin, bir TCP yığını iç bağlantılar için, başka bir TCP yığını dış bağlantılar için olabilir.

Bir çıkış kanalı tanımladığınızda:

1. Bağlantının hedef anasistemini ve kapısını ayarladınız. Bu aşağıdakilerden biri olabilir:

- IP adresi, örneğin 10.20.4.6
- anasistem adı, örneğin mvs-prod.myorg.com

Hedefi belirlemek için bir anasistem adı kullanırsanız, IBM MQ, hedefin IP adresini çözümlmek için Etki Alanı Ad Sistemi 'ni (DNS) kullanır.

2. Birden çok TCP yığını kullanıyorsanız, kullanılacak IP Yığını adresini belirleyen kanal tanımında **LOCLADDR** parametresini belirleyebilirsiniz.

Yüksek kullanılabilirlikli bir DNS sunucusuna ya da DNS sunucularına sahip olmayı planlamanız gerekir. DNS kullanılmıyorsa, giden kanallar başlatılamayabilir ve bir anasistem adı kullanılarak gelen bir bağlantıyı eşleyen kanal doğrulama kuralları işlenemez.

APPC ve LU 6.2

APPC kullanıyorsanız, kanal başlatıcının APPC ' de bir LU adı ve yapılanışı olması gerekir.

Kuyruk paylaşım grupları

Tek bir sistem görüntüsü sağlamak ve gelen IBM MQ bağlantı isteğinin kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisine gitmesini sağlamak için bazı yapılandırmaları gerçekleştirmeniz gerekir. Örneğin:

1. Bir donanım ağ yönlendiricisi. Bu yönlendirici, işletme tarafından görülen bir IP adresine sahiptir ve ilk isteği, bu donanıma bağlı herhangi bir kuyruk yöneticisine yönlendirebilir.
2. Bir Sanal IP adresi (VIPA). Kurumsal çapta bir IP adresi belirlenir ve bu adres, sistem şebekesinde bulunan TCP yığınlarından herhangi birine yönlendirilebilir. Daha sonra TCP yığını, sistem şebekesinde herhangi bir dinleme kuyruğu yöneticisine yönlendirebilir.

IBM MQ trafiğini koruma

Aktarım kanalındaki verileri korumak için TLS bağlantılarını kullanmak üzere IBM MQ ' i yapılandırabilirsiniz. TLS kullanmak için dijital sertifikaları ve anahtar halkalarını kullanmanız gerekir.

Uyumlu IBM MQ tanımlarına ve uyumlu sertifikalarına sahip olduğunuzdan emin olmak için kanalın uzak ucundaki personelle de çalışmanız gerekir.

Hangi bağlantıların IBM MQ ' e ve kullanıcı kimliğine bağlanabileceğini denetleyebilirsiniz.

- IP adresi
- İstemci kullanıcı kimliği

- Uzak kuyruk yöneticisi ya da
- Dijital sertifika (bkz. [Kanal Kimlik Doğrulama Kayıtları](#))

Ayrıca, istemci uygulamalarının geçerli bir kullanıcı kimliği ve parola sağladığından emin olarak istemci uygulamalarını kısıtlayabilirsiniz (bkz. [Bağlantı Doğrulaması](#)).

Kanal başlatıcıyı çalıştırabilir ve her kanalı TLS kullanacak şekilde yapılandırabilirsiniz.

Kanal başlatıcının izlenmesi

Kanal başlatıcısına ve kanallara ilişkin bilgi veren MQSC komutları vardır:

- [DISPLAY CHINIT](#) komutu, kanal başlatıcısına ve etkin dinleyicilere ilişkin bilgi verir.
- [DISPLAY CHSTATUS](#) komutu, bir kanalın etkinliğini ve durumunu görüntüler.

Kanal başlatıcı, kanal başlatıcı görevleri ve kanal etkinliği hakkında bilgi içeren SMF kayıtları da üretebilir. Ek bilgi için bkz. [“Kanal başlatıcı SMF verilerinin planlanması” sayfa 171](#) .

Kanal başlatıcı, kanallar başlatılıp durdurulduğunda iş günlüğüne ileti yayar. Kuruluşunuzda otomasyon, durumu yakalamak için bu iletileri kullanabilir. Bazı kanallar sadece birkaç saniye aktif olduğundan, birçok mesaj üretilebilir. Bu iletileri z/OS ileti işleme olanağını kullanarak ya da **EXCLMSG** ayarını [SET SYSTEM](#) komutuyla yaparak gizleyebilirsiniz.

IBM MQ kanal tanımlarınızın yapılandırılması

Birbirine bağlı çok sayıda kuyruk yöneticiniz varsa, tüm nesne tanımlamalarını yönetmek zor olabilir. IBM MQ kümelemenin kullanılması bunu basitleştirebilir.

İki kuyruk yöneticisini tam havuz olarak belirtirsiniz. Diğer kuyruk yöneticilerinin havuzlardan biriyle bir bağlantı kurmaları gerekir. Diğer kuyruk yöneticileriyle bağlantı gerektiğinde, kuyruk yöneticisi kanalları otomatik olarak yaratır ve başlatır.

Bir kümede çok sayıda kuyruk yöneticisi bulunmasını planlıyorsanız, özel olarak ayrılmış havuz görevi yapan ve uygulama trafiği olmayan kuyruk yöneticilerine sahip olmayı planlamalısınız.

Ek bilgi için bkz. [“Dağıtılmış kuyruklarınızın ve kümelerinizin planlanması” sayfa 19](#) .

Kanal başlatıcısını yapılandırmadan önceki işlemler

1. TCP/IP ya da APPC kullanılıp kullanılmayacağına karar verin.
2. TCP kullanıyorsanız, IBM MQ için en az bir kapı ayırın.
3. Bir DNS sunucusuna gereksinim duyarsanız, sunucuyu yüksek kullanılabilirlikli olacak şekilde yapılandırın.
4. APPC kullanıyorsanız, bir LU adı ayırın ve APPC ' yi yapılandırın.

Kanal başlatıcısını yapılandırdıktan sonra, üretime geçmeden önce yapılacak işlemler

1. Sahip olacağınız bağlantıları planlayın:
 - a. Uzak uygulamalardan istemci bağlantıları.
 - b. Diğer kuyruk yöneticilerine ve diğer kuyruk yöneticilerinden MCA kanalları. Genellikle, her bir uzak kuyruk yöneticisine/yöneticisinden bir kanalınız vardır.
2. Kümelemeyi ayarlayın ya da var olan bir kümeleme ortamına katılın.
3. Kanal başlatıcısının önünde kullanılabilirlik için birden çok TCP yığını, VIPA ' yı ya da bir dış yönlendiriciyi kullanmanız gerekip gerekmediğini göz önünde bulundurun.
4. TLS kullanmayı planlıyorsanız:
 - a. Anahtarlığı ayarla

- b. Sertifikaları ayarla
5. Kanal kimlik doğrulamasını kullanmayı planlıyorsanız:
 - a. Gelen oturumları MCA kullanıcı kimlikleriyle eşlemeye ilişkin ölçütlere karar verin
 - b. **REVDNS** kuyruk yöneticisi parametresini ayarlayarak ters DNS aramasını etkinleştir
 - c. Güvenliği gözden geçirin. Örneğin, varsayılan kanalları silin ve bir kanal için **MCAUSER** özniteliğinde yalnızca gerekli yetkiye sahip kullanıcı kimliklerini belirtin.
6. Kanal başlatıcısı tarafından üretilen muhasebe ve istatistik SMF kayıtlarını yakalayın ve bunları işleme gönderin.
7. İş günlüğü iletilerinin izlenmesini otomatikleştirin.
8. Gerekliyse, verimi artırmak için ağ ortamınızı ayarlayın. TCP ile, büyük gönderme ve alma arabellekleri verimi artırır. MQ ' yu aşağıdaki komutları kullanarak belirli TCP arabellek boyutlarını kullanmaya zorlayabilirsiniz:

```
RECOVER QMGR(TUNE CHINTCPRBDYNSZ nnnnn)  
RECOVER QMGR(TUNE CHINTCPSBDYNSZ nnnnn)
```

kanallar için SO_RCVBUF ve SO_SNDBUF değerlerini nnnnniçinde belirlenen bayt cinsinden büyüklüğe ayarlar.

İlgili kavramlar

“Kuyruk yöneticinizin planlanması” sayfa 141

Bir kuyruk yöneticisi ayarlarken, kuyruk yöneticisinin işletmenizin gereksinimlerini karşılaması için planlamanız kuyruk yöneticisinin büyümesine izin vermelidir.

Kanal başlatıcı SMF verilerinin planlanması

Kanal başlatıcı için SMF verilerini toplama uygulamasını planlamanız gerekir.

Kanal başlatıcı iki tip kayıt üretir:

- Kanal başlatıcısı ve içindeki görevlerle ilgili bilgileri içeren istatistik verileri.
- DISPLAY CHSTATUS komutuna benzer bilgileri içeren kanal muhasebe verileri.

Aşağıdaki komutu kullanarak istatistik verilerini toplamaya başlarsınız:

```
START TRACE(STAT) CLASS(4)
```

ve şu komutu kullanarak durdurun:

```
STOP TRACE(STAT) CLASS(4)
```

Aşağıdaki komutu kullanarak muhasebe verilerini toplamaya başlarsınız:

```
START TRACE(ACCTG) CLASS(4)
```

ve şu komutu kullanarak durdurun:

```
STOP TRACE(ACCTG) CLASS(4)
```

Kanal tanımlamasında ya da kuyruk yöneticisinde **STATCHL** özniteliğini kullanmak için hangi kanalların muhasebe verilerinin toplandığını denetleyebilirsiniz.

- İstemci kanalları için **STATCHL** kuyruk yöneticisi düzeyinde ayarlanmalıdır.
- Otomatik olarak tanımlanan küme gönderen kanalları için, muhasebe verilerinin toplanmasını **STATACLS** kuyruk yöneticisi özniteliğiyle denetleyebilirsiniz.

Kuyruk yöneticisi için **STATCHL** varsayılan değeri OFF' tur. Kanal muhasebe verilerini toplamak için, sınıf 4 muhasebe izlemesini başlatmanın yanı sıra, kuyruk yöneticisindeki ya da kanal tanımındaki **STATCHL** değerini varsayılan değerden değiştirmeniz gerekir.

SMF kayıtları aşağıdaki durumda üretilir:

- **V9.3.0** IBM MQ for z/OS 9.3.0 ' den başlayarak, CSQ6SYSP **STATIME** ya da **ACCTIME** parametreleriyle gösterilen zaman aralığı geçmiştir; ya da **STATIME** ya da **ACCTIME** SMF veri toplama yayınında sıfırsa. Kanal başlatıcı ve kuyruk yöneticisi için SMF verilerini toplama istekleri eşitlenir.
- Bir STOP TRACE (ACCTG) CLASS(4) ya da STOP TRACE (STAT) CLASS(4) komutu yayınlandı ya da
- Kanal başlatıcı kapatıldı. Bu noktada, herhangi bir SMF verisi yazılır.

Bir kanal SMF aralığı sırasında durursa, SMF işleminin bir sonraki çalışmasında muhasebe verileri SMF ' ye yazılır. Bir istemci bağlanırsa, bir miktar iş yapar ve bağlantıyı keser, daha sonra yeniden bağlantı kurar ve bağlantıyı keserse, üretilen iki kanal muhasebe verisi kümesi vardır.

İstatistik verileri olağan olarak bir SMF kaydına sığar, ancak çok sayıda görev kullanılıyorsa birden çok SMF kaydı oluşturulabilir.

Muhasebe verileri, etkinleştirildiği her kanal için toplanır ve genellikle bir SMF kaydına sığar. Ancak çok sayıda kanal etkinse birden çok SMF kaydı oluşturulabilir.

Kanal başlatıcı SMF verilerini toplama maliyeti azdır. Genellikle CPU kullanımındaki artış birkaç yüzde altındadır ve genellikle ölçüm hatası içindedir.

Bu işlevi kullanmadan önce, SMF ' nin ek kayıtlar için kapasiteye sahip olduğundan ve SMF kayıtlarını yeni SMF verilerini içerecek şekilde çıkarma işlemlerini değiştirdiğinden emin olmak için z/OS sistem programcınızla çalışmanız gerekir.

Kanal başlatıcı istatistikleri verileri için SMF kayıt tipi 115 ve alt tip 231 'dir.

Kanal başlatıcı muhasebe verileri için SMF kayıt tipi 116 ve alt tip 10 'dur.

Bu verileri işlemek için kendi programlarınızı yazabilir ya da verileri yazdırmak için MQSMF programını içeren SupportPac MP1B programını kullanabilir ve bir dağıtma yaprağına içe aktarma için uygun CSV (Virgülle Ayrılmış Değerler) biçiminde veri oluşturabilirsiniz.

Kanal başlatıcısı SMF verilerinin yakalanmasıyla ilgili sorunlarla karşılaşıyorsanız, daha fazla bilgi için [Kanal başlatıcısı \(CHINIT\) için SMF verilerinin yakalanması sırasında sorunlarla baş etme](#) başlıklı konuya bakın.

İlgili görevler

[IBM MQ performans istatistiklerinin yorumlanması](#)

[Kanal muhasebe verileriyle ilgili sorunların giderilmesi](#)

z/OS z/OS TCP/IP ortamınızın planlanması

Ağınız üzerinden en iyi verimi almak için, 64 KB ya da daha büyük büyüklükteki TCP/IP gönderme ve alma arabelleklerini kullanmanız gerekir. Bu büyüklükle, sistem arabellek boyutlarını eniyiler.

[Yüksek Gecikme Süresi Ağları için Dinamik Doğru Boyutlandırma Nedir?](#) başlıklı konuya bakın. ek bilgi için.

Aşağıdaki Netstat komutunu kullanarak sistem arabelleğinizi denetleyebilirsiniz:

```
TSO NETSTAT ALL (CLIENT csq1CHIN
```

Sonuçlar, aşağıdaki iki değer de içinde olmak üzere çok fazla bilgi görüntüler:

```
ReceiveBufferSize: 0000065536  
SendBufferSize: 0000065536
```

65536, 64 KB 'dir. Arabellek büyüklükleriniz 65536 'dan azsa, TCPIP yordamındaki PROFILE DDName içindeki **TCPSENDBFRSIZE** ve **TCPRCVBUFRSIZE** değerlerini artırmak için ağ ekibinizle birlikte çalışmanız gerekir. Örneğin, aşağıdaki komutu kullanabilirsiniz:

```
TCPCONFIG TCPSENDBFRSIZE 65536 TCPRCVBUFRSIZE 65536
```

Sistem genelinde **TCPSENDBFRSIZE** ya da **TCPRCVBUFRSIZE** ayarlarınızı değiştiremiyorsanız, IBM Software Support merkezimize başvurun.

z/OS

Kuyruk paylaşım grubunuz (QSG) planlanması

Paylaşılan bir kuyruğa alma ortamını gerçekleştirmenin en kolay yolu, bir kuyruk yöneticisi yapılandırmak, bu kuyruk yöneticisini bir QSG 'ye eklemek ve QSG 'ye diğer kuyruk yöneticilerini eklemektir.

Bir kuyruk paylaşım grubu, yapılandırma bilgilerini saklamak için Db2 tablolarını kullanır. Aynı Db2 veri paylaşım grubunu paylaşan tüm QSG'liler tarafından kullanılan bir tablo kümesi vardır.

Paylaşılan kuyruk iletileri, bir bağlaşım olanağındaki (CF) bir yapıda saklanır. Her QSG 'nin kendi CF yapıları kümesi vardır. Gereksinimlerinizi karşılayacak yapıları yapılandırmanız gerekir.

63KB boyutundaki iletiler CF 'de saklanamaz. Bu iletiler için Paylaşılan İleti Veri Kümelerini (SMDS) ya da Db2 kullanmanız gerekir.

İleti profilleri ve kapasite planlaması

Paylaşılan kuyruk iletilerinizin ileti tanımını anlamanız gerekir. Aşağıda göz önünde bulundurmanız gereken etmenlere ilişkin örnekler verilmiştir:

- Ortalama ve ileti boyutu üst sınırı
- Tipik kuyruk derinliği ve kural dışı durum kuyruğu derinliği. Örneğin, bir gün boyunca iletileri tutmak için yeterli kapasiteye sahip olmak gerekebilir ve tipik kuyruk derinliği 100 iletinin altındadır.

İleti profili değişirse, daha sonraki bir tarihte yapıların boyutunu artırabilir ya da SMDS uygulayabilirsiniz.

Büyük hacimli iletileri işleyebilmek istiyorsanız, yapı kullanımını kullanıcı tarafından belirtilen eşiklere ulaştığında IBM MQ 'i iletileri SMDS'ye boşaltacak şekilde yapılandırabilirsiniz.

CF yapılarını çift yönlü yapmak isteyip istemediğinize karar vermeniz gerekir. Bu, CFRM ilkesindeki CF yapısı tanımını tarafından denetlenir:

1. Çift yönlü bir yapı iki bağlantı tesisi kullanır. Bir CF ile ilgili bir sorun varsa, hizmette herhangi bir kesinti yoktur ve yapı, varsa, üçüncü bir CF 'de yeniden oluşturulabilir. Çift yönlü yapılar, paylaşılan kuyruklardaki işlemlerin performansını önemli ölçüde etkileyebilir.
2. Yapı çift yönlü değilse, CF ile ilgili bir sorun, yapı başka bir CF 'de yeniden oluşturuluncaya kadar o CF 'deki yapılarda paylaşılan kuyrukların kullanılmayacağı anlamına gelir.

IBM MQ , bu durumda başka bir CF 'deki yapıları otomatik olarak yeniden oluşturacak şekilde yapılandırılabilir. Kalıcı iletiler, kuyruk yöneticilerinin günlüklerinden kurtarılır.

CF tanımlamalarını değiştirmenin kolay olduğunu unutmayın.

Yalnızca kalıcı olmayan iletileri tutabilmesi ya da kalıcı ve kalıcı olmayan iletileri tutabilmesi için bir yapı tanımlayabilirsiniz.

Kalıcı iletileri tutabilen yapıların düzenli aralıklarla yedeklenmesi gerekir. Bir arıza durumunda yapıyı kurtarmak için gereken süreyi en az her saat başı yedeklemek için CF yapılarınızı yedekin. Yedekleme, yedekleme işlemi gerçekleştiren kuyruk yöneticisinin günlük verileri kümesinde saklanır.

Paylaşılan kuyruklarınızda yüksek bir ileti verimi olmasını bekliyorsanız, CF yapılarını yedeklemek için özel olarak ayrılmış bir kuyruk yöneticisinin olması en iyi uygulamadır. Bu, kuyruk yöneticisi günlüklerinden daha az veri okunması gerektiği için yapıları kurtarmak için gereken süreyi kısaltır.

