



IBM Software Group

チャネルを常に活動状態にするには (Keeping Channels Up and Running)

Morag Hughson, Paul Clarke
hughson@uk.ibm.com, paulg_clarke@uk.ibm.com

WebSphere. software

@business on demand software

© 2003 IBM Corporation

V1.0.1

IBM Software Group | WebSphere software



はじめに

この書は WebSphere MQ おび MQSeries に関する IBM の専門職、これらの製品ご使用お客様の環境目標おる経験基して、IBM の推して提するものです。この書が提する内容採れるかどうかは、お客様自身判断なます。WebSphere MQ には、個の WebSphere MQ の実環境よって、事異なる可能性あるエリアがいくつかあり、専門家意が分かる場合があります。

N
O
T
E
S

この書お読んだら、る皆、WebSphere MQ に関する理が深れば、いその提採採用べきを自分で判断できるようになり、また、ご使用環境最最適の判断ができるようになるでしょう。

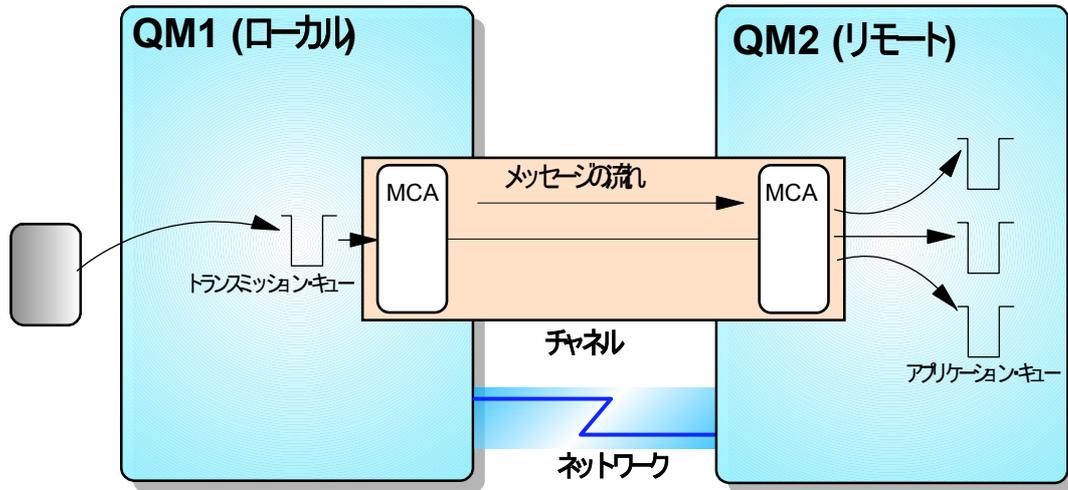
本書は、システムアドミニストレーター、おび WebSphere MQ ネットワークのサポートを担当る方々使用することを想定しています。本書よって、以下のようなメリットを提しと考ておます。

- 管理プロセスの一貫性
- アプリケーション可用性の最大化
- 初者問題 やすい問題の迅速な支援
- WebSphere MQ の専門家めず皆様の初期難題の支援
- WebSphere MQ プロジェクト全般のスムなスタートおび成功を支援

本書提される情報の多は MQSeries おび WebSphere MQ のバージョンに關係します。ご使用のバージョンのマニュアルを参照、リレーターがサポートされているかどうか確認下さい。

© 2003 IBM Corporation

チャネルアーキテクチャ



チャネルアーキテクチャ (説明)

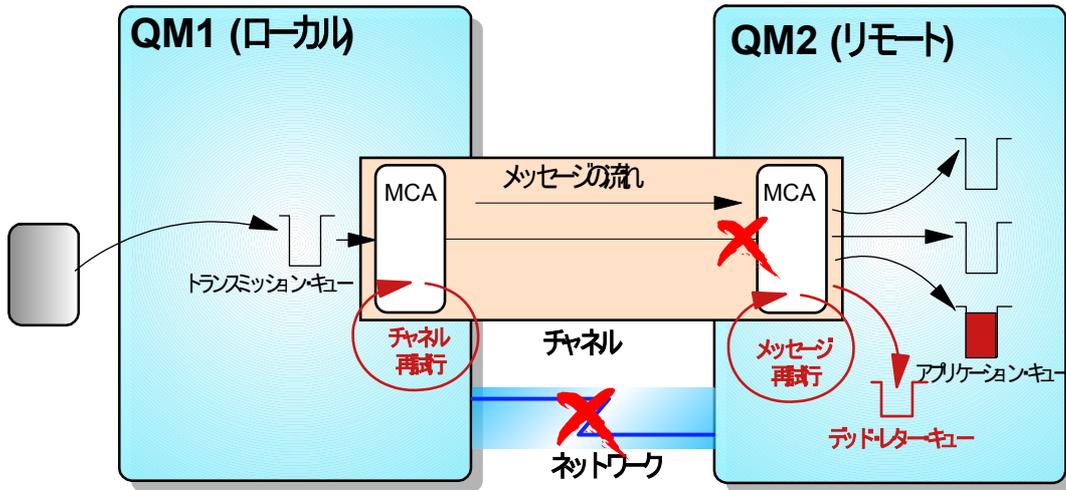
この図は WebSphere MQ のチャネルアーキテクチャを表しています。このルセレーション全体を通じて、この図を使用してチャネルに関するさまざまなトピックを説明します。