Kanallar

IBM MQ QSG ' ye bağlanan uygulamalar için tek bir sistem görüntüsü sağlamak üzere paylaşılan giriş kanalları tanımlayabilirsiniz. Bunlar kuruluysa, kuyruk paylaşım grubu ortamına gelen bir bağlantı, QSG ' deki herhangi bir kuyruk yöneticisine gidebilir.

Bu kanallar için bir ağ yönlendiricisi ya da Sanal IP adresi (VIPA) ayarlamamız gerekebilir.

Paylaşılan çıkış kanallarını tanımlayabilirsiniz. Paylaşılan çıkış kanalı eşgörünümü, QSG ' deki herhangi bir kuyruk yöneticisinden başlatılabilir.

Ek bilgi için [Paylaşılan kanallar](#) konusuna bakın.

Durumu

IBM MQ kaynaklarını bir dış güvenlik yöneticisi kullanarak koruyabilirsiniz. RACF kullanıyorsanız, RACF tanımlarının başına kuyruk yöneticisi adı eklenir. Örneğin, APPLICATION.INPUT , qmqgrName . APPLICATION . INPUT . adlı MQQUEUE sınıfındaki bir tanım kullanılarak korunur.

Bir kuyruk paylaşım grubunu kullanırken, kuyruk yöneticisi adı önekli tanımlarla kaynakları korumaya devam edebilir ya da tanımların başına kuyruk paylaşım grubu adını koyabilirsiniz. Örneğin, qsgName . APPLICATION . INPUT .

Kuyruk paylaşım grubu adıyla profiller önekini kullanmayı hedeflemelisiniz; bu, tüm kuyruk yöneticileri için tek bir tanımlama olduğu, çalışmanızı kaydettiği ve kuyruk yöneticileri arasında tanımlarda uyumsuzluğun önleneyeceği anlamına gelir.

İlgili kavramlar

[“Kuyruk yöneticinizin planlanması” sayfa 141](#)

Bir kuyruk yöneticisi ayarlarken, kuyruk yöneticisinin işletmenizin gereksinimlerini karşılaması için planlamanız kuyruk yöneticisinin büyümesine izin vermelidir.

Bağlantı tesisinizi planlama ve depolama ortamını boşaltma

İlk boyutları ve bağlaşım olanağı (CF) yapılarınızın biçimlerini ve paylaşılan ileti veri kümesi (SMDS) ortamını ya da Db2 ortamını planlarken bu konuyu kullanın.

Bu bölüm, aşağıdaki konularla ilgili bilgi içerir:

- [“Bağlaşım olanağı kaynaklarının tanımlanması” sayfa 174](#)
 - [Boşaltma depolama düzeneğinin kararlaştırılması](#)
 - [Yapılarınızın planlanması](#)
 - [Yapılarınızın boyutunun planlanması](#)
 - [Paylaşılan kuyrukların yapılara eşlenmesi](#)
- [“Paylaşılan ileti veri kümesi \(SMDS\) ortamınızın planlanması” sayfa 180](#)
- [“Db2 ortamınızın planlanması” sayfa 183](#)

Bağlaşım olanağı kaynaklarının tanımlanması

Paylaşılan kuyrukları kullanmayı amaçlıyorsanız, IBM MQ ' un CFRM ilkenizde kullanacağı bağlaşım olanağı yapılarını tanımlamanız gerekir. Bunu yapmak için önce CFRM ilkenizi yapılarla ilgili bilgilerle güncelleniz ve daha sonra ilkeyi etkinleştirmeniz gerekir.

Kuruluşunuz, kullanılabilir bağlaşım olanaklarını açıklayan bir CFRM ilkesine sahip olabilir. Denetim verileri yardımcı programı , sağladığınız metinli deyimlere dayalı olarak ilkenin içeriğini değiştirmek için kullanılır. Yeni yapıların adlarını, tanımlandıkları bağlaşım olanaklarını ve yapıların boyutunu tanımlayan ilkeye deyimler eklemeniz gerekir.

CFRM ilkesi, IBM MQ yapılarının çift yönlü olup olmadığını ve bunların hata senaryolarında nasıl yeniden tahsis edileceğini de belirler. [Paylaşılan kuyruk kurtarma](#) , bağlaşım olanağını etkileyen hatalara dayanıklılık için CFRM ' nin yapılandırılmasına ilişkin öneriler içerir.

Boşaltma depolama ortamınızı belirleme

Paylaşılan kuyruklara ilişkin ileti verileri, bağlaşım olanağından boşaltılabilir ve Db2 çizelgesinde ya da *paylaşılan ileti veri kümesi* (SMDS) adı verilen IBM MQ yönetilen veri kümesinde saklanabilir. Bağlaşım tesisinde (yani 63 KB ' den büyük) depolanamayacak kadar büyük olan iletiler her zaman boşaltılmalıdır ve daha küçük iletiler isteğe bağlı olarak, bağlaşım olanağı alan kullanımını azaltmak için boşaltılabilir.

Daha fazla bilgi için [Paylaşılan iletiler için boşaltma seçeneklerini belirtme](#) başlıklı konuya bakın.

Yapılarınızın planlanması

Bir kuyruk paylaşım grubu (QSG), en az iki yapının tanımlanmasını gerektirir. Yönetim yapısı olarak bilinen ilk yapı, kuyruk paylaşım grubu genelinde IBM MQ iç etkinliğini koordine etmek için kullanılır. Bu yapıda kullanıcı verileri tutulmaz. Değişmez adı *qsg-nameCSQ_ADMIN* (burada *qsg-name* kuyruk paylaşım grubunuzun adıdır). Sonraki yapılar uygulama yapıları olarak bilinir ve iletileri IBM MQ paylaşılan kuyruklarında tutmak için kullanılır. Her yapı en çok 512 paylaşılan kuyruk tutabilir.

Sistem kuyrukları için *qsg-nameCSQSYSAPPL* adlı bir uygulama yapısı kullanılır. Bu yapının tanımlanması isteğe bağlıdır, ancak belirli özelliklerin kullanılması için gereklidir. Varsayılan olarak, SYSTEM.QSG.CHANNEL.SYNCQ ve SYSTEM.QSG.UR.RESOLUTION.QUEUE kuyrukları, *qsg-nameCSQSYAPPL* yapısında tanımlanır.

Birden çok yapı kullanma

Bir kuyruk paylaşım grubu, en çok 64 bağlaşım olanağı yapısına bağlanabilir. Bu yapılardan biri yönetim yapısı olmalıdır. Tanımlanmışsa, bu yapılardan biri *qsg-nameCSQSYSAPPL* yapısı olabilir. İleti verileri için en çok 63 (*qsg-nameCSQSYSAPPL* tanımlandıysa 62) yapı kullanabilirsiniz. Aşağıdaki nedenlerden ötürü birden çok uygulama yapısı kullanmayı seçebilirsiniz:

- Büyük olasılıkla çok sayıda ileti bulunduracak kuyruklarınız var ve bu nedenle tüm bağlaşım olanağının tüm kaynaklarını gerektirir.
- Çok sayıda paylaşılan kuyruk için bir gereksiniminiz vardır, bu nedenle her bir yapı yalnızca 512 kuyruk içerebileceği için bu kuyruklar birden çok yapıya bölünmelidir.
- RMF, bir yapının kullanım karakteristiğine ilişkin raporlar, içerdiği kuyrukları bir dizi bağlaşım olanağı arasında dağıtmanızı önerir.
- Veri yalıtma nedeniyle, bazı kuyruk verilerinin diğer kuyruk verilerinden fiziksel olarak farklı bir bağlaşım olanağı içinde tutulmasını istiyorsunuz.
- Kalıcı paylaşılan iletilerin kurtarılması, yapı düzeyinde öznitelikler ve komutlar (örneğin, BACKUP CFSTRUCT) kullanılarak gerçekleştirilir. Yedekleme ve kurtarmayı basitleştirmek için, kalıcı olmayan iletileri tutan kuyrukları, kalıcı iletileri tutan yapılardan farklı yapılara atayabilirsiniz.

Yapıların hangi bağlantı tesislerinde ayrılması gerektiğini seçerken, aşağıdaki noktaları göz önünde bulundurun:

- Veri yalıtma gereksinimleriniz.
- Bağlantı tesisinin uçuculuğu (yani, bir güç kesintisi yoluyla verileri koruma yeteneği).
- Erişim sistemleri ve bağlaşım tesisi arasındaki ya da bağlaşım tesisleri arasındaki arızanın bağımsızlığı.
- Bağlaşım olanağına kurulan bağlaşım olanağı denetim kodu (CFCC) düzeyi (IBM MQ için Düzey 9 ya da üstü gerekir).

Yapılarınızın boyutunun planlanması

Yönetim yapısı

Denetim yapısı (*qsg-name*CSQ_ADMIN), kuyruk paylaşım grubundaki her kuyruk yöneticisi için 1000 liste girişi içerecek büyüklükte olmalıdır. Bir kuyruk yöneticisi başlatıldığında, kuyruk paylaşım grubuna *tanımlı* kuyruk yöneticisi sayısı için yeterli büyüklükte olup olmadığını görmek için yapı denetlenir. Kuyruk yöneticileri, CSQ5PQSG yardımcı programı tarafından eklendiyse, kuyruk paylaşım grubuna tanımlandıkları varsayılır. MQSC DISPLAY GROUP komutuyla gruba tanımlanan kuyruk yöneticilerini denetleyebilirsiniz.

Not: Yapının boyutunu hesaplarken, kuyruk paylaşım grubundaki kuyruk yöneticilerinin sayısına ek olarak büyük iş birimlerinin boyutuna da izin vermelisiniz.

Çizelge 22 sayfa 176 içinde, kuyruk paylaşım grubunda tanımlanan çeşitli sayıda kuyruk yöneticisi için yönetim yapısı için gerekli minimum boyut gösterilir. Bu boyutlar CFCC seviye 14 bağlaşım tesisi yapısı için oluşturulmuştur; CFCC ' nin daha yüksek seviyeleri için muhtemelen daha büyük olması gerekir.

Çizelge 22. Yönetim yapısı boyutu alt sınırı	
Kuyruk paylaşım grubunda tanımlı kuyruk yöneticilerinin sayısı	Gerekli depolama alanı
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
15.000	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

Çizelge 22. Yönetim yapısı boyutu alt sınırı (devamı var)	
Kuyruk paylaşım grubunda tanımlı kuyruk yöneticilerinin sayısı	Gerekli depolama alanı
24	25856 KB
25	27136 KB
26	27904 KB
27	28672 KB
28	29696 KB
29	30464 KB
30	31232 KB
31	32256 KB

Var olan bir kuyruk paylaşım grubuna bir kuyruk yöneticisi eklediğinizde, depolama gereksinimi [Çizelge 22 sayfa 176](#) içinde önerilen boyutu aşmış olabilir. Bu durumda, *qsg-nameCSQ_ADMIN* yapısına ilişkin gerekli saklama alanını tahmin etmek için aşağıdaki yordamı kullanın:

1. Kuyruk paylaşım grubunun var olan bir üyesinde **DISPLAY CFSTATUS(CSQ_ADMIN)** MQSC komutunu verin.
2. CSQ_ADMIN yapısına ilişkin ENTSMAX bilgilerini alın.
3. Bu sayı, kuyruk paylaşım grubunda tanımlamak istediğiniz kuyruk yöneticilerinin toplam sayısının 1000 katından azsa, yapı boyutunu artırın.

Uygulama yapıları

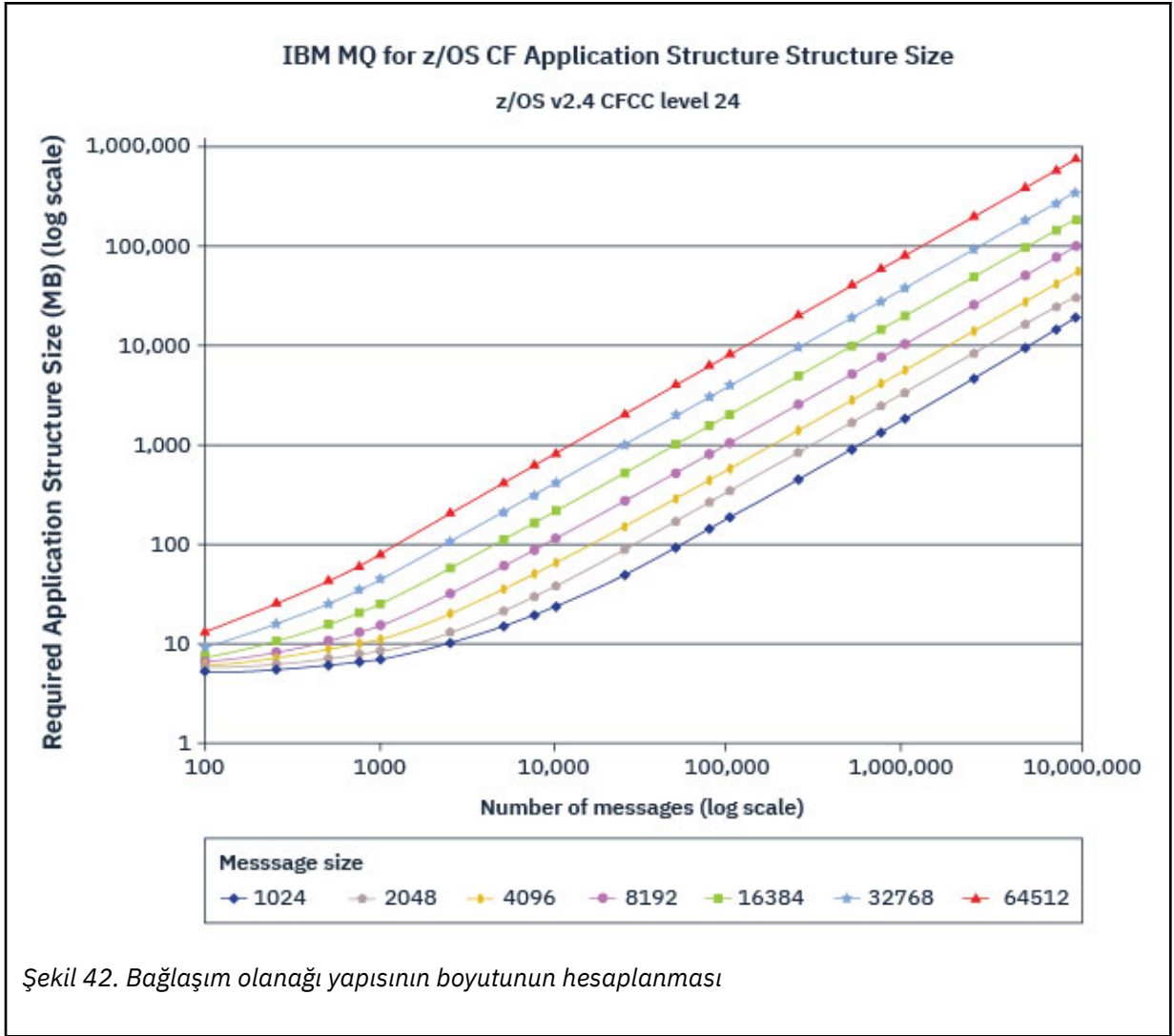
IBM MQ iletilerini tutmak için gereken uygulama yapılarının boyutu, bir yapıda aynı anda tutulacak iletilerin olası sayısına ve boyutuna bağlıdır.

[Şekil 42 sayfa 178](#) içindeki grafik, iletileri paylaşılan kuyruklarınızda tutmak için CF yapılarınızı ne kadar büyük yapmanız gerektiğini gösterir. Ayırma boyutunu hesaplamak için aşağıdaki bilgilere gereksinim duyarsınız:

- Kuyruklarınızdaki iletilerin ortalama boyutu.
- Yapıda saklanacak iletilerin toplam sayısı.

Yatay eksen boyunca iletilerin sayısını bulun. İleti boyutunuza karşılık gelen eğriyi seçin ve dikey eksenden gerekli değeri belirleyin. Örneğin, uzunluğu 1 KB olan 200 000 ileti için 256-512 MB aralığında bir değer verilir.

[Çizelge 23 sayfa 178](#) , aynı bilgileri tablo biçiminde sağlar.



Bağlaşım olanağı yapılarınızı ne kadar büyük yapacağınızı hesaplamaya yardımcı olmak için bu tabloyu kullanın:

Çizelge 23. Bağlaşım olanağı yapısının boyutunun hesaplanması

İleti sayısı	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

CFRM ilkeniz aşağıdaki deyimleri içermelidir:

- INITSIZE , ilk kuyruk yöneticisi bağlandığında yapının ayrıldığı KB cinsinden boyuttur.
- SIZE , yapının ulaşabileceği büyüklük üst sınırıdır.
- FULLTHRESHOLD , z/OS ' in IXC585E iletisini yayınladığı eşğin yüzde değerini, yapının dolduğunu belirtmek için belirler.

En iyi uygulama, INITSIZE ve SIZE ' in 2 kat içinde olmasını sağlamaktır. Örneğin, önceden belirlenmiş şekiller ile aşağıdaki deyimleri ekleyebilirsiniz:

```
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)
```

Yapı kullanımı uyarı iletilerinin verildiği eşığe ulaşırsa, müdahale gerekir. IBM MQ , uygulamaların daha fazla ileti yazmasını önlemek, kuyruklardan ileti almak için daha fazla uygulama başlatmak ya da kuyruğa ileti koyan bazı uygulamaları susturmak için yapıdaki bazı kuyruklarda MQPUT işlemlerini engellemek için kullanılabilir.

Diğer bir seçenek olarak, z/OS olanaklarını kullanarak yapının büyüklüğünü değiştirebilirsiniz. Aşağıdaki z/OS komutu:

```
SETXCF START,ALTER,STRNAME=structure-name,SIZE=newsize
```

Yapının boyutunu *newsize* olarak değiştirir; burada *newsize* , yapıya ilişkin CFRM ilkesinde belirtilen SIZE değerinden küçük, ancak yürürlükteki bağlaşım olanağı boyutundan büyük bir değerdir.

MQSC [DISPLAY CFSTATUS](#) komutuyla bir bağlaşım olanağı yapısının kullanımını izleyebilirsiniz.

Herhangi bir işlem yapılmazsa ve bir kuyruk yapısı doldurulursa, uygulamaya bir MQRC_STORAGE_MEDIUM_FULL dönüş kodu döndürülür. Yönetim yapısı dolursa, tam belirtiler hangi süreçlerin hatayla karşılaştıklarına bağlıdır, ancak bunlar aşağıdaki sorunları içerebilir:

- Komutlara yanıt yok.
- Kesinleştirme işlemi sırasında ortaya çıkan sorunların sonucu olarak kuyruk yöneticisi hatası oluştu.

CSQSYSAPPL yapısı

*qsg-name*CSQSYSAPPL yapısı, sistem kuyruklarına ilişkin bir uygulama yapısıdır. Çizelge 3 , *qsg-name*CSQSYSAPPL yapısında tanımlanan varsayılan kuyruklara ilişkin ileti veri büyüklüklerinin nasıl tahmin edileceğine ilişkin bir örnek gösterir.

Çizelge 24. Boyutlandırmaya karşı CSQSYSAPPL kullanımını gösteren çizelge.	
<i>qsg-name</i> CSQSYSAPPL kullanımı	Boyutlandırma
SYSTEM.QSG.CHANNEL.SYNCQ	Paylaşılan bir kanalın etkin eşgörünümü başına 500 baytlık 2 ileti
SYSTEM.QSG.UR.RESOLUTION.QUEUE	2 KB ' lik 1000 ileti

Önerilen ilk yapı tanımı değerleri şunlardır:

```
STRUCTURE NAME(qsg-nameCSQSYSAPPL)
INITSIZE(20480) /* 20 MB */
SIZE(30720) /* 30 MB */
FULLTHRESHOLD(85)
```

Bu değerler, paylaşılan kanalları ve grup kurtarma birimlerini kullanımınıza bağlı olarak ayarlanabilir.

Paylaşılan kuyrukların yapılar eşlenmesi

IBM MQ için bir uygulama yapısı tanımlamak üzere `DEFINE CFSTRUCT` komutunu kullanın. IBM MQ için bir yapı tanımladığınızda, yapı adına QSG adı örneğini eklemeyin. For example, to define an application structure to IBM MQ that has the name `qsg-adtAPPLICATION1` in the CFRM policy, issue the following command:

```
DEFINE CFSTRUCT(APPLICATION1)
```

Kuyruk tanımının CFSTRUCT özneliği, kuyruğu bir yapıyla eşlemek için kullanılır. Bu öznelikte QSG adı öneki olmadan CF yapısının adını belirtin. Örneğin, aşağıdaki komut APPLICATION1 yapısında paylaşılan bir kuyruğu tanımlar:

```
DEFINE QLOCAL(myqueue) QSGDISP(SHARED) CFSTRUCT(APPLICATION1)
```

Paylaşılan ileti veri kümesi (SMDS) ortamınızın planlanması

SMDS boşaltma ile kuyruk paylaşım gruplarını kullanıyorsanız, IBM MQ 'in bir grup paylaşılan ileti veri kümesine bağlanması gerekir. IBM MQ ileti verilerini depolamak için gereken yapılandırmanın ve veri kümesi gereksinimlerinin anlaşılmasına yardımcı olması için bu konuyu kullanın.

Paylaşılan ileti veri kümesi (SMDS anahtar sözcüğüyle açıklanır), bir bağlaşım olanağı yapısında saklanan paylaşılan iletilere ilişkin boşaltılmış ileti verilerini saklamak için kuyruk yöneticisi tarafından kullanılan bir veri kümesidir.

Not: Bir yapı için SMDS veri kümelerini tanımlarken, her kuyruk yöneticisi için bir tane olmalıdır.

Bu veri boşaltma biçimi etkinleştirildiğinde, **CFSTRUCT** , kuyruk paylaşım grubundaki her kuyruk yöneticisi için bir veri kümesi olacak şekilde, ilişkili bir paylaşılan ileti veri kümesi grubunu gerektirir. Paylaşılan ileti veri kümeleri grubu, **CFSTRUCT** tanımlamasındaki **DSGROUP** parametresi kullanılarak IBM MQ olarak tanımlanır. Kullanılacak arabelleklerin sayısı ve veri kümeleri için genişletme öznelikleri gibi isteğe bağlı daha fazla bilgi sağlamak için ek parametreler kullanılabilir.

Her kuyruk yöneticisi, sahip olduğu veri kümesine yazabilir, bu kuyruk yöneticisi aracılığıyla yazılan iletilere ilişkin paylaşılan ileti verilerini saklayabilir ve gruptaki tüm veri kümelerini okuyabilir.

Yapıyla ilişkili her veri kümesinin durumunu ve özneliklerini açıklayan bir liste, **CFSTRUCT** tanımlamasının bir parçası olarak içeride tutulur; böylece her kuyruk yöneticisi, kullanılabilir veri kümelerini bulmak için tanımlamayı denetleyebilir.

Bu veri kümesi bilgileri, geçerli durumu ve kullanılabilirliği görüntülemek için **DISPLAY CFSTATUS TYPE(SMDS)** komutu ve belirtilen **CFSTRUCT** ile ilişkili veri kümelerine ilişkin parametre ayarlarını görüntülemek için **DISPLAY SMDS** komutu kullanılarak görüntülenebilir.

Tek tek paylaşılan ileti veri kümeleri, sahip olan kuyruk yöneticisi adının (genellikle **SMDS** anahtar sözcüğü kullanılarak belirtilir) ve **CFSTRUCT** yapı adının birleşimiyle etkili bir şekilde tanımlanır.

Bu bölümde aşağıdaki konular açıklanmaktadır:

- [DSGROUP parametresi](#)
- [DSBLOCK parametresi](#)
- [Paylaşılan ileti veri kümesi özellikleri](#)
- [Paylaşılan ileti veri kümesi alanı yönetimi](#)
- [Paylaşılan ileti veri kümelerine erişim](#)
- [Paylaşılan ileti veri kümesi yaratılması](#)
- [Paylaşılan ileti veri kümesi başarımı ve kapasitesiyle ilgili önemli noktalar](#)
- [Paylaşılan ileti veri kümesinin etkinleştirilmesi](#)

Bu parametrelerin ayrıntıları için [DEFINE CFSTRUCT](#) başlıklı konuya bakın.

Paylaşılan ileti veri kümelerinizi yönetmeye ilişkin bilgi için [Paylaşılan ileti veri kümelerini yönetme](#) başlıklı konuya bakın.

DSGROUP parametresi

CFSTRUCT tanımlamasındaki **DSGROUP** parametresi, o yapıya ilişkin büyük iletilerin saklanacağı veri kümeleri grubunu tanımlar. Alan ayırma amacıyla kullanılacak mantıksal blok boyutunu ve arabellek havuzu büyüklüğü ve otomatik veri kümesi genişletme seçeneklerine ilişkin değerleri belirtmek için ek parametreler kullanılabilir.

Veri kümelerine boşaltma etkinleştirilmeden önce **DSGROUP** parametresi ayarlanmalıdır.

- **CFLEVEL (5)** adresinde yeni bir **CFSTRUCT** tanımlanıyorsa ve **OFFLOAD (SMDS)** seçeneği belirtilirse ya da varsayıldıysa, **DSGROUP** parametresi aynı komutta belirtilmelidir.
- Var olan bir **CFSTRUCT** , **CFLEVEL** değerini **CFLEVEL (5)** olarak artıracak şekilde değiştiriliyorsa ve **OFFLOAD (SMDS)** seçeneği belirtilirse ya da varsayıldıysa, **DSGROUP** parametresi önceden ayarlanmamışsa aynı komutta belirtilmelidir.

DSBLOCK parametresi

Her veri kümesi içindeki alan, **CFSTRUCT** tanımlamasında **DSBLOCK** parametresi kullanılarak belirtilen sabit büyüklükteki (genellikle 256 KB) mantıksal bloklar olarak kuyruklara ayrılır, daha sonra her bir mantıksal blok içinde 4 KB ' lik sayfa aralıkları olarak tek tek iletilere ayrılır (fiziksel blok boyutu ve denetim aralığı boyutuna karşılık gelir). Mantıksal blok büyüklüğü, tek bir G/Ç işleminde okunabilecek ya da yazılabilecek ileti verisi miktarı üst sınırını da belirler; bu, SMDS arabellek havuzu arabellek büyüklüğüyle aynıdır.

DSBLOCK parametresinin daha büyük bir değeri, ayrı G/Ç işlemlerinin sayısını azaltarak çok büyük iletilerin başarımını artırabilir. Ancak, daha küçük bir değer, her etkin istek için gereken arabellek miktarını azaltır. **DSBLOCK** değiştirgesinin varsayılan değeri 256 KB ' dir; bu, bu gereksinimler arasında makul bir bakiye sağlar; bu nedenle, bu değiştirgenin belirtilmesi olağan durumda gerekli olmayabilir.

Paylaşılan ileti veri kümesi özellikleri

Paylaşılan ileti veri kümesi, bir VSAM doğrusal veri kümesi (LDS) olarak tanımlanır. Boşaltılan her ileti, veri kümesindeki bir ya da daha fazla blokta saklanır. Saklanan veriler, genişletilmiş bir sanal saklama alanı biçimi gibi, bağlaşım olanağı girişlerindeki bilgilerle doğrudan adreslenir. Veri kümesinin kendisinde saklanan ayrı bir dizin ya da benzer bir denetim bilgisi yoktur.

Doğrudan adresleme şeması, bir bloğa sığan iletiler için bloğu okumak ya da yazmak için yalnızca tek bir G/Ç işleminin gerektiği anlamına gelir. Bir ileti birden çok bloğa yayıldığında, yeterli arabelleklerin kullanılabilir olması koşuluyla, her bloğa ilişkin G/Ç işlemleri, geçen süreyi en aza indirmek için tam olarak çakışabilir.

Paylaşılan ileti veri kümesi, kurtarma ve yeniden başlatma durumu bilgilerini içeren birinci sayfadaki bir üstbilgiden ve kuyruk yöneticisi normal sonlandırmasında boş blok alanı haritasını kaydetmek için kullanılan bir alan haritası denetim noktası alanından oluşan küçük bir miktar genel denetim bilgisi de içerir.

Paylaşılan ileti veri kümesi alanı yönetimi

Kapasite, performans ve işletim konularına ilişkin arka plan bilgileri olarak, paylaşılan ileti veri kümelerindeki alanın kuyruk yöneticileri tarafından nasıl yönetildiğine ilişkin kavramların anlaşılması yararlı olabilir.

Her bir paylaşılan ileti veri kümesindeki boş alan, her bir mantıksal blok içinde kullanılmakta olan sayfaların sayısını gösteren bir alan eşlemi kullanılarak sahip olan kuyruk yöneticisi tarafından izlenir. Alan eşlemi, veri kümesi açıkken ve olağan bir şekilde kapatıldığında veri kümesine kaydedilirken ana depoda tutulur. (Kurtarma durumlarında alan haritası, hangi veri kümesi sayfalarının kullanıldığını öğrenmek için bağlaşım olanağı yapısındaki iletiler taranarak otomatik olarak yeniden oluşturulur).

Boşaltılmış ileti verileriyle paylaşılan bir ileti yazılırken, kuyruk yöneticisi her ileti öbeği için bir sayfa aralığı ayırır. Belirtilen kuyruk için kısmen kullanılan bir mantıksal blok varsa, kuyruk yöneticisi o bloktaki sonraki boş sayfadan başlayarak yer ayırır; tersi durumda yeni bir mantıksal blok ayırır. İletinin tamamı

yürürlükteki mantıksal öbeğe sığmazsa, kuyruk yöneticisi mantıksal öbeğin sonunda ileti verilerini böler ve sonraki ileti öbeği için yeni bir mantıksal öbek ayırır. Bu, iletinin tamamı için alan ayrılıncaya kadar yinelenir. Son mantıksal bloktaki kullanılmayan alanlar, kuyruk için yeni yürürlükteki mantıksal blok olarak saklanır. Veri kümesi olağan bir şekilde kapatıldığında, yürürlükteki mantıksal bloklardaki kullanılmayan sayfalar saklanmadan önce alan eşlemine döndürülür.

Boşaltılmış ileti verileri içeren paylaşılan bir ileti okunduğunda ve silinmeye hazır olduğunda, kuyruk yöneticisi, ileti için bağlaşım olanağı girişini sahip kuyruk yöneticisi tarafından izlenen bir temizleme listesine (aynı kuyruk yöneticisi olabilir) aktararak silme isteğini işler. Bu listeye giriş geldiğinde, sahip olan kuyruk yöneticisi girişleri okur ve siler ve serbest bırakılan sayfa aralıklarını alan eşlemine döndürür. Bir mantıksal bloktaki tüm kullanılan sayfalar serbest bırakıldığında, blok yeniden kullanılabilir duruma gelir.

Paylaşılan ileti veri kümelerine erişim

Her paylaşılan ileti veri kümesi, kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından erişilebilen paylaşılan doğrudan erişim depolamasında olmalıdır.

Olağan çalışma sırasında, her kuyruk yöneticisi okuma/yazma erişimi için kendi paylaşılan ileti veri kümesini açar ve diğer kuyruk yöneticileri için etkin paylaşılan ileti veri kümelerini salt okunur erişim için açar; böylece, bu kuyruk yöneticileri tarafından saklanan iletileri okuyabilir. Bu, her kuyruk yöneticisi kullanıcı kimliğinin kendi paylaşılan ileti veri kümesine en az UPDATE erişimi ve yapı için diğer tüm paylaşılan ileti veri kümelerine READ erişimi gerektirdiği anlamına gelir.

Paylaşılan ileti veri kümelerinin **RECOVER CFSTRUCT** kullanılarak kurtarılması gerekiyorsa, kurtarma işlemi kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisinden yürütülebilir. Kurtarma işlemi gerçekleştirilmek için kullanılacak bir kuyruk yöneticisi, kurtarılması gerekebilecek tüm veri kümelerine UPDATE erişimi gerektirir.

Paylaşılan ileti veri kümesi yaratılması

Her paylaşılan ileti veri kümesi, normal olarak, bu ileti boşaltma biçiminin kullanımını etkinleştirmek için ilgili **CFSTRUCT** tanımlaması yaratılmadan ya da değiştirilmeden önce yaratılmalıdır; çünkü **CFSTRUCT** tanımlaması değişiklikleri olağan durumda hemen yürürlüğe girer ve bir kuyruk yöneticisi o yapıya atanmış paylaşılan bir kuyruğa erişmeye çalışır çalışmaz veri kümesi gerekir. Paylaşılan ileti veri kümesini ayırmak ve önceden biçimlemek için örnek bir iş SCSQPROC (CSQ4SMDS) içinde sağlanır. İş, OFFLOAD (SMDS) ile CFSTRUCT kullanılan her kuyruk yöneticisi için bir paylaşılan ileti kümesi ayrılacak şekilde uyarlanmalı ve çalıştırılmalıdır.

Kuyruk yöneticisi offload desteğinin etkinleştirildiğini bulursa ve paylaşılan ileti veri kümesini açmaya çalışırsa, ancak henüz yaratılmamışsa, paylaşılan ileti veri kümesi kullanılamaz olarak işaretlenir. Daha sonra kuyruk yöneticisi, veri kümesi yaratılıncaya ve kuyruk yöneticisine yeniden denemesi için bildirim gönderilinceye (örneğin, **START SMDSCONN** komutu kullanılarak) kadar büyük iletileri saklayamaz.

Paylaşılan ileti veri kümesi, bir Erişim Yöntemi Hizmetleri **DEFINE CLUSTER** komutu kullanılarak VSAM doğrusal veri kümesi olarak oluşturulur. Tanımlama, bir kuyruk yöneticisinin bir kuyruk yöneticisini yazma erişimi için açmasına ve herhangi bir sayıda kuyruk yöneticisinin bunu aynı anda okumasına izin vermek için **SHAREOPTIONS(2 3)** belirtmelidir. Varsayılan denetim aralığı büyüklüğü olan 4 KB kullanılmalıdır. Veri kümesinin 4 GB ' den fazla genişletilmesi gerekebilirse, VSAM genişletilmiş adreslenebilirlik özneliğine sahip bir SMS veri sınıfı kullanılarak tanımlanması gerekir. Paylaşılan ileti veri kümesi, genişletilmiş adres birimlerinin (EAV) genişletilmiş adresleme alanı (EAS) bölümünde bulunmaya uygundur.

Her paylaşılan ileti veri kümesi, ilk kullanımından önce boş ya da ikili sınıflara (**CSQJUFMT** kullanılarak) ya da SCSQPROC (CSQ4SMDS gibi benzer bir yardımcı program kullanılarak) önceden biçimlendirilmiş olabilir. Boşsa ya da açıldığında kısmen biçimlenmişse, kuyruk yöneticisi kalan alanı otomatik olarak ikili sınıflara biçimlendirir.

Paylaşılan ileti verileri kümesi başarımı ve kapasitesiyle ilgili önemli noktalar

Her paylaşılan ileti veri kümesi, aynı sistem içindeki bölgelerden, sahip kuyruk yöneticisi tarafından ilişkili **CFSTRUCT** ' e yazılan paylaşılan iletilere ilişkin boşaltılmış verileri saklamak için kullanılır. Boşaltılan her

ileti, giriş için 256 bayttan ve üstbilgi ve tanımlayıcının iki ögesi için 512 bayttan oluşan 768 bayta kadar CF depolamasını alır. Boşaltılan her ileti, veri kümesindeki bir ya da daha çok sayfada (4 KB boyutundaki fiziksel bloklar) saklanır.

Belirli bir sayıda boşaltılmış ileti için gereken veri kümesi alanı, genel ileti boyutu (tanımlayıcı dahil) 4 KB 'nin sonraki katına yuvarlanıp daha sonra ileti sayısıyla çarpılarak tahmin edilebilir.

Bir sayfa kümesine gelince, paylaşılan bir ileti veri kümesi neredeyse dolduğunda, isteğe bağlı olarak otomatik olarak genişletilebilir. Bu otomatik genişletme için varsayılan davranış, **CFSTRUCT** tanımlamasındaki **DSEXPAND** parametresi kullanılarak ayarlanabilir. Bu ayar, **ALTER SMDS** komutundaki **DSEXPAND** parametresi kullanılarak her kuyruk yöneticisi için geçersiz kılınabilir. Veri kümesi %90 tam alana ulaştığında ve daha fazla alana gereksinim duyulduğunda otomatik genişletme tetiklenir. Veri kümesi tanımlandığında ikincil alan ayırması belirtilmediği için genişletmeye izin verilir, ancak genişletme girişimi VSAM tarafından reddedilirse, veri kümesinin yürürlükteki boyutunun %20 'si ikincil ayırma kullanılarak genişletme yeniden denir.

Paylaşılan ileti veri kümesinin genişletilmiş adreslenirlik özneliğiyle tanımlanması koşuluyla, büyüklük üst sınırı yalnızca VSAM ile sınırlanır ve en çok 16 TB ya da 59 birim olur. Bu, bir yerel sayfa kümesinin maksimum 64 GB boyutundan önemli ölçüde daha büyüktür.