N
O
T
E
S

この図は、アプリケーションがリモートキュー定義を介して、または明示的にアドレスを指定してメッセージを書き込む状態を表しています。メッセージはトランミッションキューに書き入れ、リモートシステムに送信されます。この図は、QM1 がローカルシステムで、QM2 がリモートシステムです。トランミッションキューを処理する送信側 MCA はキューからメッセージを取り出し、チャネル定義で指定された通信プロトコルを介してメッセージを送ります。受信側 MCA はメッセージを受信し、適切なターゲットアプリケーションキューにメッセージを書き込みます。この MCA のペアをチャネルと呼びます。

この図は大まかに示していますが、どのような問題発生する可能性がありますか？

トラブルの起るポイント



- チャンネルの終了 - リモートのキューマネージャの停止
- メッセージ送不能 - キューが無効または満杯
- ネットワーク障害
- パフォーマンス

トラブルの起るポイント (説明)

このスライドは、今日の内容の目次のようなものです。トラブルの発生する可能性のあるいくつかのポイントの概要を簡単に理解した後、それぞれについて詳細に説明します。

チャンネルが終了することがあります。これは、例えば、リモートキューマネージャの停止などが原因として考えられます。

N
O
T
E
S

このような問題の場合、チャンネルイニシエーターを使用してチャンネルを再試行することにより、リカバリできます。

メッセージ送不能な場合もあります。これは、アプリケーションエラー、例外指定されたキューが無効であるなどが原因である場合があります。

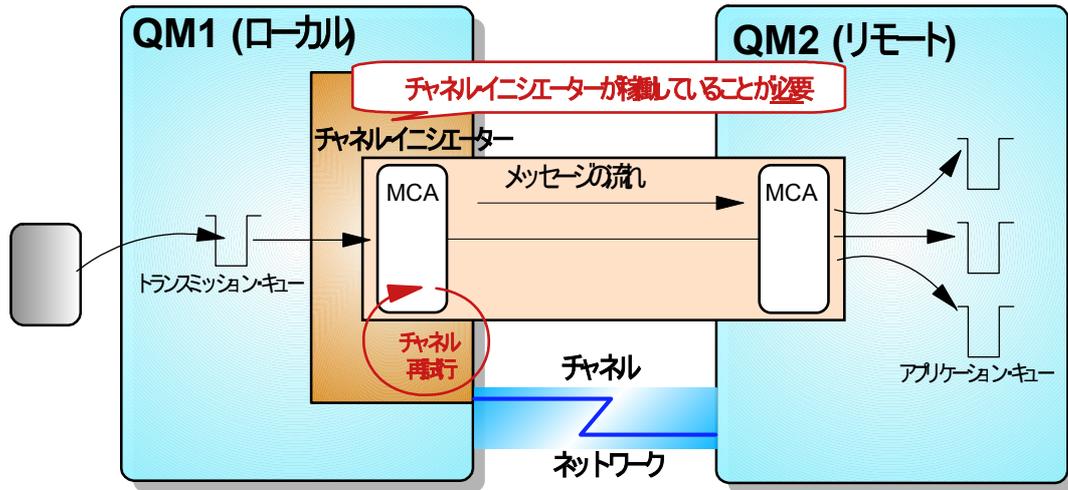
また、キューが満杯であるなどのシステム問題の原因である可能性があります。

このような問題の場合、最終で続行するには、デッドレターキューを使用するか、メッセージ再試行機能を利用します。

ネットワーク障害が発生する可能性があります。障害のタイプに応じて、ご使用のチャンネルに適した方法で、障害を検出しリカバリします。

パフォーマンスが問題となる場合があるため、いくつかのパフォーマンス上の考慮事項についても説明します。

チャネル再試行



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

チャネル再試行(説明)

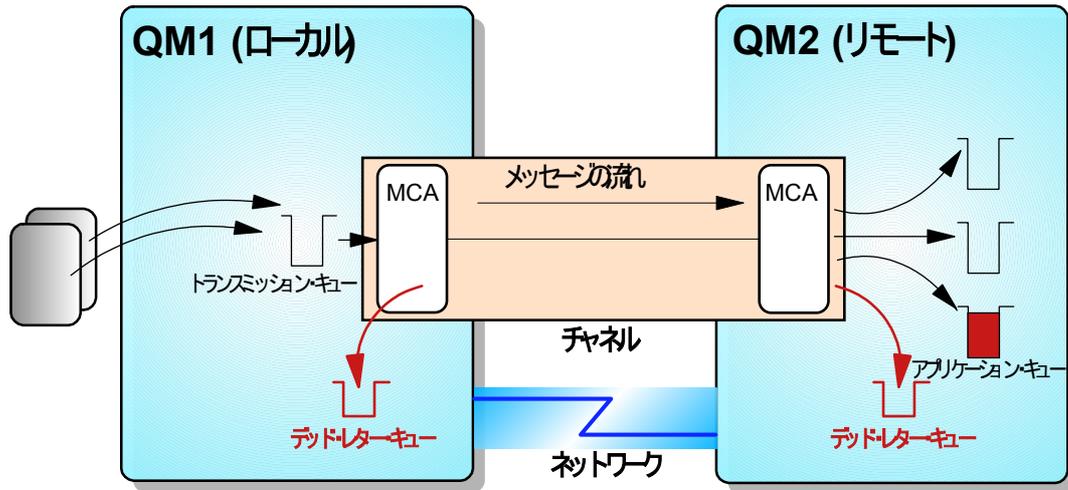
NOTES

チャネルが確立不能であるようにする「問題のない」メカニズムは WebSphere MQ が、障害後チャネルを再構築できるようにすることです。これをチャネル再試行(Channel Retry)と呼びます。チャネル再試行は、送信側チャネルタイプ、つまり送信チャネル、サーバーチャネル、またはクラスター送信チャネルで使用できます。すべての送信チャネルは、適切な再試行値を使用して構築する必要があります。また、再試行メカニズムを使用するには、チャネルイニシエーターが稼働している必要があります。再試行値は、ネットワークの問題が解消するのに十分な時間を指定する必要があります。