Paylaşılan ileti veri kümesini etkinleştirme

Bir kuyruk yöneticisi bir uygulama bağlaşım olanağı yapısına başarıyla bağlandığında, o yapı tanımlamasının ilişkili bir **DSGROUP** parametresi kullanarak boşaltmayı belirtip belirtmediğini denetler. Bu durumda, kuyruk yöneticisi yazma erişimi için kendi paylaşılan ileti veri kümesini ayırır ve açar; daha sonra, diğer kuyruk yöneticilerinin sahip olduğu var olan paylaşılan ileti verileri kümelerine okuma erişimi için açılır.

Paylaşılan bir ileti veri kümesi ilk kez açıldığında (kuyruk paylaşım grubu içinde etkin olarak kaydedilmeden önce), ilk sayfa henüz geçerli bir üstbilgi içermez. Kuyruk yöneticisi, kuyruk paylaşım grubunu, yapı adını ve sahip olan kuyruk yöneticisini tanımlamak için üstbilgi bilgilerini doldurur.

Üstbilgi tamamlandıktan sonra, kuyruk yöneticisi yeni paylaşılan ileti veri kümesini etkin olarak kaydeder ve diğer etkin kuyruk yöneticilerine yeni veri kümesi hakkında bildirim göndermek için bir olay yayınlar.

Bir kuyruk yöneticisi paylaşılan ileti veri kümesini her açtığında, doğru veri kümesinin kullanılmaya devam edildiğinden ve zarar görmediğinden emin olmak için üstbilgi bilgilerini doğrular.

Db2 ortamınızın planlanması

Kuyruk paylaşım gruplarını kullanıyorsanız, IBM MQ 'in bir veri paylaşım grubunun üyesi olan bir Db2 altsistemine bağlanması gerekir. IBM MQ verilerini tutmak için kullanılan Db2 gereksinimlerini anlamanıza yardımcı olması için bu konuyu kullanın.

IBM MQ 'in bağlanacağı veri paylaşım grubunun adını ve bu veri paylaşım grubuna erişmek için bağlanılacak Db2 altsisteminin (ya da Db2 grubunun) adını bilmesi gerekir. Bu adlar, CSQ6SYSP sistem parametresi makrosunun QSGDATA parametresinde belirtilir ([CSQ6SYSP' nin Kullanılması](#) başlıklı konuda açıklanmıştır).

Veri paylaşım grubu içinde, paylaşılan Db2 tabloları şunları tutmak için kullanılır:

- Kuyruk paylaşım grubuna ilişkin yapılandırma bilgileri.
- IBM MQ paylaşılan ve grup nesnelerinin özellikleri.
- İsteğe bağlı olarak, boşaltılan IBM MQ iletileriyle ilgili veriler.

IBM MQ , gerekli Db2 çizelge alanlarını, çizelgeleri ve dizinleri tanımlamak için tek bir örnek iş kümesi sağlar. Bu işler, UTS 'yi (Universal Table Spaces; Evrensel Çizelge Alanları) kullanır. Ürünün önceki sürümlerinde biri UTS için olmak üzere iki iş kümesi ve diğeri Db2' in en son sürümleri tarafından kullanımdan kaldırılan daha eski çizelge alanı tipleri için vardı.

IBM MQ , daha eski çizelge alanı tipleriyle kullanılabilir ve önceden varolan bir kuyruk paylaşım grubunuz varsa bu uygun olabilir. Ancak, yeni bir kuyruk paylaşım grubu oluşturuyorsanız, bu grup UTS kullanmalıdır.

Db2 V12 İşlev düzeyi 508 , çok çizelgeli çizelge alanlarını evrensel çizelge alanlarına geçirmek için kesintiye neden olmayan bir geçiş işlemi sağlar. Bu yaklaşımı, var olan kuyruk paylaşım grupları tarafından kullanılan çok çizelgeli çizelge alanlarını, tüm kuyruk paylaşım grubunda kesinti olmadan evrensel çizelge alanlarına geçirmek için kullanabilirsiniz.

Db2 V13' te ALTER TABLESPACE deyiminin MOVE TABLE seçeneğini kullanın. Ek bilgi için [Çizelgelerin çok çizelgeli çizelge alanlarından büyümeye göre bölümlere taşınması](#) başlıklı konuya bakın.

Varsayılan olarak Db2 , Db2 kaynaklarının sahibi olarak işleri çalıştıran kişinin kullanıcı kimliğini kullanır. Bu kullanıcı kimliği silinirse, ilişkili kaynaklar silinir ve çizelge silinir. Tek bir kullanıcı kimliği yerine tabloların sahibi olmak için bir grup kimliği kullanmayı düşünün. Bunu, JOB kartına GROUP=groupname ekleyerek ve herhangi bir SQL deyiminden önce SET CURRENT SQLID= 'groupname' değerini belirterek yapabilirsiniz.

IBM MQ , Db2RRS Attach olanağını kullanır. Bu, bağlanmak istediğiniz Db2 grubunun adını belirtebileceğiniz anlamına gelir. Bir Db2 grup bağlantısı adına (belirli bir Db2 altsistemine değil) bağlanmanın yararı, IBM MQ ' un o grubun üyesi olan z/OS görüntüsündeki herhangi bir kullanılabilir Db2 altsistemine bağlanmasıdır (ya da yeniden bağlanabilir). Bir kuyruk paylaşımı IBM MQ altsistemini çalıştıracığınız her z/OS görüntüsünde etkin veri paylaşım grubunun üyesi olan bir Db2 altsistemi olmalı ve RRS etkin olmalıdır.

Db2 depolama

Çoğu kuruluş için, gerekli Db2 depolama alanı miktarı 3390 aygıtında yaklaşık 20 ya da 30 silindiridir. Ancak, depolama gereksiniminizi hesaplamak istiyorsanız, aşağıdaki çizelgede Db2 ' in IBM MQ verileri için ne kadar depolama alanı gerektirdiğini belirlemenize yardımcı olacak bazı bilgiler verilmiştir. Çizelge, her Db2 satırının uzunluğunu ve her bir satırın ilgili Db2 çizelgesine ne zaman eklendiğini ya da çizelgeden ne zaman silindiğini açıklar. Bu bilgileri, *z/OS Kuruluş Kılavuzu için Db2* içindeki Db2 çizelgeleri ve dizinleri için alan gereksinimlerini hesaplamaya ilişkin bilgilerle birlikte kullanın.

<i>Çizelge 25. Db2 depolama gereksinimlerinizin planlanması</i>			
Db2 Çizelge Adı	Satırın uzunluğu	Aşağıdaki durumda bir satır eklenir:	Bu durumda bir satır silinir:
CSQ.ADMIN_B_QSG	252 bayt	Çizelgeye, CSQ5PQSG yardımcı programının ADD QSG işleviyle bir kuyruk paylaşım grubu eklenir.	CSQ5PQSG yardımcı programının REMOVE QSG işleviyle bir kuyruk paylaşım grubu çizelgeden kaldırılır. (Kuyruk paylaşım grubu kaydı silindiğinde, bu kuyruk paylaşım grubuyla ilgili tüm satırlar diğer tüm Db2 tablolarından otomatik olarak silinir.)
CSQ.ADMIN_B_QMGR	En çok 3828 bayt	Çizelgeye CSQ5PQSG yardımcı programının ADD QMGR işleviyle bir kuyruk yöneticisi eklenir.	CSQ5PQSG yardımcı programının REMOVE QMGR işleviyle bir kuyruk yöneticisi çizelgeden kaldırılır.
CSQ.ADMIN_B_STRUCTURE	1454 bayt	Kuyruk paylaşım grubu içinde önceden bilinmeyen bir yapıya ad veren QSGDISP (SHARED) özneliğini belirten ilk yerel kuyruk tanımlaması tanımlanır.	Kuyruk paylaşım grubu içindeki bir yapıya ad veren QSGDISP (SHARED) özneliğini belirten son yerel kuyruk tanımlaması silinir.
CSQ.ADMIN_B_SCST	342 bayt	Paylaşılan bir kanal başlatıldı.	Paylaşılan bir kanal devre dışı olur.

Çizelge 25. Db2 depolama gereksinimlerinizin planlanması (devamı var)

Db2 Çizelge Adı	Satırın uzunluğu	Aşağıdaki durumda bir satır eklenir:	Bu durumda bir satır silinir:
CSQ.ADMIN_B_SSKT	254 bayt	NPMSPEED (NORMAL) özniteliğine sahip paylaşılan bir kanal başlatıldı.	NPMSPEED (NORMAL) özniteliğine sahip paylaşılan bir kanal devre dışı olur.
CSQ.ADMIN_B_STRBACKUP	514 bayt	CSQ.ADMIN_B_STRUCTURE tablosu. BACKUP CFSTRUCT komutu çalıştırılana kadar her giriş işlevsiz bir giriştir ve bu da kukla girişlerin üzerine yazılır.	CSQ.ADMIN_B_STRUCTURE tablosu.
CSQ.OBJ_B_AUTHINFO	3400 bayt	QSGDISP (GROUP) ile bir kimlik doğrulama bilgileri nesnesi tanımlandı.	QSGDISP (GROUP) ile bir kimlik doğrulama bilgileri nesnesi silinir.
CSQ.OBJ_B_QUEUE	En çok 3707 bayt	<ul style="list-style-type: none"> QSGDISP (GROUP) özniteliğine sahip bir kuyruk tanımlandı. QSGDISP (SHARED) özniteliğine sahip bir kuyruk tanımlandı. DEFTYPE (SHAREDYN) özniteliğine sahip bir model kuyruğu açıldı. 	<ul style="list-style-type: none"> QSGDISP (GROUP) özniteliğine sahip bir kuyruk silindi. QSGDISP (SHARED) özniteliğine sahip bir kuyruk silindi. DEFTYPE (SHAREDYN) özniteliğine sahip dinamik bir kuyruk DELETE seçeneğiyle kapatıldı.
CSQ.OBJ_B_NAMELIST	En çok 15127 bayt	QSGDISP (GROUP) özniteliğine sahip bir ad listesi tanımlandı.	QSGDISP (GROUP) özniteliğine sahip bir ad listesi silindi.
CSQ.OBJ_B_CHANNEL	En çok 14127 bayt	QSGDISP (GROUP) özniteliğine sahip bir kanal tanımlandı.	QSGDISP (GROUP) özniteliğine sahip bir kanal silindi.
CSQ.OBJ_B_STGCLASS	En çok 2865 bayt	QSGDISP (GROUP) özniteliğine sahip bir depolama sınıfı tanımlandı.	QSGDISP (GROUP) öznitelik sınıfına sahip bir depolama sınıfı silinir.
CSQ.OBJ_B_PROCESS	En çok 3347 bayt	QSGDISP (GROUP) özniteliğine sahip bir işlem tanımlandı.	QSGDISP (GROUP) özniteliğine sahip bir işlem silinir.
CSQ.OBJ_B_TOPIC	En çok 14520 bayt	QSGDISP (GROUP) özniteliğine sahip bir konu nesnesi tanımlandı.	QSGDISP (GROUP) özniteliğine sahip bir konu nesnesi silindi.
CSQ.EXTEND_B_QMGR	430 baytta n az	Çizelgeye CSQ5PQSG yardımcı programının ADD QMGR işleviyle bir kuyruk yöneticisi eklenir.	CSQ5PQSG yardımcı programının REMOVE QMGR işleviyle bir kuyruk yöneticisi çizelgeden kaldırılır.
CSQ.ADMIN_B_MESSAGES	87 bayt	Büyük ileti PUT için (BLOB başına 1).	Büyük ileti için GET (BLOB başına 1).

Çizelge 25. Db2 depolama gereksinimlerinizin planlanması (devamı var)			
Db2 Çizelge Adı	Satırın uzunluğu	Aşağıdaki durumda bir satır eklenir:	Bu durumda bir satır silinir:
CSQ.ADMIN_MSGS_BAUX1 CSQ.ADMIN_MSGS_BAUX2 CSQ.ADMIN_MSGS_BAUX3 CSQ.ADMIN_MSGS_BAUX4		Bu 4 çizelge, iletinin her BLOB için bu 4 çizelgenin birine eklenen büyük iletilere ilişkin ileti bilgi yükünü içerir. BLOBS en çok 511 KB uzunluğundadır, bu nedenle ileti boyutu > 711 KB ise, bu ileti için birden çok BLOB olacaktır.	

63 KB ' den büyük paylaşılan kuyruk iletilerinin çok sayıda kullanılması, IBM MQ sisteminizde önemli başarımlar etkileri olabilir. Daha fazla bilgi için SupportPac MP16, Capacity Planning and Tuning for IBM MQ for z/OS başlıklı konuya bakın: [SupportPacs for IBM MQ and other project areas](#).

z/OS Yedekleme ve kurtarma planlaması

Sitenizde yedekleme ve kurtarma yordamlarının geliştirilmesi, yüksek maliyetli ve zaman alan veri kayıplarını önlemek için çok önemlidir. IBM MQ , bir sistem arızasından sonra hem kuyrukların hem de iletilerin yürürlükteki durumlarına geri gönderilmesini sağlar.

Bu konu aşağıdaki kısımları içerir:

- [“Kurtarma yordamları” sayfa 186](#)
- [“Yedekleme ve kurtarma için ipuçları” sayfa 187](#)
- [“Sayfa kümeleri kurtarılıyor” sayfa 189](#)
- [“CF yapılarının kurtarılması” sayfa 190](#)
- [“Belirli kurtarma hedeflerine ulaşılması” sayfa 191](#)
- [“Diğer ürünlere ilişkin yedeklemede dikkate alınması gerekenler” sayfa 192](#)
- [“Kurtarma ve CICS” sayfa 192](#)
- [“Kurtarma ve IMS” sayfa 193](#)
- [“Alternatif bir sitede kurtarma hazırlığı” sayfa 193](#)
- [“Kuyruk yöneticisi yedekleme etkinliği örneği” sayfa 193](#)

Kurtarma yordamları

IBM MQ için aşağıdaki yordamları geliştirin:

- Bir kurtarma noktası yaratıyor.
- Sayfa kümeleri yedekleniyor.
- CF yapıları yedekleniyor.
- Sayfa kümeleri kurtarılıyor.
- Alan dışı koşullardan (IBM MQ günlükleri ve sayfa kümeleri) kurtarılıyor.
- CF yapıları kurtarılıyor.

Bunlara ilişkin bilgi için bkz. [IBM MQ for z/OS Yönetimi](#) .

Aşağıda belirtilenler için sitenizde kullanılan yordamlara aşına olun:

- Donanım ya da güç arızasından kurtarma.

- z/OS bileşeni hatasından kurtarma.
- İşyeri dışında kurtarma kullanılarak, bir yer kesintisinden kurtarma.

z/OS Yedekleme ve kurtarma için ipuçları

Bazı yedekleme ve kurtarma görevlerini anlamak için bu konuyu kullanın.

Kuyruk yöneticisi yeniden başlatma işlemi, günlük bilgilerini sayfa kümelerine uygulayarak verilerinizi tutarlı bir duruma getirir. Sayfa kümeleriniz zarar görmüşse ya da kullanılamıyorsa, sayfa kümelerinizin yedek kopyalarını kullanarak sorunu çözebilirsiniz (tüm günlükler kullanılabiliyorsa). Günlük veri kümeleriniz zarar görmüşse ya da kullanılamıyorsa, tamamen kurtarılamayabilir.

Aşağıdaki noktaları göz önünde bulundurun:

- Düzenli olarak yedek kopyaları al
- Gerek duyabileceğiniz arşiv günlüklerini atmayın
- DDname ile sayfa kümesi ilişkisini değiştirme

Düzenli aralıklarla yedek kopyaları alın

Kurtarma noktası , IBM MQ sayfa kümelerinin bir yedek kopyaları kümesini ve bu sayfa kümelerini kurtarmak için gereken ilgili günlük veri kümelerini tanımlamak için kullanılan terimdir. Bu yedek kopyalar, sayfa kümesi kaybı durumunda (örneğin, sayfa kümesi G/Ç hatası) olası bir yeniden başlatma noktası sağlar. Bu yedek kopyaları kullanarak kuyruk yöneticisini yeniden başlatacaksanız, IBM MQ içindeki veriler bu kopyaların alındığı noktaya kadar tutarlıdır. Bu noktadan itibaren tüm günlüklerin kullanılabilir olması koşuluyla, IBM MQ hata noktasına kadar kurtarılabılır.

Yedek kopyalarınız ne kadar yeni olursa, IBM MQ o kadar hızlı sayfa kümelerindeki verileri kurtarabilir. Sayfa kümelerinin kurtarılması, tüm gerekli günlük veri kümelerinin kullanılabilir olmasına bağlıdır.

Kurtarma planlarken, yedekleme kopyalarının ne sıklıkta ve kaç tam yedekleme döngüsünün tutulduğunu belirlemeniz gerekir. Bu değerler, IBM MQ kurtarma işlemi için günlük veri kümelerinizi ve sayfa kümelerinin yedek kopyalarını ne kadar süre tutmanız gerektiğini gösterir.

Yedek kopyaların ne sıklıkta alacağına karar verirken, bir sayfa kümesini kurtarmak için gereken süreyi göz önünde bulundurun. Gereken süre aşağıdakiler tarafından belirlenir:

- Geçiş için günlük miktarı.
- Bir işletmenin arşiv manyetik bant birimlerini bağlaması ve kaldırması için geçen süre.
- Kurtarma için gerekli günlüğün bölümünü okumak için gereken süre.
- Değiştirilen sayfaları yeniden işlemek için gereken süre.
- Yedek kopyalar için kullanılan depolama ortamı.
- Yedek kopya oluşturmak ve geri yüklemek için kullanılan yöntem.

Genel olarak, yedekleme kopyalarını ne kadar sık yaparsanız, kurtarma o kadar az zaman alır, ancak kopya oluşturmak için o kadar çok zaman harclar.

Her kuyruk yöneticisi için, aşağıdakilerin yedek kopyalarını almanız gerekir:

- Arşiv günlüğü veri kümeleri
- Arşiv sırasında yaratılan BSDS kopyaları
- Sayfa kümeleri
- Nesne tanımlamalarınız
- CF yapılarınız

Yedek kopyalarınızın kaybolması ya da zarar görmesi riskini azaltmak için aşağıdakileri göz önünde bulundurun:

- Yedek kopyaların farklı depolama birimlerinde özgün kopyalarda saklanması.