例えば 2 分間の再試行時間が決まるとはなりません。使用できる再試行値は SHORTRTY (短再試行) と LONGRTY (長再試行) の 2 つあります。SHORTRTY は、再試行時間が短く頻度が多い場合、つまり何かの一時停止のために送信側の再構築を試みる場合に使用されます。LONGRTY は、再試行時間が長く頻度が少ない場合、つまり、チャネル停止の原因が、単なる一時停止とは異なる問題がある場合に使用されます。LONGRTY の時間は、障害時に十分な値を指定する必要があります。

メッセージの送込が成功するまで、一連の再試行が終了してしまった場合、作業チャネルを再構築しなければならなくなるためです。

デッドレターキュー



■ 以下のようなメッセージの保管場所

- ▶ 宛先不明
- ▶ 権限エラー
- ▶ サイズが大きい

デッドレターキュー (説明)

キューマネージャーにデッドレターキューを定義することを強く推奨します。

デッドレターキューが定義されていない場合、無名キュー名が指定された単独アプリケーションプログラミング・インターフェイスによって、チャネルが停止します。

複数の異なるアプリケーションが同じチャネルを使用する場合は、これは不明な状態です。

デッドレターキューは、チャネルのどちらの側にも使用される可能性があります。

送信側は、メッセージが大きすぎてチャネルが停止してしまう場合に、デッドレターキューが使用されます。

これはメッセージサイズが、チャネルに対して定義されたメッセージのサイズを超えた場合に発生します。この場合、メッセージはトランミッションキューに残り続けます。

受信側は、メッセージをターゲットキューに書き込む (PUT) ことができない場合に、デッドレターキューが使用されます。

これはおぼろげな理由ですが、一時的な問題がある場合があります。

メッセージ再試行リテーター (チャネルごとに定義) およびメッセージ再試行リテーターは、このようなケースに対応できるように設計されています。

メッセージ再試行リテーターは、チャネルのどちらの側にも、デッドレターキューの使用は、メッセージの転送に影響を与えます。

一連のメッセージのうちのいくつかの問題があった場合は、このメッセージはデッドレターキューに送られ、メッセージシーケンスの転送は停止します。

いくつかの「急ぎ」アプリケーションは、送信したときと同じ順序でメッセージが転送されることを要求します。

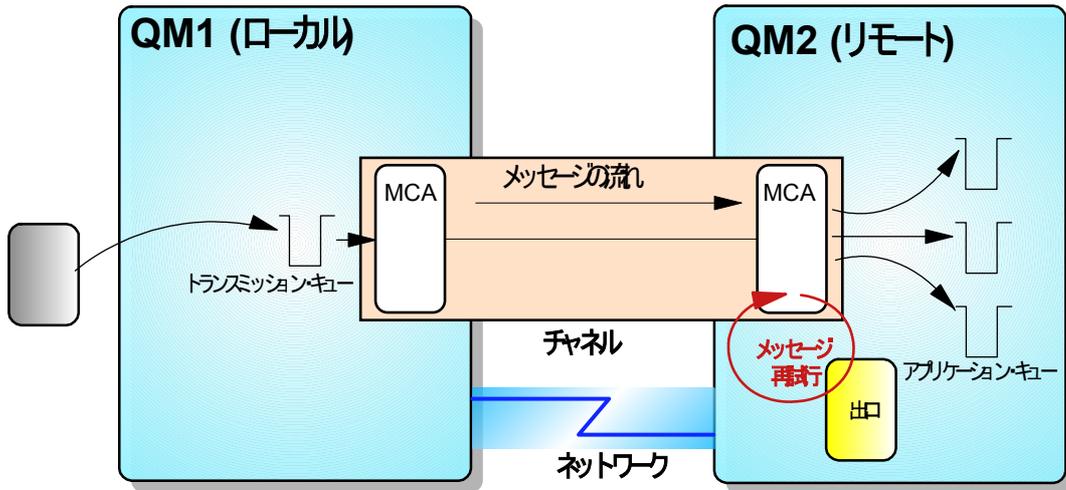
このようなアプリケーションは、デッドレターキューを定義しないことを要求します。

デッドレターキューを定義しないと、結果的にチャネルの悪いメッセージがチャネルを停止させると、すべてのアプリケーションがメッセージを受信できなくなります。

このため、インストール時にデッドレターキューを用意し、アプリケーションはメッセージが誤った順序で転送される場合にも対応できるように変更することを強く推奨します。

この変更は難しいものではないかもしれませんが、カウンターとして、相関IDを使用したり、2次キューの使用またはメッセージのグループの使用など、さまざまなテクニックがあります。

メッセージ再試行



DEF CHL
MRRTY(10) MRTMR(1000)
MREXIT(msg-retry-exit-name) MRDATA()



メッセージ再試行(説明)

メッセージ再試行(Message Retry)は、デッドレターキューを定義域に移動する場合は特に、使用することを検討してみてください。

メッセージ再試行メカニズムは、出口(EXIT) おいても出口なしでもどちらでも使用できます。

N

出口を使用しない場合、受取側MCAは、再試行インターバルごとに失敗した書き込みを単に再試行します。

再試行の間隔の値より再試行回数値は、チャンネル定義により決まります。

出口を使用する場合、出力レチンにはこれらの数値が渡りますが、値変更することができます。

O

メッセージ再試行メカニズムで出口を使用することにより、より高度な障害対応ができます。

例えば、メッセージをターゲットキューに書き込みできなかった場合に、メッセージをリダイレクトすることができます。

出口を使用することにより、再試行を実行するかどうかをメッセージごとに決定できます。

T

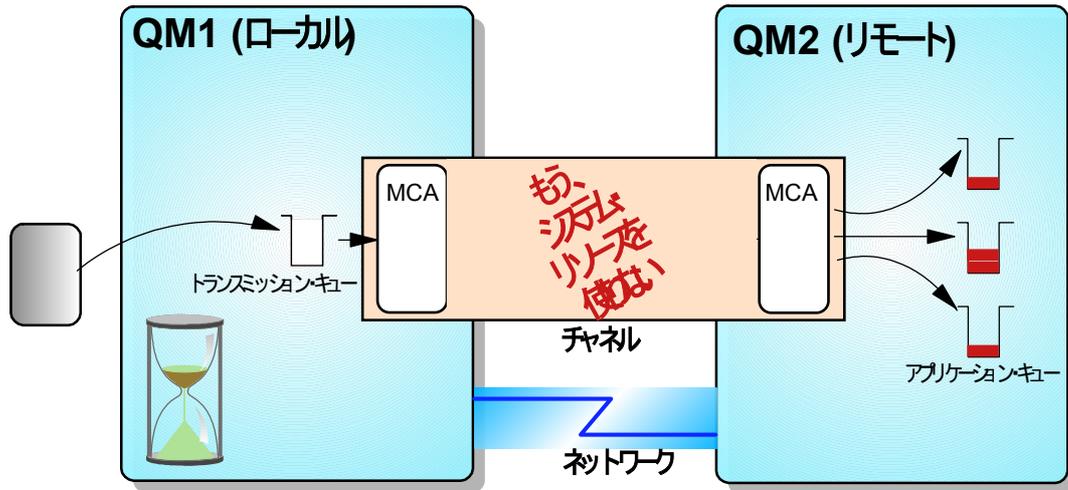
メッセージ再試行は現在 z/OS ではサポートされていません。

E

S



切断インターバル



DEF CHL DISCINT(6000)

- トリガーとの組み合わせ



切断インターバル(説明)

切断インターバル(Disconnect Interval)、すなわち DISCINT

は、送信側のチャネルに関するチャネル属性、送信するメッセージがない場合でもチャネルをどのくらいの時間維持しておくかを制御するのに使われます。

いくつかの基準によって、設定する値が変わります。

- 使用しているチャネル数およびシステムリソースの重要度
- 使用するネットワークの種類
- 通信セッションにかかる費用

望ましい値は、通信相手へのメッセージ送りの平均的間隔よりも長い時間です。つまり、その相手を送るべき仕事がない場合に、チャネルを切断することを意味します。

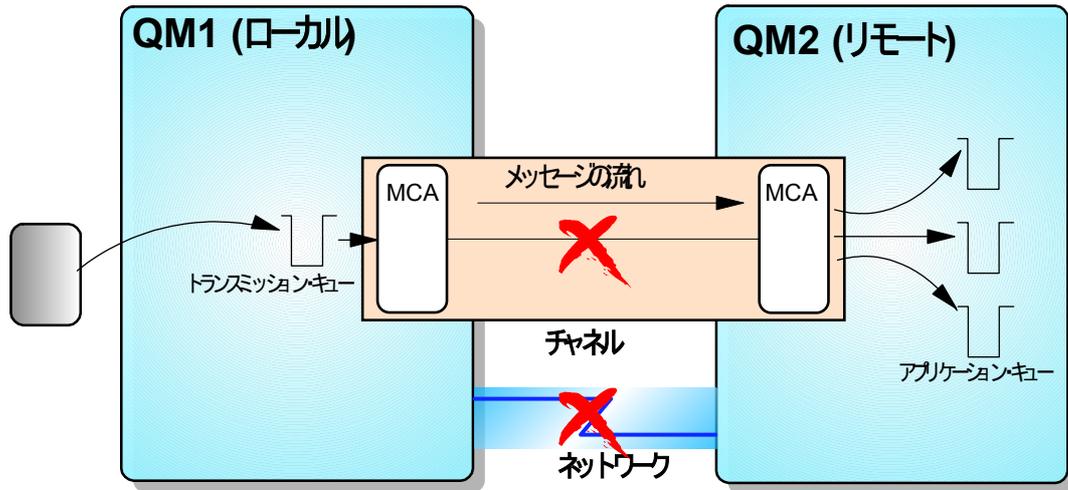
チャネルは切断インターバルを使用する場合、チャネルを再確立させるトリガーと組み合わせで使用する必要があります。

こうすることにより、チャネルは実行すべき仕事がない時は自動的に働いて、実行すべき仕事がない場合は自動的に終了します。

N
O
T
E
S



ネットワーク障害



- ハートビート
- TCP/IP キープアライブ
- MCA の切替



ネットワーク障害(説明)