- Yedek kopyaların özgün kopyalarda farklı bir yerde saklanması.
- Sayfa kümelerinizin her bir yedeğinin en az iki kopyasının oluşturulması ve tek bir günlük kaydı ya da tek bir BSDS kullanıyorsanız, arşiv günlüklerinizin ve BSDS ' nizin iki kopyasının oluşturulması. İkili günlük kaydı ya da BSDS kullanıyorsanız, hem arşiv günlüklerinin hem de BSDS ' nin tek bir kopyasını oluşturun.

IBM MQ ürününü bir üretim ortamına taşımadan önce, yedekleme yordamlarınızı tam olarak sınavın ve belgeleyin.

Sayfa kümelerinizi yedekleme

Sayfa kümelerini düzenli olarak yedeklemeniz gerekir. Bazı kuruluşlar, sayfa kümelerini günde iki kez yedekler.

Bir yedeğin yedeklemeyi kullanarak kurtarabilmesi için etkin ve arşiv günlüklerine gerek vardır. Yedekleme, kuyruk yöneticisi çalışırken alındıysa, dört denetim noktasının geri dönmesi için yeterli günlük verilerine gereksinim duyarsınız.

Sayfa kümelerini yedeklemek için ADRDSSU FastReplication işlevini kullanabilirsiniz ve kuyruk yöneticisi etkinken bunu yapabilirsiniz. Depolama havuzunda yeterli alan olduğundan emin olmanız gerektiğini unutmayın.

Nesne tanımlamalarınızın yedeklenmesi

Nesne tanımlamalarınızın yedek kopyalarını yaratın. Bunu yapmak için, yardımcı programın COMMAND işlevinin MAKEDEF özelliğini kullanın ([CSQUTIL ' in COMMAND işlevinin kullanılması](#) başlıklı konuda açıklanmaktadır).

Kuyruk yöneticisi veri kümelerinizin yedek kopyalarını her aldığınızda bunu yapmalı ve en güncel sürümünü saklamalısınız.

Bağlaşım olanağı yapılarınızın yedeklenmesi

Herhangi bir kuyruk paylaşım grubu oluşturduysanız, bunları kullanmıyor olsanız bile, CF yapılarınızın düzenli olarak yedeklerini almanız gerekir. Bunu yapmak için IBM MQ [BACKUP CFSTRUCT](#) komutunu kullanın. Bu komutu yalnızca, RECOVER (YES) özneteliğiyle tanımlanan CF yapılarında kullanabilirsiniz. Kalıcı paylaşılan iletilere ilişkin CF girişleri paylaşılan ileti veri kümesinde (SMDS) ya da Db2 olarak saklanan boşaltılmış ileti verilerine gönderme yaparsa, boşaltılan veriler alınır ve CF girişleriyle birlikte yedeklenir. Paylaşılan ileti veri kümeleri ayrı olarak yedeklenmemelidir.

CF yapısını geri yükleme süresini en aza indirmek için, her saat başı tüm CF yapılarınızın bir yedeğini almanız önerilir.

Günlük kullanımında artışı tek bir kuyruk yöneticisiyle sınırlama avantajına sahip olan tek bir kuyruk yöneticisinde tüm CF yapısı yedeklerinizi gerçekleştirilebilirsiniz. Diğer bir yöntem olarak, iş yükünü kuyruk paylaşım grubuna dağıtma avantajına sahip olan kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine yedekleme gerçekleştirilebilirsiniz. Hangi stratejiyi kullanırsanız kullanın, IBM MQ yedeklemeyi bulabilir ve kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisinden RECOVER CFSTRUCT işlemini gerçekleştirilebilir. CF yapısını kurtarmak için kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinin günlüklerine erişilmesi gerekir.

İleti güvenliği ilkelerinizin yedeklenmesi

İleti güvenliği ilkelerinizin yedeğini yaratmak için Advanced Message Security komutunu kullanıyorsanız, **dspmqsp1** komutunu -export parametresiyle çalıştırmak için [ileti güvenliği ilkesi yardımcı programını \(CSQOUTIL\)](#) kullanarak bir yedekleme yaratın ve EXPORT DD ' ye çıkışı olan ilke tanımlamalarını saklayın.

Kuyruk yöneticisi veri kümelerinizin yedek kopyalarını her aldığınızda, ileti güvenliği ilkelerinizin bir yedeğini oluşturmanız ve en güncel sürümü tutmanız gerekir.

Gerek duyabileceğiniz arşiv günlüklerini atmayın

IBM MQ , yeniden başlatma sırasında arşiv günlüklerini kullanması gerekebilir. Sistemin tam olarak geri yüklenebilmesi için yeterli arşiv günlüğü tutmanız gerekir. IBM MQ , geri yüklenen yedek kopyadan bir

sayfa kümesini kurtarmak için bir arşiv günlüğünü kullanabilir. Bu arşiv günlüğünü attıysanız, IBM MQ sayfa kümesini yürürlükteki durumuna geri yükleyemez. Arşiv günlüklerini ne zaman ve nasıl atacağınız [Arşiv günlüğü veri kümelerinin atılması](#) kısmında açıklanır.

Kuyruk yöneticinizin sayfa kümelerini ve kuyruk paylaşım grubu yapılarını kurtarmak için gereksinim duyduğunuz günlük RBA ve günlük aralığı sıra numarasını (LRSN) görüntülemek için /cpf DIS USAGE TYPE (ALL) komutunu kullanabilirsiniz. Kuyruk yöneticisine ilişkin önyükleme veri kümesi (BSDS) bilgilerini yazdırmak ve günlük RBA 'yı içeren günlükleri bulmak için [günlük eşlemi yazdırma yardımcı programını](#) (CSQJU004) kullanmalısınız.

CF yapılarında, LRSN 'yi içeren günlükleri bulmak için kuyruk paylaşım grubundaki her kuyruk yöneticisinde CSQJU004 yardımcı programını çalıştırmanız gerekir. Sayfa kümelerini ve yapılarını kurtarabilmek için bu günlüklere ve sonraki günlüklere gereksinim duyarsınız.

DDname ile sayfa kümesi ilişkisini değiştirme

IBM MQ , sayfa kümesi numarası 00 ile DDname CSQP0000, sayfa kümesi numarası 01 ile DDname CSQP0001'i ilişkilendirir ve CSQP0099' a kadar bu şekilde devam eder. IBM MQ , sayfa kümesinin ilişkilendirildiği DDname değerine dayalı olarak bir sayfa kümesine ilişkin kurtarma günlüğü kayıtlarını yazar. Bu nedenle, önceden bir PSID DDname ile ilişkilendirilmiş sayfa kümelerini taşımamalısınız.

Sayfa kümeleri kurtarıyor

Sayfa kümeleri kurtarılırken yer alan katsayıları ve yeniden başlatma sürelerinin nasıl azaltılacağını anlamak için bu konuyu kullanın.

Kurtarma stratejisindeki önemli bir faktör, bir kuyruk yöneticisi kesintisi için ne kadar süre tolerans gösterebileceğinizi belirler. Toplam kesinti süresi, bir sayfa kümesini yedekten kurtarmak ya da olağandışı bir sonlandırmadan sonra kuyruk yöneticisini yeniden başlatmak için geçen süreyi içerebilir. Yeniden başlatma süresini etkileyen etkenler, sayfa kümelerinizi ne sıklıkta yedeklediğinizi ve denetim noktaları arasında günlüğe ne kadar veri yazdığınızı içerir.

Olağandışı bir sonlandırmadan sonra yeniden başlatma süresini en aza indirmek için, iş birimlerini kısa tutun; böylece, sistem yeniden başlatıldığında en çok iki etkin günlük kullanılır. Örneğin, bir IBM MQ uygulaması tasarlıyorsanız, ilk zaman uyumlu MQI çağrısı ile kesinleştirme noktası arasında uzun bekleme aralığı olan bir MQGET çağrısı yerleştirmekten kaçınınız; bu, uzun süreli bir iş birimiyle sonuçlanabilir. Uzun çalışma birimlerinin bir diğer yaygın nedeni, kanal başlatıcısı için 5 dakikadan fazla olan toplu iş aralıklarıdır.

İş birimlerinin RBA 'sını görüntülemek ve eskilerin çözümüne yardımcı olmak için [DISPLAY THREAD](#) komutunu kullanabilirsiniz.

Bir sayfa kümesini ne sıklıkta yedeklemeniz gerekir?

Makul bir kısa kurtarma süresi gerekiyorsa, sık sık sayfa kümesi yedeklemesi gereklidir. Bu, bir sayfa kümesi çok küçük olduğunda ya da o sayfa kümesindeki kuyruklarda az miktarda etkinlik olduğunda bile geçerlidir.

Bir sayfa kümesinde kalıcı iletiler kullanıyorsanız, yedekleme sıklığı gün yerine saat olarak olmalıdır. Bu, sayfa kümesi sıfır için de bir durumdur.

Yaklaşık bir yedekleme sıklığını hesaplamak için, hedef toplam kurtarma süresini belirleyerek başlayın. Bu, aşağıdakilerden oluşur:

1. Soruna tepki vermek için geçen süre.
2. Sayfa kümesi yedek kopyasının geri yüklenmesi için geçen süre.

SnapShot yedekleme/geri yükleme özelliğini kullanırsanız, bu görevi gerçekleştirmek için geçen süre birkaç saniyedir. SnapShothakkında bilgi için bkz. *DFSMSdss Storage Administration Guide*.

3. Sayfa kümesini kurtarmak için gereken ek süre de içinde olmak üzere, kuyruk yöneticisinin yeniden başlatılması gereken süre.

Bu, sayfa kümesinin son yedeklenmesinden bu yana etkin ve arşiv günlüklerinden okunması gereken günlük verisi miktarına önemli ölçüde bağlıdır. Hasarlı sayfa kümesiyle doğrudan ilişkili günlük verilerine ek olarak, bu tür tüm günlük verileri de okunmalıdır.

Not: *Fuzzy backup* (bir iş birimi etkinken günlüklerin ve sayfa kümelerinin anlık görüntüsünün alındığı yerde) kullanıldığında, en çok üç ek denetim noktasının okunması gerekebilir ve bu, bir ya da daha fazla ek günlük okunması gerekmesine neden olabilir.

Sayfa kümesinin kurtarılmasına ne kadar süre izin vereceğinize karar verirken göz önünde bulundurmanız gereken etkenler şunlardır:

- Normal işleme sırasında etkin günlüklere veri yazma hızı, ileti hızına ek olarak, iletilerin sisteminize nasıl geldiğine bağlıdır.

Bir kanal üzerinden alınan ya da gönderilen iletiler, yerel olarak oluşturulan ve alınan iletilerden daha fazla veri günlüğüne kaydedilmesiyle sonuçlanır.

- Verilerin arşivden ve etkin günlüklerden okunma hızı.

Günlükleri okurken, ulaşılabilir veri hızı, kullanılan aygıtlara ve belirli DASD altsisteminizdeki toplam yüke bağlıdır.

Çoğu manyetik bant birimiyle, büyük blok boyutuna sahip arşivlenmiş günlükler için daha yüksek veri hızları elde etmek mümkündür. Ancak, kurtarma için bir arşiv günlüğü gerekiyorsa, etkin günlükteki tüm veriler de okunmalıdır.

CF yapılarının kurtarılması

CF yapılarına ilişkin kurtarma işlemini anlamak için bu konuyu kullanın.

RECOVER CFSTRUCT komutunu işlemek için kuyruk paylaşım grubundaki en az bir kuyruk yöneticisi etkin olmalıdır. Kurtarma zaten etkin bir kuyruk yöneticisi tarafından gerçekleştirildiğinden, CF yapısı kurtarma işlemi kuyruk yöneticisinin yeniden başlatma süresini etkilemez.

Kurtarma işlemi, RECOVER CFSTRUCT komutu tarafından yönetilen iki mantıksal adımdan oluşur:

1. Yedeğin bulunması ve geri yüklenmesi.
2. CF yapısında tutulan tüm günlüğe kaydedilen güncellemeler, CF yapısını kullanan kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinin günlüklerinden kalıcı iletilerle birleştiriliyor ve değişiklikler yedeklemeye uygulanıyor.

Çok sayıda günlük verisinin okunması gerekebileceğinden, ikinci adım çok daha uzun sürebilir. Sık sık yedekleme yaparsanız ya da aynı anda birden çok CF yapısını kurtardıysanız ya da her ikisini birden yaparsanız geçen süreyi azaltabilirsiniz.

Kurtarma işlemini gerçekleştiren kuyruk yöneticisi, Db2 ve önyüklemeye veri kümelerindeki verileri kullanarak diğer tüm kuyruk yöneticilerinin günlüklerinde ilgili yedekleri bulur. Kuyruk yöneticisi, bu yedeklemeleri, son yedeklemeden hemen önce, hata noktasına kadar, kuyruk paylaşım grubunda doğru zaman sırasıyla yeniden gerçekleştirir.

Bir CF yapısını kurtarmak için gereken süre, yeniden oynatılması gereken kurtarma günlüğü verilerinin miktarına bağlıdır; bu da yedeklerin sıklığına bağlıdır. En kötü ihtimalle, bir kuyruk yöneticisinin günlüğünü yazmak kadar uzun sürer. Örneğin, altı kuyruk yöneticisi içeren bir kuyruk paylaşım grubunuz varsa, bir saatlik günlük etkinliğinin yeniden oynatılması altı saat sürebilir. Okuma toplu olarak yapılabildiğinden ve farklı kuyruk yöneticisinin günlükleri paralel olarak okunabildiğinden, genel olarak bu işlem bundan daha kısa sürer. Başlangıç noktası olarak, CF yapılarınızı her saat yedeklemenizi öneririz.

Başarısız bir CF yapısı varken, tüm kuyruk yöneticileri diğer CF yapılarındaki paylaşılmayan kuyruklar ve kuyruklar ile çalışmaya devam edebilir. Denetim yapısı da başarısız olduysa, RECOVER CFSTRUCT

komutunu vermeden önce kuyruk paylaşım grubundaki kuyruk yöneticilerinden en az birinin başlatılması gerekir.

CF yapılarının yedeklenmesi kayda değer günlük yazma kapasitesi gerektirebilir ve bu nedenle, yedeklemeyi yapan kuyruk yöneticisine büyük bir yük yükleyebilir. Yedeklemeleri yapmak için hafif yüklü bir kuyruk yöneticisi seçin; meşgul sistemler için, kuyruk paylaşım grubuna ek bir kuyruk yöneticisi ekleyin ve bunu özel olarak yedeklemeler için ayırın.

z/OS Belirli kurtarma hedeflerine ulaşılması

Yedekleme sıklığını ayarlayarak belirli kurtarma hedefi sürelerine nasıl ulaşabileceğinize ilişkin yol gösterici bilgiler için bu konuyu kullanın.

Ulaşılması gereken belirli kurtarma hedefleriniz varsa (örneğin, kuyruk yöneticisi kurtarma işleminin tamamlanması ve yeniden başlatma işleminin xx saniye içinde olağan başlatma süresine ek olarak), yedekleme sıklığını (saat cinsinden) tahmin etmek için aşağıdaki hesaplamayı kullanabilirsiniz:

$$\text{Formül (A)} \\ \text{Backup frequency (in hours)} = \frac{\text{Required restart time (in secs)} * \text{System recovery log read rate (in MB/sec)}}{\text{Application log write rate (in MB/hour)}}$$

Not: Daha sonra verilen örnekler, sayfa kümelerinizi sık sık yedeklemeniz gerektiğini vurgulamayı amaçlar. Hesaplamalar, çoğu günlük etkinliğinin çok sayıda kalıcı iletiden türetildiğini varsayar. Ancak, günlük etkinliği miktarının kolayca hesaplanmadığı durumlar vardır. Örneğin, bir kuyruk paylaşım grubu ortamında, diğer kaynaklara ek olarak paylaşılan kuyrukların güncellendiği bir iş birimi, UOW kayıtlarının IBM MQ günlüğüne yazılmasına neden olabilir. Bu nedenle, Formül (A) içindeki Uygulama günlüğü yazma hızı yalnızca IBM MQ günlüklerinin doldurduğu gözlemlenen hızdan doğru olarak türetilir.

Örneğin, IBM MQ MQI clients ' in saniyede toplam 100 kalıcı ileti yüklediği bir sistem düşünün. Bu durumda, tüm iletiler yerel olarak oluşturulur.

Her ileti kullanıcı uzunluğu 1 KB ise, her saat için günlüğe kaydedilen veri miktarı yaklaşık olarak aşağıdaki gibidir:

$$100 * (1 + 1.3) \text{ KB} * 3600 = \text{approximately } 800 \text{ MB} \\ \text{where} \\ 100 = \text{the message rate a second} \\ (1 + 1.3) \text{ KB} = \text{the amount of data logged for each 1 KB of persistent messages}$$

75 dakikalık genel hedef kurtarma süresini göz önünde bulundurun. Soruna yanıt vermek ve sayfa kümesi yedek kopyasını geri yüklemek için 15 dakika izin verdiyseniz, kuyruk yöneticisi kurtarma ve yeniden başlatma işlemi formül (A) uygulayarak 60 dakika (3600 saniye) içinde tamamlanmalıdır. Tüm gerekli günlük verilerinin RVA2-T82 DASD üzerinde olduğu varsayılarak (saniyede yaklaşık 2.7 MB kurtarma hızına sahip), bu işlem için en az her aralığa ilişkin bir sayfa kümesi yedekleme sıklığı gerekir:

$$3600 \text{ seconds} * 2.7 \text{ MB a second} / 800 \text{ MB an hour} = 12.15 \text{ hours}$$

IBM MQ uygulama gününüz yaklaşık 12 saat sürerse, her gün bir yedekleme uygun olur. Ancak, uygulama günü 24 saat sürerse, her gün iki yedek daha uygundur.

Başka bir örnek, tüm iletilerin istek yanıt uygulamaları için olduğu bir üretim sistemi olabilir (başka bir deyişle, bir alıcı kanalda kalıcı bir ileti alınır ve kalıcı bir yanıt iletisi oluşturulur ve bir gönderen kanalı gönderilir).

Bu örnekte, ulaşılan toplu iş boyutu birdir ve bu nedenle her ileti için bir toplu iş vardır. Saniyede 50 istek yanıtları varsa, toplam yük saniyede 100 kalıcı iletidir. Her iletinin uzunluğu 1 KB ise, her saat için günlüğe kaydedilen veri miktarı yaklaşık olarak aşağıdaki gibidir:

$$50((2 * (1+1.3) \text{ KB}) + 1.4 \text{ KB} + 2.5 \text{ KB}) * 3600 = \text{approximately } 1500 \text{ MB}$$

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

Kuyruk yöneticisi kurtarma ve yeniden başlatma işlemlerini 30 dakika (1800 saniye) içinde gerçekleştirmek için, tüm gerekli günlük verilerinin RVA2-T82 DASD üzerinde olduğunu varsayarak, sayfa kümesi yedeklemenin en az her zaman gerçekleştirilmesini gerektirir:

$$1800 \text{ seconds} * 2.7 \text{ MB a second} / 1500 \text{ MB an hour} = 3.24 \text{ hours}$$

Yedekleme sıklığının düzenli olarak incelenmesi

IBM MQ günlük kullanımınızı bir saat MB cinsinden izleyin. Bu denetimi düzenli aralıklarla gerçekleştirin ve gerekirse sayfa kümesi yedekleme sıklığını ayarlayın.