チャネルにネットワーク障害がレポートされると、チャネルは終了します。ネットワーク障害がチャネルにレポートされないならば、チャネルは終了しません。この場合、チャネルはネットワーク障害を検知しない可能性があります。

NOTES

チャネルは、ネットワーク障害レポートを低レベルの通信プロトコルの関数に依存しています。

例えば、RECEIVER チャネルの外出先は TCP/IP の `recv()` コールに依存します。

`recv()` コールからリターンがない場合、チャネルは通信相手の接続が切れたことを認識しません。

MCA に関する以前の問題で、通信プロトコルが必ずしも適切なタイミングでエラーを戻さないため

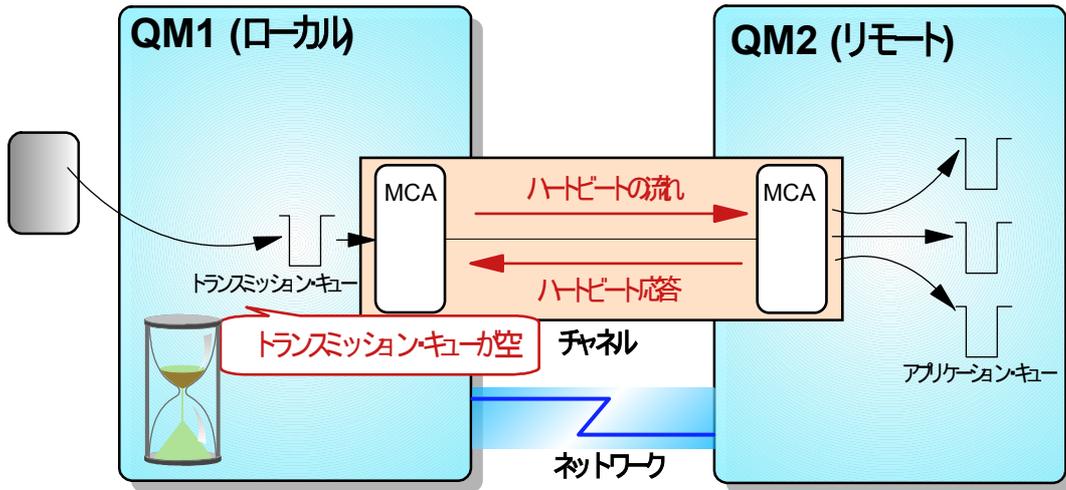
実際ネットワーク障害が発生した際、タイムリーな方法で障害を認識できなかったことがあります。

特にこの点に関しては TCP/IP よりも LU6.2 の方がるかに優れています (より複雑なプロトコルです)。

MCA のさまざまな機能拡張により、状況は大幅に改善されました。次に、そのうちのいくつかについて紹介します。



ハートビート



DEF CHL HBINT(300)

バッファの解放 ✓

キューのクローズ ✓

ハートビート (説明)

ハートビートインターバル(Heartbeat Interval) すなわち HBINT

は、送るメッセージがない場合に、チャンネルの受信側で問題ないかを定期的に検出する頻度を制御するチャンネル属性です。

値はチャンネルの単位で指定でき、チャンネルの単位で、使用すべき値はチャンネルの単位で指定して取り決めます。

双方の指定された値のうち小さい方のインターバルになります。片側が0 (ハートビートなし) が指定された場合、最小の値で、この値を選択します。

デフォルトでは、チャンネルのハートビートインターバルは300 秒5 分に設定されています。ネットワークの負荷の影響を受けないように、比較的高い値が設定されています。

実際は 60 秒程度が適している場合が多いでしょう。

ハートビートは、実際には、受信MCA が使用可能であることを検出するために、受信MCA に送る短いメッセージです (すなわち MQGET 中の CLNTCONN)。

これは、ハートビートのないものの数日間、チャンネルに他のアクティビティが実行されている場合のみ送ります。このメッセージで 2 つの問題が解決します。

まず送信MCA では、いままで MQGET を続けるので、ネットワークから直にエラーが戻ります。

また受信MCA では、キューマネージャが終了したのをチェックするなどの処理を行うから応答を送ることができます。

さらに、このチャンネルは常にメッセージを転送するもので、ストレージのキャッシュおよびキューを保持する必要はありません。

そのため、ハートビート時に、チャンネルは不要なストレージバッファやメッセージエリアを解放し、キャッシュ・キューをクローズします。

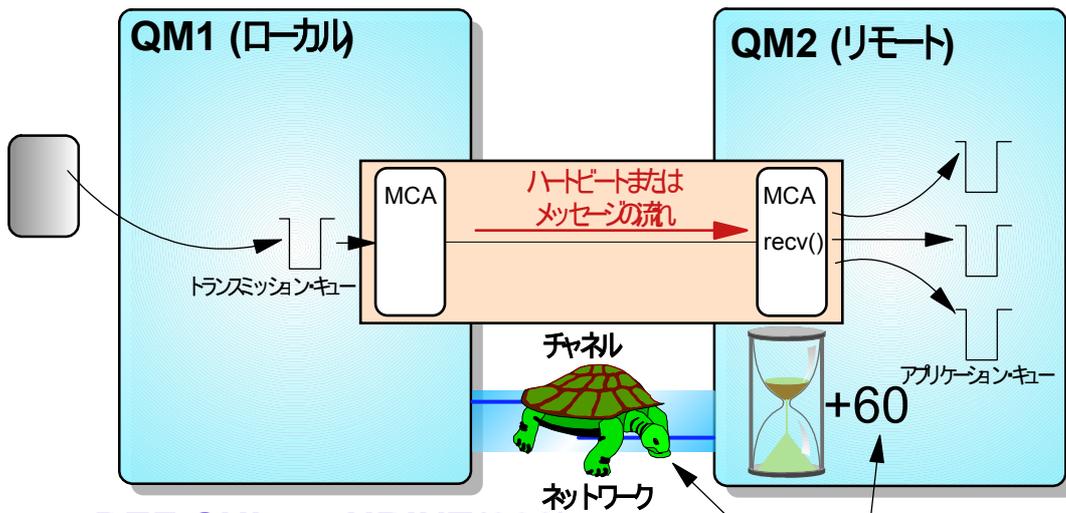
従ってパフォーマンスを考えると、ハートビート値を低くするほうが良いでしょう。

しかし、ハートビート値が小さくても、チャンネルがビジー状態が続く場合、ネットワークトラフィックを増大させるようなことはありません。

2 秒おきにチャンネルにメッセージを送るようなアプリケーションを考へてみてください。

3 秒おきにハートビート値を設定しては、常に送るべきメッセージがあるはずですから、ハートビートが送られることはありません。

受信帯タイムアウト



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

ネットワーク待時間を反映し



受信帯タイムアウト (説明)

先ほど説明したように、送信チャネルが定期的ハートビート・メッセージを送信するのに使用するハートビート・インターバルは、チャネルの確立に消費される値であるため、双方でその値を知っています。

N
O
T
E
S

ハートビートがない場合、メッセージの到着は確定的なものではないため、次のデータがチャネルに到着するのを受信は待たせません。

しかし、ハートビートがある場合、受信はハートビート・インターバル以内にメッセージまたはハートビートが到着することを「知っています」。

時間に達しない場合は、通信セッションが破断している可能性があります。そこで、これはTCP/IP ネットワーク障害検出の基礎として使用されます。

CLNTCONN チャネルの場合、このタイムアウトはMQI コマンドの応答等の間にも使用されます (SVRCONN がz/OS 上において、ハートビートのクライアント・リアントをサポートしない場合を除く)。

受信帯障害タイムアウトは、デフォルトではハートビート・インターバルが60秒未満の場合はハートビート・インターバルの2倍時間経過したとき、またはハートビート・インターバルが60秒以上の場合はハートビート・インターバル+60秒経過してもデータを受信しない場合に、通信障害が発生したと想定します。

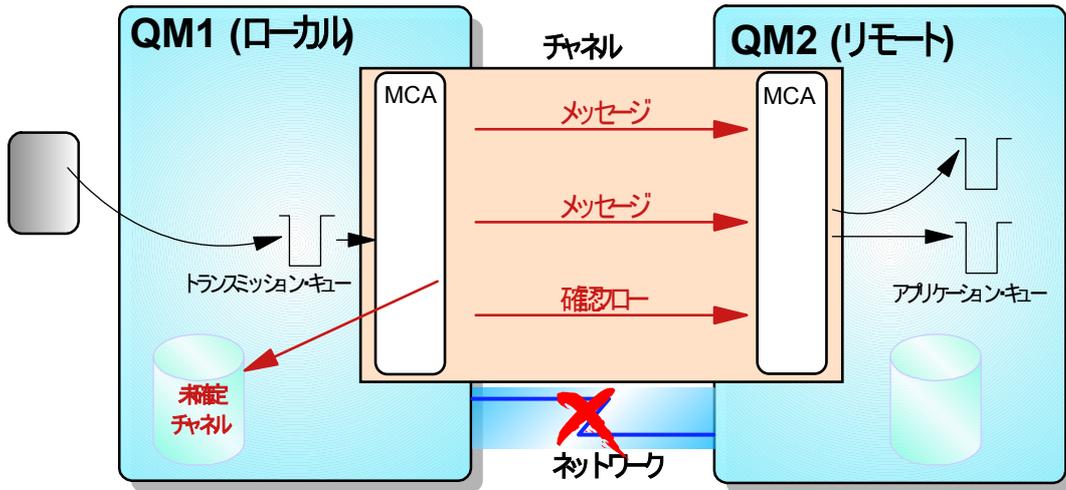
この数値は、実際には環境変数MQRCVBLKTOを変更して、分散プラットフォームでネットワークの待時間を反映するように調整できます。

MQRCVBLKMIN という環境変数もあります。この変数を使用すると、ハートビートに基づいてタイムアウトを算出する場合に使用する最小値を設定できます。

V5.3.1ではz/OSでタイムアウトの条件を設定するにはXPARM 属性のRCVTIME を使用して動作を設定します。最小値を設定するにはRCVTMIN を使用します。

従って、値が大きいと迅速に障害検出できますが、チャネルの負荷が低い場合は、ネットワーク・トラフィックおよびCPU リソースに関してコストが高くなります。

送りの保証



- 同期データが何枚ディスクに書込れる
 - ▶ ネットワーク障害後再送可能
 - ▶ 未確定状態のメッセージ群(パッチ)は、クラスタリング・アルゴリズムによる割り当てはできない

送りの保証(説明)