Diğer ürünlere ilişkin yedeklemede dikkate alınması gerekenler

IBM MQ ürününü CICS ya da IMS ile kullanıyorsanız, bu ürünlerle ilgili yedekleme stratejinizin etkilerini de göz önünde bulundurmanız gerekir. Veri olanağı sıradüzensel depolama yöneticisi (DFHSM), veri depolamasını yönetir ve IBM MQ tarafından kullanılan depolama alanıyla etkileşimde bulunabilir.

DFHSM ile yedekleme ve kurtarma

Veri olanağı hiyerarşik depolama yöneticisi (DFHSM), sisteminizdeki depolama aygıtları arasında otomatik alan kullanılabilirliği ve veri kullanılabilirliği yönetimi sağlar. Bunu kullanırsanız, IBM MQ depolama alanına/depolama alanından otomatik olarak veri taşıdığını bilmeniz gerekir.

DFHSM, son zamanlarda kullanılmayan veri kümelerini alternatif depolamaya taşıyarak DASD alanınızı verimli bir şekilde yönetir. Ayrıca, yeni ya da değiştirilen veri kümelerini manyetik banda ya da DASD yedekleme birimlerine otomatik olarak kopyalayarak verilerinizi kurtarma için kullanılabilir duruma getirir. Veri kümelerini silebilir ya da başka bir aygıtta taşıyabilir. İşlemleri günlük olarak, belirli bir zamanda gerçekleşir ve bir veri kümesini silmeden ya da taşımadan önce önceden belirlenmiş bir süre boyunca saklamaya izin verir.

Tüm DFHSM işlemlerini el ile de gerçekleştirebilirsiniz. DFHSM hakkında daha fazla bilgi için [z/OS DFSMS](#) ürün belgelerine bakın. IBM MQ ile DFHSM kullanıyorsanız, DFHSM 'nin aşağıdakileri yaptığını unutmayın:

- Kataloğa alınmış veri kümelerini kullanır.
- Sayfa kümeleri ve günlüklerde çalışır.
- VSAM veri kümelerini destekler.

Kurtarma ve CICS

CICS kaynaklarının kurtarılması, IBM MQvarlığından etkilenmez. CICS , IBM MQ ögesiniCICS dışı bir kaynak (ya da dış kaynak yöneticisi) olarak tanır ve CICS kaynak yöneticisi arabirimini (RMI) kullanarak uyumluluk noktası eşgüdümü isteklerine katılımcı olarak IBM MQ içerir. CICS kurtarma ve CICS kaynak yöneticisi arabirimiyle ilgili daha fazla bilgi için [CICS](#) ürün belgelerine bakın.

Kurtarma ve IMS

IMS , IBM MQ ' i bir dış altsistem olarak ve eşitleme noktası koordinasyonunda bir katılımcı olarak tanır. Dış altsistem kaynaklarına ilişkin IMS kurtarma, [IMS](#) ürün belgelerinde açıklanır.

Alternatif bir sitede kurtarma hazırlığı

IBM MQ bilgi işlem merkezinin toplam kaybı söz konusu olursa, kurtarma yerindeki başka bir IBM MQ sisteminde kurtarma yapabilirsiniz.

Kurtarma yerindeki bir IBM MQ sistemini kurtarmak için sayfa kümelerini ve günlükleri düzenli olarak yedeklemeniz gerekir. Tüm veri kurtarma işlemlerinde olduğu gibi, olağanüstü durum kurtarma hedefleri, mümkün olduğunca az veri, iş yükü işleme (güncelleme) ve zaman kaybıdır.

Kurtarma yerinde:

- Kurtarma IBM MQ kuyruk yöneticisi , kayıp kuyruk yöneticisiyle aynı ada sahip olmalıdır.
- Kurtarma kuyruğu yöneticisinde kullanılan sistem parametresi modülünün, kayıp kuyruk yöneticisiyle aynı parametreleri içerdiğini doğrulayın.

Daha fazla bilgi için bkz. [IBM MQ for z/OS Yönetimi](#) ve [IBM MQ for z/OS sorunlarının giderilmesi](#) .

Kuyruk yöneticisi yedekleme etkinliği örneği

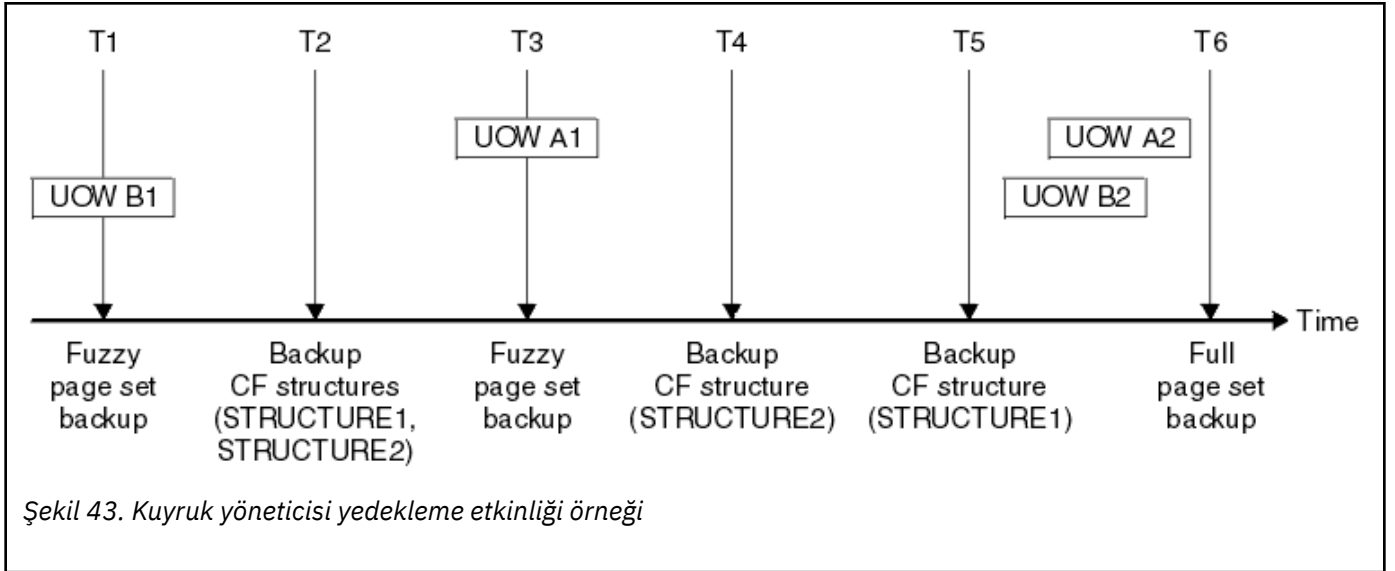
Bu konu, kuyruk yöneticisi yedekleme etkinliğinin bir örneği olarak gösterilir.

Kuyruk yöneticisi yedekleme stratejinizi planladığınızda, günlük verilerinin doğru miktarda tutulması önemli bir unsurdur. [Günlüklerin yönetilmesi](#) , kuyruk yöneticisinin sistem kurtarma RBA 'sına başvurarak hangi günlük veri kümelerinin gerekli olduğunu nasıl belirleyeceğini açıklar. IBM MQ , aşağıdakilere ilişkin bilgileri kullanarak sistem kurtarma RBA ' sını belirler:

- Şu anda etkin iş birimleri.
- Arabellek havuzlarından diske henüz temizlenmemiş sayfa kümesi güncellemeleri.
- CF yapısı yedeklemeleri ve bu kuyruk yöneticisinin günlüğünün bunları kullanan herhangi bir kurtarma işleminde gerekli bilgileri içerip içermediğini belirler.

Ortam kurtarma işlemini gerçekleştirebilmek için yeterli günlük verilerini saklamanız gerekir. Sistem kurtarma RBA zaman içinde artarken, saklanması gereken günlük verileri miktarı yalnızca sonraki yedekler alındığında azalır. CF yapısı yedeklemeleri IBM MQ tarafından yönetilir ve sistem kurtarma RBA raporlanırken dikkate alınır. Bu, pratikte tutulması gereken günlük verilerinin miktarının yalnızca sayfa kümesi yedekleri alındığında azaldığı anlamına gelir.

Şekil 43 [sayfa 194](#) içinde, bir kuyruk paylaşım grubunun üyesi olan bir kuyruk yöneticisindeki yedekleme etkinliği örneği, kurtarma RBA ' sının her bir yedeklemeye göre nasıl değişeceği ve bunun saklanması gereken günlük verisi miktarını nasıl etkilediği gösterilmektedir. Örnekte, kuyruk yöneticisi yerel ve paylaşılan kaynakları kullanır: sayfa kümeleri ve iki CF yapısı: STRUCTURE1 ve STRUCTURE2.



Bu, her bir zaman noktasında olan şey:

Zaman noktası T1

Sayfa kümelerinizin yaklaşık bir yedeği, Sayfa kümelerinin yedeklenmesi ve kurtarılması konusunda açıklandığı gibi oluşturulur.

Kuyruk yöneticisinin sistem kurtarma RBA ' sı aşağıdakilerden en düşüktür:

- Bu noktada yedeklenmekte olan sayfa kümelerinin kurtarma RNA ' ları.
- CF uygulama yapılarını kurtarmak için gereken en düşük kurtarma RBA 'si. Bu, daha önce yaratılan STRUCTURE1 ve STRUCTURE2 yedeklerinin kurtarılması ile ilgilidir.
- Kuyruk yöneticisi içindeki en eski etkin iş birimine ilişkin kurtarma RBA 'si (UOWB1).

Bu zaman noktasına ilişkin sistem kurtarma RBA, yaklaşık yedekleme işleminin bir parçası olan DISPLAY USAGE komutu tarafından verilen iletiler tarafından verilir.

Zaman noktası T2

CF yapılarının yedekleri yaratılır. CF yapısı STRUCTURE1 önce yedeklenir, ardından STRUCTURE2 gelir.

T1 adresindeki sistem kurtarma RBA 'sından belirlenenle aynı veriler, T1' de alınan sayfa kümesi yedekleri kullanılarak kurtarılması gerektiğinden, saklanması gereken günlük verisi miktarı değiştirilmez.

Zaman noktası T3

Başka bir bulanık yedekleme yaratılır.

Kuyruk yöneticisinin sistem kurtarma RBA ' sı aşağıdakilerden en düşüktür:

- Bu noktada yedeklenmekte olan sayfa kümelerinin kurtarma RNA ' ları.
- STRUCTURE1 , STRUCTURE2öncesinde yedeklendiğinden, STRUCTURE1CF yapısını kurtarmak için gereken en düşük kurtarma RBA 'si.
- Kuyruk yöneticisi içindeki en eski etkin iş birimine ilişkin kurtarma RBA 'si (UOWA1).

Bu zaman noktasına ilişkin sistem kurtarma RBA, yaklaşık yedekleme işleminin bir parçası olan DISPLAY USAGE komutu tarafından verilen iletiler tarafından verilir.

Artık, bu yeni sistem kurtarma RBA tarafından belirlendiği şekilde, alıkonan günlük verilerini azaltabilirsiniz.

Zaman noktası T4

STRUCTURE2CF yapısının bir yedeği alınır. En eski CF yapısı yedeğinin kurtarılmasına ilişkin kurtarma RBA 'si, T2zamanında yedeklenen STRUCTURE1CF yapısının yedeğiyle ilgilidir.

Bu CF yapısı yedeğinin yaratılması, saklanması gereken günlük verileri miktarını etkilemez.

Zaman noktası T5

STRUCTURE1CF yapısının bir yedeği alınır. Gerekli en eski CF yapısı yedeğinin kurtarılmasına ilişkin kurtarma RBA, şimdi T4zamanında yedeklenen STRUCTURE2CF yapısının kurtarılmasına ilişkin.

Bu CF yapısı yedeğinin yaratılması, saklanması gereken günlük verileri miktarını etkilemez.

Zaman noktası T6

Sayfa kümelerinizin tam yedeği, Sayfa kümelerinin yedeklenmesi ve kurtarılması konusunda açıklandığı gibi alınır.

Kuyruk yöneticisinin sistem kurtarma RBA ' sı aşağıdakilerden en düşüktür:

- Bu noktada yedeklenmekte olan sayfa kümelerinin kurtarma RNA ' ları.
- CF yapılarını kurtarmak için gereken en düşük kurtarma RBA 'si. Bu, STRUCTURE2CF yapısının kurtarılması ile ilgilidir.
- Kuyruk yöneticisi içindeki en eski etkin iş birimine ilişkin kurtarma RBA ' sı. Bu durumda, geçerli iş birimi yoktur.

Bu zaman noktasına ilişkin sistem kurtarma RBA, tam yedekleme işleminin bir parçası olan DISPLAY USAGE komutu tarafından verilen iletiler tarafından verilir.

Tam yedekle ilişkili sistem kurtarma RBA daha yeni olduğundan, günlük verileri azaltılabilir.

z/OS

z/OS UNIX ortamının planlanması

IBM MQ kuyruk yöneticisi, kanal başlatıcısı ve mqweb sunucusu içindeki belirli işlemler, olağan işlemleri için z/OS UNIX System Services (z/OS UNIX) kullanır.

Kuyruk yöneticisi ve kanal başlatıcısı tarafından başlatılan görev kullanıcı kimliklerinin z/OS UNIX'e erişebilmesi için bir UID' ye sahip bir OMVS kesiminin tanımlanmış olması gerekir. Kullanıcı kimlikleri z/OS UNIX içinde özel izinler gerektirmez.

Not: Kuyruk yöneticisi ve kanal başlatıcısı z/OS UNIX olanaklarından (örneğin, TCP/IP hizmetleriyle arabirim kurmak için) yararlı olsa da, z/OS UNIX dosya sistemindeki IBM MQ kuruluş dizininin içeriğine erişimleri gerekmez. Sonuç olarak, kuyruk yöneticisi ve kanal başlatıcısı, z/OS UNIX dosya sisteminin yolunu belirtmek için herhangi bir yapılandırma gerektirmez.

IBM MQ Console ve REST API' ı barındıran mqweb sunucusu, z/OS UNIX dosya sistemindeki IBM MQ kuruluş dizinindeki dosyaları kullanır. Yapılandırma ve günlük dosyaları gibi verileri depolamak için kullanılan başka bir dosya sistemine de erişilmesi gerekir. Bu z/OS UNIX dosya sistemlerine başvurmak için, mqweb tarafından başlatılan JCL görevinin uyarlanması gerekir.

z/OS UNIX dosya sistemindeki IBM MQ dizininin içeriği, IBM MQ' e bağlanan uygulamalar tarafından da kullanılır. Örneğin, IBM MQ classes for Java ya da IBM MQ classes for JMS arabirimlerini kullanan uygulamalar.

İlgili yapılandırma yönergeleri için aşağıdaki konulara bakın:

- [IBM MQ classes for Java ile ilgili ortam değişkenleri](#)
- [IBM MQ classes for Java kitaplıkları](#)
- [Ortam değişkenlerini tanımlama](#)
- [Java Native Interface \(JNI\) kitaplıklarının yapılandırılması](#)

z/OS

Advanced Message Security planlaması

TLS (veya SSL), bir ağda akan iletileri şifrelemek ve korumak için kullanılabilir, ancak bu, kuyruktayken iletileri korumaz ("atıl"). Advanced Message Security (AMS) iletileri, yalnızca iletinin amaçlanan alıcılarının o iletiyi okuyabilmesi için kuyruğa ilk yerleştirdikleri andan itibaren korur. İletiler, koyma işlemi sırasında şifrelenir ve imzalanır ve alma işlemi sırasında korunmaz.

AMS , iletileri farklı şekillerde koruyacak şekilde yapılandırılabilir:

1. Bir ileti imzalanabilir. İleti açık metin içinde, ancak imzalanmış bir sağlama toplamı var. Bu, ileti içeriğinde yapılan değişikliklerin saptanmasını sağlar. İmzalı içerikten verileri kimin imzaladığını belirleyebilirsiniz.
2. Bir ileti şifrelenebilir. Şifre çözme anahtarı olmadan içerik hiç kimse tarafından görülemez. Şifre çözme anahtarı her alıcı için şifrelenir.
3. Bir ileti şifrelenebilir ve imzalanabilir. Şifre çözme anahtarı her alıcı için şifrelenir ve imzalamadan iletiyi kimin gönderdiğini belirleyebilirsiniz.

Şifreleme ve imzalama, dijital sertifikaları ve anahtar halkalarını kullanır.

Verilerin istemci kanalına konmadan önce korunması için AMS' i kullanmak üzere bir istemci ayarlayabilirsiniz. Korunan iletiler uzak kuyruk yöneticisine gönderilebilir ve uzak kuyruk yöneticisini bu iletileri işleyecek şekilde yapılandırmanız gerekir.

ayarlamaAMS

AMS işini yapmak için bir AMS adres alanı kullanılır. Bu, anahtar halkalarına ve sertifikalarına erişim vermek ve bunların kullanımını korumak için ek güvenlik ayarlarına sahiptir.

Hangi kuyrukların korunacağını, kuyruklara ilişkin güvenlik ilkelerini tanımlamak için bir yardımcı program (CSQOUTIL) kullanarak yapılandırabilirsiniz.

AMS ayarlandıktan sonra

İleti koyan kişiler ve ileti alan kişiler için bir dijital sertifika ve bir anahtarlık ayarlamanız gerekir.

z/OS üzerindeki bir kullanıcının Bob 'a bir ileti göndermesi gerekiyorsa, AMS , Bob için genel sertifikanın bir kopyasına gereksinim duyar.

Bob, Alice 'den gelen bir iletiyi işlemek isterse, AMS Alice için genel sertifikaya ya da Alice tarafından kullanılan sertifika yetkilisi sertifikasına ihtiyaç duyar.



Uyarı: Aşağıdakileri yapmanız gerekir:

- Kuyruklara kimlerin girebileceğini ya da kuyruklardan kimlerin gelebileceğini dikkatli bir şekilde planlayın
- Kişileri ve sertifika adlarını tanımlayın.

Hata yapmak kolaydır ve sorunları çözmek zor olabilir.