この図は、一群メッセージを送り込むチャネルの保証を示します。

MCA が一群メッセージの最後ディスクデータを書込方法に注意してください。これはメッセージ群回復能力が弱くなります。

N

書込れるデータは、すべてのメッセージを取り出たため、送りが使トランザクションのトランザクションID が含まれます。

送りが通ずる「確認(confirm)」フローが出されると、応答を受取るまで「未確定(indoubt)」状態になります。

つまり、チャネルの送りは、メッセージ群が正常に送られたかどうか分かりません。

O

もしこのフェーズで通信障害が発生すると、チャネルは未確定のまま終了します。

通信障害が再発する場合、その群に対して未確定であったことがデータストアから通知され、前のメッセージ群が送られたかどうかを今後のチャネルで確認します。

相対から送られた応答を使用して、チャネルはトランザクションキューのメッセージをミットするかどうかを判断します。

T

この同期データは DIS CHSTATUS(*) SAVED コマンドを発行することにより表示できます。チャネルの保証に、同じ値が表示されるはずです。

未確定で、チャネルを再構築すると、チャネルは自動的に未確定を解決します。しかし、同じ相対を通ずる場合に限ります。

E

チャネル属性を変更した場合、別のキューマネージャーがIP

アドレスを引き継いだ場合、または同じトランザクションキューを処理する別のチャネルが稼働した場合、チャネルは直ちに終了、

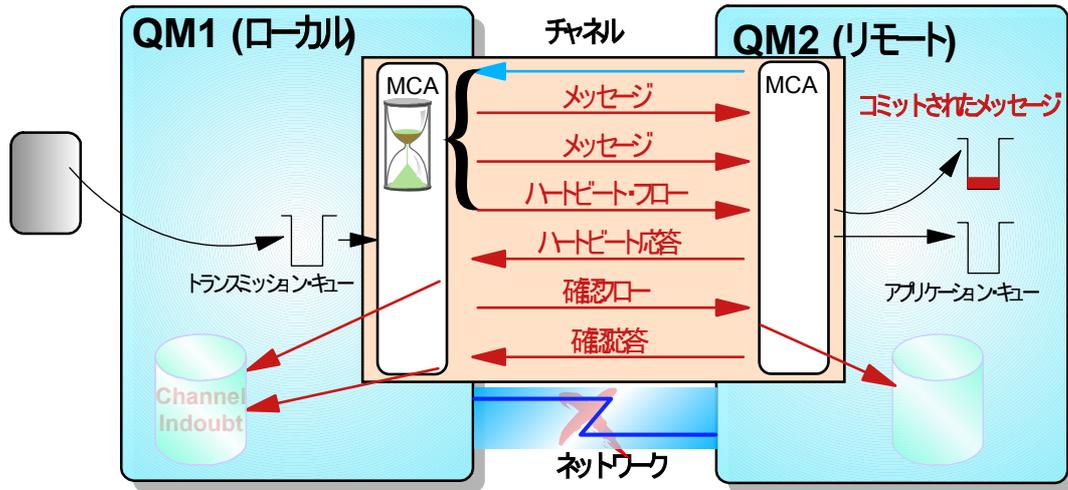
異なるキューマネージャーで未確定のままであることを知らせるメッセージを表示します。

S

ユーザは正しいキューマネージャーに対してチャネルを稼働し、RESOLVE CHANNEL コマンドを出して再度未確定を解決する必要があります。

この場合 COMMIT またはBACKOUT の正しいアクションを選択するためにDIS CHS(*) SAVED の結果を使用しなければならぬことに注意してください。

バッチハートビート



DEF CHL ... BATCHHB(500)

- 最後の通関からの経過時間がインターバルより長 場合 ハートビート・フローが送られる

ネットワークの安定性を反映へ

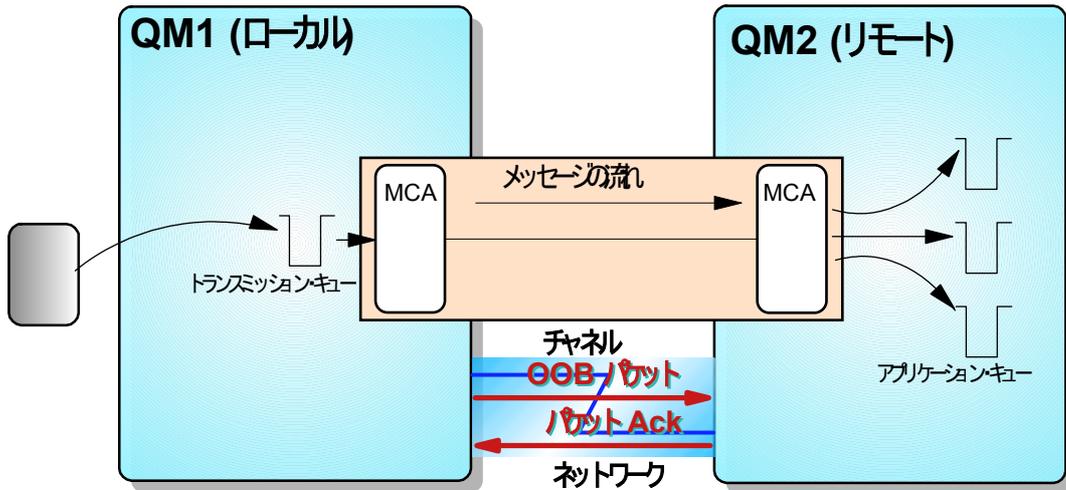


バッチハートビート (説明)

未確定終了するチャネルに関する問題は、クラスタリング・アルゴリズムで発生します。
 クラスタリング・チャネルが異常終了した場合、このチャネルによって送られる予定だったメッセージは、別のチャネルおよびターゲット・クラスター・キューの別のインスタンスに再割り当てされます。
 しかし、受信チャネルがメッセージを受取ったかどうか送信チャネルが確定できないメッセージ群の一部がある場合は、再割り当てされません。
 チャネルが未確定状態で終了する可能性を低くするために、チャネル・プロトコルに新たなフローが追加されました。このフローはバッチハートビートといいます。
 バッチハートビート・フローは、メッセージ群を送信した後送りますが、チャネルが自ら未確定状態として確認ローを送信する前です。
 チャネルの受信側から応答を受ければ、バッチメッセージが完了したことがかなり確信できます。
 ここで受信チャネルを終了して未確定状態にする非同期可能なウィンドウがあります。
 バッチハートビート・フローを使用しない場合は、このウィンドウはもっと大きなサイズ(最後に相対フローを受信した時点までのデータ・サイズ)になります。
 このバッチハートビート・フローを送信かどうかを判断するために、相対に最後フローを受信した時間に着目します。
 この時間、チャネルのBATCHHB パラメータ (BatchHeartbeat) で指定したインターバルより長 場合、フローを送ります。
 短 場合は送信する必要がありません。従って、指定するインターバルは、ネットワークの安定性を反映したものが最適です。このインターバルは、ミリ秒で指定します。

N
O
T
E
S

TCP/IP キーブライブ



- QM.INI **KEEPALIVE=YES**
 - XPARM **TCPKEEP=YES**
- DEF CHL ... KAINTE(360) (z/OS のみ)**

TCP/IP キーブライブ (説明)

NOTES

TCP/IP には、ネットワーク層を認識可能なキーブライブと呼ばれる独自のハートビートプロトコルがあります。キーブライブを使用することを推奨します。これは z/OS 上のみ、チャンネルリモーターとして構成可能で、チャンネルごとに個別タイムアウト値を設定することも、単体システム値を使用することもできます。このチャンネルリモーターは、使用に TCP/IP でキーブライブが有効になっていることが前提です。

キーブライブは、すべてのプラットフォームで、WebSphere MQ のキューマネージャベースで、qm.ini ファイルに相当するもので構成されます。この場合、タイマーおよびカウンタとして使用される実際の値が、そのマシン上の TCP/IP スタック全体に対して設定されます。

標準値は、ネットワーク層がデフォルトである 2 時間後とされています。この値は WebSphere MQ 環境にとってはおそく大きすぎますが、同一マシン上の他のアプリケーションもキーブライブを使用している場合、異なるアプリケーションの要求を満たす必要がある場合があります。お手持の TCP/IP マニュアルを参照して、キーブライブの構成方法を確認する必要があります。

AIX の場合は "no" コマンドを使用します。HP-UX の場合は "netttune" コマンドを使用します。Windows NT の場合は、レジストリを編集する必要があります。

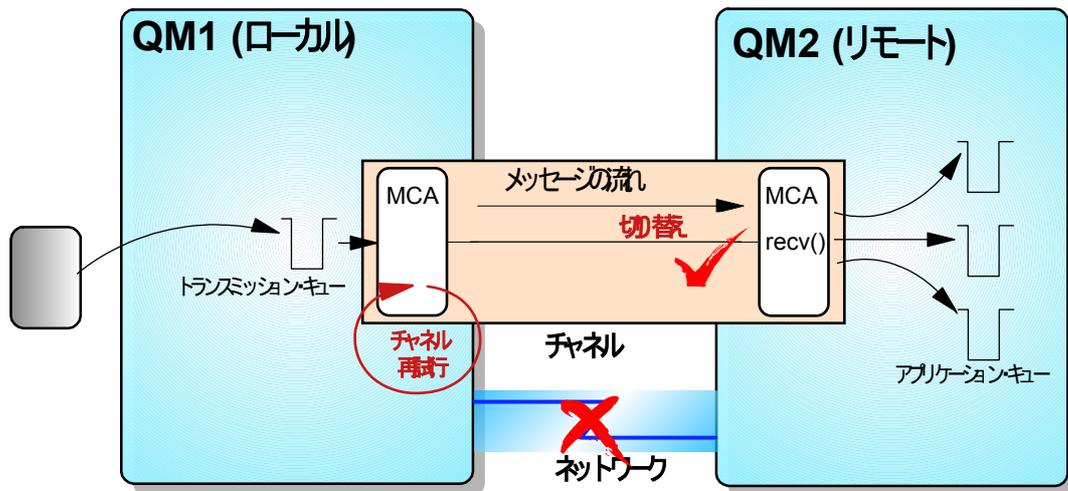
z/OS の場合は TCP/IP PROFILE データセットをアップグレードして、TCPCONFIG セクションで INTERVAL リモーターを追加変更します。
注 z/OS では TCP/IP APAR PQ75195 の適用を推奨します。

ハートビートおよび受信帯タイムアウトを使用すれば、チャンネルの補助アプリケーションをサポートしている場合は、キーブライブを使用する必要はありませんが、さらにキーブライブを使用することもできます。

ハートビートおよび送信帯タイムアウトをサポートしていない WebSphere MQ システムと通信する場合は、キーブライブを使用する必要があります。

SVRCONN チャンネルに対しては、クライアントハートビートが使用可能な場合でも、MQGET 中が使用されないため、キーブライブの使用を強く推奨します。

MCA の切替



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

MCA の切替 (説明)

例えば、通信エラーなどでチャネルが失敗した場合、チャネルの確立と同時に障害を検出するよう保証はありませぬ。

多くの場合はチャネルの送受