İlgili kavramlar

“Kuyruk yöneticinizin planlanması” sayfa 141

Bir kuyruk yöneticisi ayarlarken, kuyruk yöneticisinin işletmenizin gereksinimlerini karşılaması için planlamanız kuyruk yöneticisinin büyümesine izin vermelidir.

z/OS

Managed File Transfer planlaması

Bu bölümü, sisteminizi z/OSüzerinde Managed File Transfer (MFT) uygulamasını çalıştırmak üzere nasıl ayarlamanız gerektiğiyle ilgili yol gösterici olarak kullanın.

z/OS

Managed File Transfer için planlama-donanım ve yazılım gereksinimleri

z/OSüzerinde Managed File Transfer (MFT) programını çalıştırmak için sisteminizde donanım ve yazılım gereksinimlerini nasıl ayarlamanız gerektiği konusunda yol gösterici olarak bu konuyu kullanın.

Yazılım gereksinimleri

Managed File Transfer , bazı kabuk komut dosyaları ve programı yapılandırmak ve çalıştırmak için JCL ile Java içinde yazılır.

Önemli: Managed File Transfer' yi yapılandırmak için z/OS UNIX System Services (z/OS UNIX) ile ilgili bilgi sahibi olmanız gerekir. Örneğin:

- /u/userID/myfile.txt gibi adlarla birlikte dosya dizini yapısı
- z/OS UNIX komutları, örneğin:
 - cd (dizini değiştir)
 - ls (Liste)
 - chmod (dosya izinlerini değiştirme)
 - chown (dosyaya ya da dizine erişebilen dosya iyeliğini ya da gruplarını değiştirme)

MFT ' yi yapılandırabilmek ve çalıştırabilmek için z/OS UNIX içinde aşağıdaki ürünlere gereksinim duyarsınız:

1. Java, örneğin, dizinde /java/java80_bit64_GA/J8.0_64/
2. IBM MQ 9.3.0, örneğin, dizinde /mqm/V9R3M0
3. Durum ve geçmiş için Db2 komutunu kullanmak istiyorsanız, Db2 JDBC kitaplıklarını (örneğin, /db2/db2v10/jdbc/libsdizinine) kurmanız gerekir.

Ürün kaydı

Başlangıçta Managed File Transfer , sys1.parm.lib(IFAPRDxx) birleştirmesinde kaydı denetler. Aşağıdaki kod, MFT ' yi nasıl kaydettiğinize bir örnektir:

```
PRODUCT OWNER('IBM CORP')
NAME('WS MQ FILE TRANS')
ID(5655-MFT)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('WS MQ FILE TRANS')
STATE(ENABLED)
```

Disk Yeri

IBM MQ for z/OS Program Dizini, Managed File Transfer için DASD ve zFS depolama gereksinimlerini belirtir. IBM MQ for z/OS için Program Dizinine ilişkin karşıdan yükleme bağlantıları için bkz. [IBM MQ 9.3 Ürün belgeleri ve Program Dizinlerine ilişkin PDF dosyaları](#).

Managed File Transfer için planlama-topolojiler

z/OS üzerinde Managed File Transfer (MFT) uygulamasını çalıştırmak için sisteminizde hangi topolojiye gereksinim duyacağınız konusunda yol gösterici olarak bu konuyu kullanın.

Managed File Transfer kuyruk yöneticileri

IBM MQ Managed File Transfer topolojileri şunlardan oluşur:

Aracılar ve ilişkili kuyruk yöneticileri

Aracı, durum bilgilerini korumak ve iş isteklerini almak için aracı kuyruk yöneticisinde barındırılan sistem kuyruklarını kullanır.

Bir komut kuyruğu yöneticisi

Bu, MFT topolojisine bir ağ geçidi işlevi görür. Aracı kuyruk yöneticilerine gönderen ve alıcı kanalları ya da kümeleme yoluyla bağlanır. Belirli komutlar çalıştırıldığında, doğrudan komut kuyruğu yöneticisine bağlanır ve belirtilen aracıya bir ileti gönderir. Bu ileti, IBM MQ ağı üzerinden aracı kuyruk yöneticisine yönlendirilir ve aracı tarafından alınır ve işlenir.

Bir koordinasyon kuyruk yöneticisi

Bu, topolojinin tamamını bilen merkezi bir merkezdir. Eşgüdüm kuyruk yöneticisi, bir topolojideki tüm aracı kuyruk yöneticilerine gönderen ve alıcı kanalları aracılığıyla ya da kümeleme kullanılarak

bağlanır. Aracılar düzenli olarak durum bilgilerini koordinasyon kuyruk yöneticisine yayınlar ve aktarım şablonlarını orada saklar.

Tek bir kuyruk yöneticisinin bir topoloji içinde birden çok rol gerçekleştirmesi mümkündür. Örneğin, aynı kuyruk yöneticisi bir topoloji için hem koordinasyon kuyruk yöneticisi hem de komut kuyruk yöneticisi olarak yapılandırılabilir.

Birden çok kuyruk yöneticisi kullanıyorsanız, kuyruk yöneticileri arasında kanallar oluşturmanız gerekir. Bunu kümeleme kullanarak ya da noktadan noktaya bağlantıları kullanarak yapabilirsiniz.

IBM MQ Managed File Transfer for z/OS kullanırken, bir topoloji içindeki farklı roller için hangi kuyruk yöneticilerinin kullanılacağını belirlerken göz önünde bulundurulması gereken birçok şey vardır.

Aracı kuyruğu yöneticileri

Bir IBM MQ Managed File Transfer for z/OS aracısının aracı kuyruk yöneticisi z/OS üzerinde çalışıyor olmalıdır.

Eğer:

- Aracı, IBM MQ 9.1 üzerinde ya da daha sonraki bir sürümde Managed File Transfer for z/OS çalıştırıyor
- Ve aracı kuyruk yöneticisi, IBM MQ Advanced for z/OS Value Unit Edition (Advanced VUE) için lisanslanır

Aracı, CLIENT iletimini kullanarak kuyruk yöneticisine bağlanabilir.



Şekil 44. z/OS üzerindeki MFT 9.1 araçları, kuyruk yöneticisinin Advanced VUE için lisanslandığı varsayılarak, CLIENT iletimini kullanarak bir kuyruk yöneticisine bağlanabilir.

Eğer:

- Aracı, IBM MQ 9.0 üzerinde ya da daha önceki bir sürümde Managed File Transfer for z/OS çalıştırıyor
- Ya da Agent kuyruk yöneticisi Managed File Transfer for z/OS komutunu IBM MQ 9.0 üzerinde ya da daha sonraki bir sürümde çalıştırıyor ve aracı kuyruk yöneticisi MFT, IBM MQ Advanced for z/OS ya da Advanced VUE için lisanslıdır.

Aracının BINDINGS iletimini kullanarak kuyruk yöneticisine bağlanması gerekir.



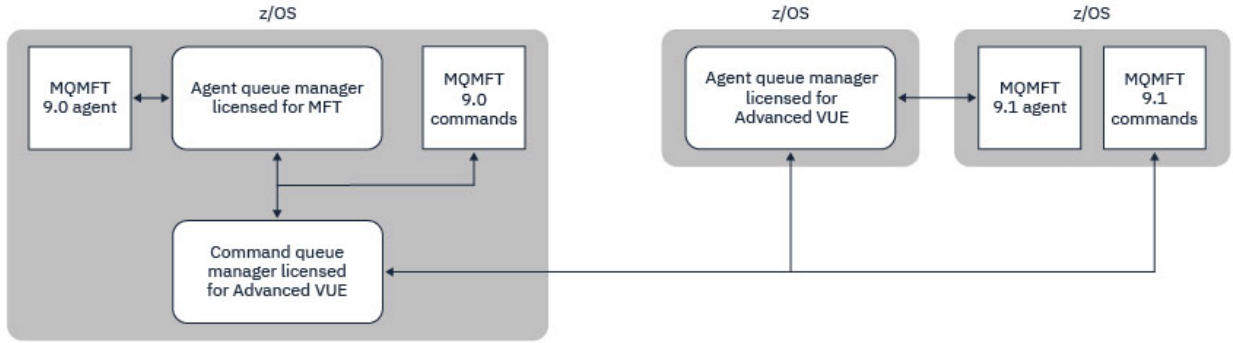
Şekil 45. MFT ya da IBM MQ Advanced için lisanslı bir aracı kuyruk yöneticisi olan z/OS ve 9.1 araçlarında MFT 9.0 araçları, BINDINGS iletimini kullanarak bağlanmalıdır.

Komut kuyruğu yöneticileri

Hangi MFT komutlarının ve işlemlerinin hangi kuyruk yöneticisine bağlandığı başlıklı konuda, Managed File Transfer topolojisi için komut kuyruğu yöneticisine bağlanan tüm komutlar gösterilir.

Not: Bu komutlar z/OS üzerinde çalıştırılırken, komut kuyruğu yöneticisinin de z/OS üzerinde olması gerekir.

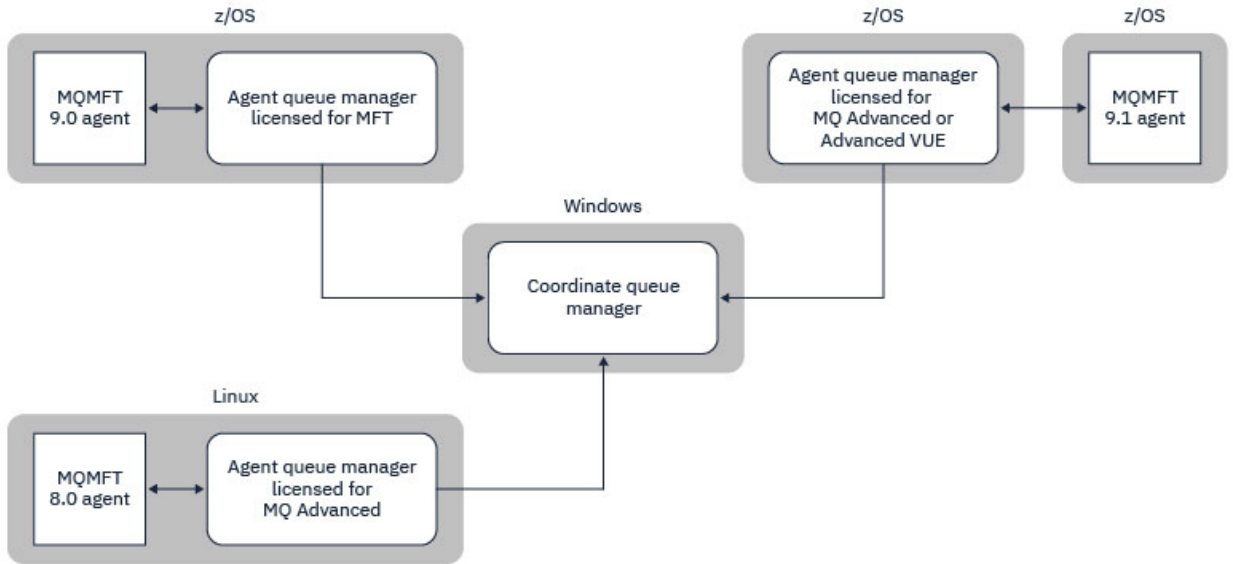
Komut kuyruğu yöneticisi Advanced VUE için lisanslandıysa, komutlar CLIENT iletimini kullanarak kuyruk yöneticisine bağlanabilir. Tersi durumda, komutların BINDINGS iletimini kullanarak komut kuyruğu yöneticisine bağlanması gerekir.



Şekil 46. Komutlar, MFT topolojisi için komut kuyruğu yöneticisine bağlanır. Bu komutlar z/OS üzerinde çalıştırılırken, komut kuyruğu yöneticisinin de z/OS üzerinde olması gerekir.

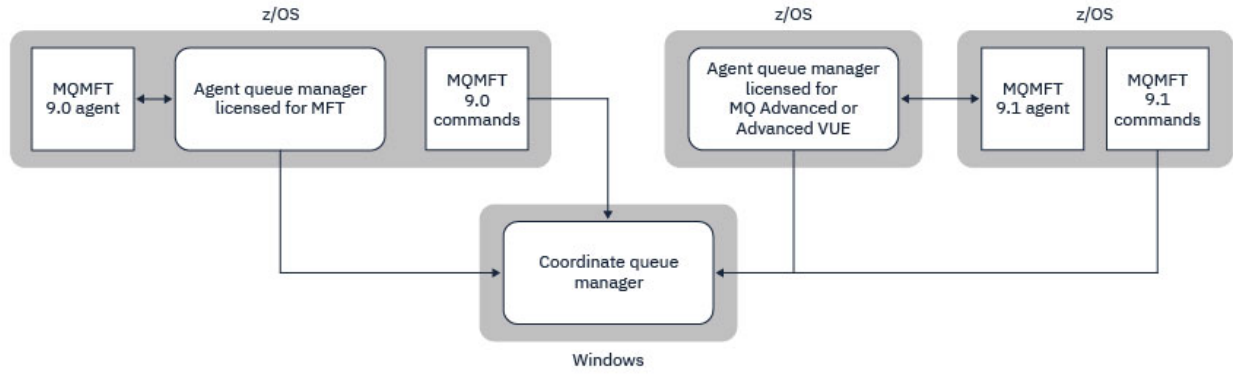
Koordinasyon kuyruğu yöneticileri

IBM MQ Managed File Transfer for z/OS araçları, koordinasyon kuyruk yöneticisinin z/OS üzerinde çalıştığı ya da çoklu platform üzerinde çalıştığı bir topolojinin parçası olabilir.



Şekil 47. z/OS üzerinde çalışan MFT araçları, koordinasyon kuyruk yöneticisinin IBM MQ çoklu platformunda çalıştığı bir MFT topolojisinin parçası olabilir.

Hangi MFT komutlarının ve işlemlerinin hangi kuyruk yöneticisine bağlandığı başlıklı konuda, Managed File Transfer topolojisi için eşgüdüm kuyruğu yöneticisine bağlanan komutlar gösterilir. Bu komutlar z/OS üzerinde çalıştırılabilir ve daha sonra farklı bir altyapıda çalışan eşgüdüm kuyruğu yöneticisine bağlanabilir.



Şekil 48. **ftelistagents** gibi bazı komutlar, MFT topolojisi için doğrudan koordinasyon kuyruğu yöneticisine bağlanır.

Kaç ajanıma ihtiyacım var?

Aracılar verileri aktarırken iş yapar ve veri aktarma isteğinde bulunduğunuzda bir aracının adını belirtirsiniz.

Varsayılan olarak, bir aracı 25 gönderme ve 25 alma isteğini eşzamanlı olarak işleyebilir. Bu süreçleri yapılandırabilirsiniz. Ek bilgi için bkz. [Managed File Transfer yapılandırma seçenekleri z/OS](#).

Aracı meşgulse, iş kuyruğa alınır. Bir isteği işleme süresi birden çok etkene bağlıdır; örneğin, gönderilecek veri miktarı, ağ bant genişliği ve ağ üzerindeki gecikme.

İşi koşturarak işlemek için birden çok aracıya sahip olmak isteyebilirsiniz.

Ayrıca, bir aracının hangi kaynaklara erişebileceğini denetleyebilirsiniz; bu nedenle, bazı araçların sınırlı bir veri alt kümesiyle çalışmasını isteyebilirsiniz.

Farklı önceliğe sahip istekleri işlemek istiyorsanız, birden çok aracı kullanabilir ve işlerin önceliğini ayarlamak için iş yükü yöneticisini kullanabilirsiniz.

Aracıları çalıştırma

Genellikle araçlar uzun süre çalışan işlemlerdir. Süreçler, toplu kipte çalışan işler ya da başlatılan görevler olarak sunulabilir.

z/OS Managed File Transfer planlaması-güvenlikle ilgili önemli noktalar

z/OS üzerinde Managed File Transfer (MFT) programını çalıştırmak için sisteminizde hangi güvenlik konularına gereksinim duyacağınız konusunda yol gösterici olarak bu konuyu kullanın.

Durumu

MFT yapılandırması ve MFT işlemi için kullanılacak kullanıcı kimliklerini belirlemeniz gerekir.

Aktardığınız dosyaları ya da kuyrukları ve hangi kullanıcı kimliklerinin MFT 'ye aktarma istekleri göndereceğini belirlemeniz gerekir.

Aracıları ve kaydediciyi özelleştirdiğinizde, MFT hizmetlerini çalıştırmasına ya da MFT yönetimini yapmasına izin verilen kullanıcı grubunu belirtirsiniz.

MFT 'yi özelleştirmeye başlamadan önce bu grubu ayarlamamız gerekir. MFT IBM MQ kuyruklarını kullandığından, kuyruk yöneticisinde güvenliğiniz etkinleştirildiyse, MFT aşağıdaki kaynaklara erişmenizi gerektirir:

Çizelge 26. MQADMIN kaynak sınıfı	
Ad	Erişim gereklidir
QUEUE.SYSTEM.FTE.EVENT.agent_name	Güncelle
QUEUE.SYSTEM.FTE.COMMAND.agent_name	Güncelle
CONTEXT.SYSTEM.FTE.COMMAND.agent_name	Güncelle
QUEUE.SYSTEM.FTE.STATE.agent_name	Güncelle
QUEUE.SYSTEM.FTE.DATA.agent_name	Güncelle
QUEUE.SYSTEM.FTE.REPLY.agent_name	Güncelle
QUEUE.SYSTEM.FTE.AUTHAGT1.agent_name	Güncelle
QUEUE.SYSTEM.FTE.AUTHTRN1.agent_name	Güncelle
QUEUE.SYSTEM.FTE.AUTHOPS1.agent_name	Güncelle
QUEUE.SYSTEM.FTE.AUTHSCH1.agent_name	Güncelle
QUEUE.SYSTEM.FTE.AUTHMON1.agent_name	Güncelle
QUEUE.SYSTEM.FTE.AUTHADM1.agent_name	Güncelle

Çizelge 27. MQQUEUE kaynak sınıfı	
Ad	Erişim gereklidir
SYSTEM.FTE.AUTHAGT1.agent_name	Güncelle
SYSTEM.FTE.AUTHTRN1.agent_name	Güncelle
SYSTEM.FTE.AUTHOPS1.agent_name	Güncelle
SYSTEM.FTE.AUTHSCH1.agent_name	Güncelle
SYSTEM.FTE.AUTHMON1.agent_name	Güncelle

Aktarma isteğinde bulunan kullanıcının dosya sisteminin hangi bölümlerine erişebileceğini belirlemek için kullanıcı çalışma yeri kullanımını kullanabilirsiniz.

Kullanıcı korumalı alanı kullanımını etkinleştirmek için, kısıtlamak istediğiniz aracı için `userSandboxes=true` deyimini `agent.properties` dosyasına ekleyin ve `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml` dosyasına uygun değerleri ekleyin.

Daha fazla bilgi için [Kullanıcı kum havuzlarıyla çalışma](#) başlıklı konuya bakın.

Bu kullanıcı kimliği `UserSandboxes.xml` dosyalarında yapılandırılır.

Bu XML dosyası, kullanıcı kimliği ya da kullanıcı kimliği * gibi bilgileri ve kullanılabilen (içerilebilen) ya da kullanılmayan (dışlanamayan) bir kaynak listesini içerir. Hangi kaynaklara erişebilen belirli kullanıcı kimliklerini tanımlamanız gerekir: Örneğin:

Çizelge 28. Belirli kaynaklara erişimle birlikte örnek kullanıcı kimliği			
Kullanıcı Kimliği	Erişim	İçerme ya da Dışlama	Kaynak
Yönetici *	<	Ekle	/home/user/**
Yönetici *	<	Dışla	/home/user/private/**

Çizelge 28. Belirli kaynaklara erişimle birlikte örnek kullanıcı kimliği (devamı var)

Kullanıcı Kimliği	Erişim	İçerme ya da Dışlama	Kaynak
Sysprog (Sistem prog)	<	Ekle	/home/user/**
Yönetici *	<	Ekle	Application.reply.queue

Notlar:

1. type=queue belirtilirse, kaynak bir kuyruk adı ya da queue@mqmrolur.
2. Kaynak //ile başlıyorsa, kaynak bir veri kümeidir; aksi takdirde kaynak, z/OS UNIXiçindeki bir dosyadır.
3. Kullanıcı kimliği MQMD yapısındaki kullanıcı kimliğidir; bu nedenle, iletiyi gerçekten koyan kullanıcı kimliğini yansıtmayabilir.
4. Yerel kuyruk yöneticisindeki istekler için MQADMIN CONTEXT.* kullanabilirsiniz bu değeri hangi kullanıcıların ayarlayabileceğini sınırlamak için.
5. Uzak kuyruk yöneticisi üzerinden gelen istekler için, dağıtılmış kuyruk yöneticilerinin MQMD yapısında kullanıcı kimliğinin yetkisiz olarak ayarlanmasını önlemek için güvenliğin etkinleştirildiğini varsaymanız gerekir.
6. Bir Linux makinesindeki SYSPROG1 kullanıcı kimliği, z/OSüzerinde güvenlik denetimi için SYSPROG1 kullanıcı kimliğiyle aynıdır.

z/OS z/OS üzerinde IBM MQ Console ve REST API kullanılması planlanması

IBM MQ Console ve REST API , mqweb olarak bilinen bir WebSphere Liberty (Liberty) sunucusunda çalışan uygulamalardır. mqweb sunucusu başlatılan bir görev olarak çalışır. IBM MQ Console , kuyruk yöneticilerini yönetmek için bir web tarayıcısının kullanılmasına izin verir. REST API , uygulamaların kuyruk yöneticisi denetimi yapması ve ileti alışverişi gerçekleştirmesi için basit bir programlı arabirim sağlar.

Kuruluş ve yapılandırma dosyaları

z/OS UNIX System Services (z/OS UNIX) içinde mqweb sunucusunu çalıştırmak için gereken dosyaları kuracak IBM MQ for z/OS UNIX System Services Web Components özelliğini kurmanız gerekir. mqweb sunucusunu yapılandırmak ve yönetmek için z/OS UNIX ile ilgili bilgi sahibi olmanız gerekir.

IBM MQ for z/OS UNIX System Services Componentsürününü kurmaya ilişkin bilgi için bkz. [IBM MQ for z/OS Program Dizini PDF dosyaları](#) .

z/OS UNIX içindeki IBM MQ dosyaları, mqweb sunucusunun doğru çalışması için gerekli olan çeşitli öznitelikler ayarlanmış olarak kurulur. IBM MQ z/OS UNIX kuruluş dosyalarını kopyalamanız gerekiyorsa (örneğin, bir sisteme IBM MQ kurduysanız ve IBM MQ komutunu farklı bir sistemde çalıştırdıysanız), kuruluş sırasında oluşturulan IBM MQ ZFS ' yi kopyalamanız ve yalnızca hedefe bağlamanız gerekir. Dosyaların başka yollarla kopyalanması bazı dosya özniteliklerinin kaybolmasına neden olabilir.

mqweb sunucusunu yaratırken bir Liberty kullanıcı dizininin yerine karar vermeniz ve bu dizini yaratmanız gerekir. Bu dizin, yapılandırma ve günlük dosyalarını içerir ve konum /var/mqm/mqwebile benzer olabilir.

Farklı düzeylerde kuyruk yöneticileriyle IBM MQ Console ve REST API kullanılması

REST API , yalnızca REST APIçalıştıran mqweb sunucusu ile aynı Sürüm, Yayın ve Değişiklik (VRM) üzerinde çalışan kuyruk yöneticileriyle doğrudan etkileşimde bulunabilir. Örneğin, IBM MQ 9.3.0 REST API yalnızca IBM MQ 9.3.0adresindeki yerel kuyruk yöneticileriyle doğrudan etkileşimde bulunabilir ve IBM MQ 9.2.5 REST API yalnızca IBM MQ 9.2.5adresindeki yerel kuyruk yöneticileriyle doğrudan etkileşimde bulunabilir.

Bir ağ geçidi kuyruk yöneticisini yapılandırarak, kuyruk yöneticisini mqweb sunucusundan farklı bir sürümde denetlemek için REST API kullanabilirsiniz. Ancak, ağ geçidi kuyruk yöneticisi olarak işlev görmesi için mqweb sunucusu ile aynı sürümde en az bir kuyruk yöneticisine gereksinim duyarsınız. Daha fazla bilgi için bkz. [REST API kullanarak uzaktan yönetim](#).

IBM MQ Console , IBM MQ Console ile aynı sürümde çalışan yerel kuyruk yöneticilerini yönetmek için kullanılabilir. **V 9.3.0** IBM MQ 9.3.0 olanağından, uzak bir sistemde çalışan bir kuyruk yöneticisini ya da IBM MQ Console' un farklı bir sürümünü denetlemek için de IBM MQ Console olanağını kullanabilirsiniz. Daha fazla bilgi için, bkz. [IBM MQ Console ürününe uzak kuyruk yöneticisi eklenmesi](#).

Veri Taşıma

Yalnızca bir kuyruk yöneticiniz varsa, mqweb sunucusunu tek bir başlatılan görev olarak çalıştırabilir ve kuyruk yöneticinizi geçirirken kullandığı kitaplıkları değiştirebilirsiniz.

Birden çok kuyruk yöneticiniz varsa, geçiş sırasında farklı adlarla başlatılan görevleri kullanarak mqweb sunucularını farklı sürümlerde başlatabilirsiniz. Bu isimler istediğiniz herhangi bir isim olabilir. Örneğin, bir IBM MQ 9.1.0 mqweb sunucusunu MQWB0910 adlı başlatılan bir görevi ve IBM MQ 9.0.5 mqweb sunucusunu MQWB0905 adlı başlatılan bir görevi kullanarak başlatabilirsiniz.

Daha sonra kuyruk yöneticilerini bir sürümden daha sonraki bir sürüme geçirdiğinizde, kuyruk yöneticileri daha sonraki sürüm için mqweb sunucusunda kullanılabilir olur ve daha önceki sürüm için mqweb sunucusunda artık kullanılamaz.

Tüm kuyruk yöneticilerini sonraki sürüme geçirdikten sonra, önceki sürüm için mqweb sunucusunu silebilirsiniz.

HTTP kapıları

mqweb sunucusu HTTP için en çok iki kapı kullanır:

- Varsayılan değeri 9443 olan HTTPS için bir değer.
- HTTP için bir tane. HTTP varsayılan olarak etkinleştirilmez, ancak etkinleştirilirse 9080 varsayılan değerine sahiptir.

Varsayılan kapı değerleri kullanıyorsa, diğer kapıları ayırmanız gerekir. Birden çok IBM MQ sürümü için eşzamanlı olarak çalışan birden çok mqweb sunucunuz varsa, her sürüm için ayrı kapılar ayırmanız gerekir. mqweb sunucusunun kullandığı kapıları ayarlama hakkında daha fazla bilgi için bkz. [HTTP ve HTTPS kapılarını yapılandırma](#).

Bir kapıya ilişkin bilgileri görüntülemek için aşağıdaki TSO komutunu kullanabilirsiniz:

```
NETSTAT TCP tcpip (PORT portNumber)
```

Burada *tcpip* , TCP/IP adres alanının adıdır ve *portNumber* , ilgili bilgilerin görüntüleneceği kapının numarasını belirtir.

Güvenlik-mqweb sunucusu başlatılıyor

mqweb sunucusu kullanıcı kimliği için bazı yetkiler gerekiyor. Daha fazla bilgi için [mqweb sunucusu tarafından başlatılan görev kullanıcı kimliğinin gerektirdiği yetkibaşlıklı konuya](#) bakın.

Güvenlik- IBM MQ Console ve REST API kullanılarak

IBM MQ Console ve REST API kullandığınızda, yapılandırılan bir kayıt dosyasında bulunan bir kullanıcı olarak kimlik doğrulaması yapmanız gerekir. Bu kullanıcılara, kullanıcıların gerçekleştirebileceği işlemleri belirleyen belirli roller atanır. Örneğin, messaging REST API ögesini kullanmak için bir kullanıcıya MQWebUser rolü atanmalıdır. IBM MQ Console ve REST API için kullanılabilir roller ve bu rollerin izin vereceği erişim hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API üzerindeki roller](#).

IBM MQ Console ve REST API için güvenliđi yapılandırma hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API güvenlik](#).

Özel notlar

Bu belge, ABD'de kullanıma sunulan ürünler ve hizmetler için hazırlanmıştır.

IBM, bu belgede sözü edilen ürün, hizmet ya da özellikleri diğer ülkelerde kullanıma sunmayabilir. Bulduğunuz yerde kullanıma sunulan ürün ve hizmetleri yerel IBM müşteri temsilcisinden ya da çözüm ortağınızdan öğrenebilirsiniz. Bir IBM ürün, program ya da hizmetine gönderme yapılması, açık ya da örtük olarak yalnızca o IBM ürünü, programı ya da hizmetinin kullanılabilirliğini göstermez. Aynı işlevi gören ve IBM'in fikri mülkiyet haklarına zarar vermeyen herhangi bir ürün, program ya da hizmet de kullanılabilir. Ancak, IBM dışı ürün, program ya da hizmetlerle gerçekleştirilen işlemlerin değerlendirilmesi ve doğrulanması kullanıcının sorumluluğundadır.

IBM'in, bu belgedeki konularla ilgili patentleri ya da patent başvuruları olabilir. Bu belgenin size verilmiş olması, patentlerin izinsiz kullanım hakkının da verildiği anlamına gelmez. Lisansla ilgili sorularınızı aşağıdaki adrese yazabilirsiniz:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Çift byte (DBCS) bilgilerle ilgili lisans soruları için, ülkenizdeki IBM'in Fikri Haklar (Intellectual Property) bölümüyle bağlantı kurun ya da sorularınızı aşağıda adrese yazın:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japonya

İzleyen paragraf, bu tür kayıt ve koşulların, yasalarıyla bağdaşmadığı ülkeler ya da bölgeler için geçerli değildir: IBM BU YAYINI, "OLDUĞU GİBİ", HİÇBİR KONUDA AÇIK YA DA ÖRTÜK GARANTİ VERMEKSİZİN SAĞLAMAKTADIR; TİCARİ KULLANIMA UYGUNLUK AÇISINDAN HER TÜRLÜ GARANTİ VE BELİRLİ BİR AMACA UYGUNLUK İDDİASI AÇIKÇA REDDEDİLİR. Bazı ülkeler bazı işlemlerde garantinin açık ya da örtük olarak reddedilmesine izin vermez; dolayısıyla, bu bildirim sizin için geçerli olmayabilir.

Bu yayın teknik yanlışlar ya da yazım hataları içerebilir. Buradaki bilgiler üzerinde düzenli olarak değişiklik yapılmaktadır; söz konusu değişiklikler sonraki basımlara yansıtılacaktır. IBM, önceden bildirimde bulunmaksızın, bu yayında açıklanan ürünler ve/ya da programlar üzerinde iyileştirmeler ve/ya da değişiklikler yapabilir.

Bu belgede IBM dışı Web sitelerine yapılan göndermeler kullanıcıya kolaylık sağlamak içindir ve bu Web sitelerinin onaylanması anlamına gelmez. Bu Web sitelerinin içerdiği malzeme, bu IBM ürününe ilişkin malzemenin bir parçası değildir ve bu tür Web sitelerinin kullanılmasının sorumluluğu size aittir.

IBM'e bilgi ilettiğinizde, IBM bu bilgileri size karşı hiçbir yükümlülük almaksızın uygun gördüğü yöntemlerle kullanabilir ya da dağıtabilir.

(i) Bağımsız olarak yaratılan programlarla, bu program da içinde olmak üzere diğer programlar arasında bilgi değiş tokuşuna ve (ii) değiş tokuş edilen bilginin karşılıklı kullanımına olanak sağlamak amacıyla bu program hakkında bilgi sahibi olmak isteyen lisans sahipleri şu adrese yazabilirler:

IBM Corporation
Yazılım Birlikte Çalışabilirlik Koordinatörü, Bölüm 49XA
3605 Karayolu 52 N
Rochester, MN 55901
U.S.A.

Bu tür bilgiler, ilgili kayıt ve koşullar altında ve bazı durumlarda bedelli olarak edinilebilir.

Bu belgede açıklanan lisanslı program ve bu programla birlikte kullanılacak tüm lisanslı malzeme, IBM tarafından IBM Müşteri Sözleşmesi, IBM Uluslararası Program Lisans Sözleşmesi ya da taraflar arasında yapılan herhangi bir eşdeğer sözleşmenin koşulları kapsamında sağlanır.

Burada belirtilen performans verileri denetimli bir ortamda elde edilmiştir. Bu nedenle, başka işletim ortamlarında çok farklı sonuçlar alınabilir. Bazı ölçümler geliştirilme düzeyindeki sistemlerde yapılmıştır ve bu ölçümlerin genel kullanıma sunulan sistemlerde de aynı olacağı garanti edilemez. Ayrıca, bazı sonuçlar öngörü yöntemiyle elde edilmiş olabilir. Dolayısıyla, gerçek sonuçlar farklı olabilir. Bu belgenin kullanıcıları, kendi ortamları için geçerli verileri kendileri doğrulamalıdır.

IBM dışı ürünlerle ilgili bilgiler, bu ürünleri sağlayan firmalardan, bu firmaların yayın ve belgelerinden ve genel kullanıma açık diğer kaynaklardan alınmıştır. IBM bu ürünleri sinamamıştır ve IBM dışı ürünlerle ilgili performans doğruluğu, uyumluluk gibi iddiaları doğrulayamaz. IBM dışı ürünlerin yeteneklerine ilişkin sorular, bu ürünleri sağlayan firmalara yöneltilmelidir.

IBM'in gelecekteki yönelim ve kararlarına ilişkin tüm bildirimler değişebilir ve herhangi bir duyuruda bulunulmadan bunlardan vazgeçilebilir; bu yönelim ve kararlar yalnızca amaç ve hedefleri gösterir.

Bu belge, günlük iş ortamında kullanılan veri ve raporlara ilişkin örnekler içerir. Örneklerin olabildiğince açıklayıcı olması amacıyla kişi, şirket, marka ve ürün adları belirtilmiş olabilir. Bu adların tümü gerçek dışıdır ve gerçek iş ortamında kullanılan ad ve adreslerle olabilecek herhangi bir benzerlik tümüyle rastlantıdır.

YAYIN HAKKI LİSANSI:

Bu belge, çeşitli işletim platformlarında programlama tekniklerini gösteren, kaynak dilde yazılmış örnek uygulama programları içerir. Bu örnek programları, IBM'e herhangi bir ödemede bulunmadan, örnek programların yazıldığı işletim altyapısına ilişkin uygulama programlama arabirimiyle uyumlu uygulama programlarının geliştirilmesi, kullanılması, pazarlanması ya da dağıtılması amacıyla herhangi bir biçimde kopyalayabilir, değiştirebilir ve dağıtabilirsiniz. Bu örnekler her koşul altında tüm ayrıntılarıyla sinanmamıştır. Dolayısıyla, IBM bu programların güvenilirliği, bakım yapılabilirliği ya da işlevleri konusunda açık ya da örtük güvence veremez.

Bu bilgileri elektronik kopya olarak görüntülediyseniz, fotoğraflar ve renkli resimler görünmeyebilir.

Programlama arabirimi bilgileri

Sağlandıysa, programlama arabirimi bilgileri, bu programla birlikte kullanılmak üzere uygulama yazılımı oluşturmanıza yardımcı olmak amacıyla hazırlanmıştır.

Bu kitapta, müşterinin WebSphere MQ hizmetlerini elde etmek üzere program yazmasına olanak sağlayan amaçlanan programlama arabirimlerine ilişkin bilgiler yer alır.

Ancak, bu bilgiler tanılama, değiştirme ve ayarlama bilgilerini de içerebilir. Tanılama, değiştirme ve ayarlama bilgileri, uygulama yazılımlarınızda hata ayıklamanıza yardımcı olur.

Önemli: Bu tanılama, değiştirme ve ayarlama bilgilerini bir programlama arabirimi olarak kullanmayın; bu bilgiler değişebilir.

Ticari Markalar

IBM, IBM logosu, ibm.com, IBM Corporation 'ın dünya çapında birçok farklı hukuk düzeninde kayıtlı bulunan ticari markalarıdır. IBM ticari markalarının güncel bir listesine Web üzerinde "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml (Telif hakkı ve ticari marka bilgileri) başlıklı konudan ulaşılabilir. Diğer ürün ve hizmet adları IBM'in veya diğer şirketlerin ticari markaları olabilir.

Microsoft ve Windows, Microsoft Corporation firmasının ABD'de ve/ya da diğer ülkelerdeki markalarıdır.

UNIX, The Open Group şirketinin ABD ve diğer ülkelerdeki tescilli ticari markasıdır.

Linux, Linus Torvalds'ın ABD ve/ya da diğer ülkelerdeki tescilli ticari markasıdır.

Bu ürün, Eclipse Project (<https://www.eclipse.org/>) tarafından geliştirilen yazılımları içerir.

Java ve Java tabanlı tüm markalar ve logolar, Oracle firmasının ve/ya da iřtiraklerinin markaları ya da tescilli markalarıdır.



Parça numarası:

(1P) P/N: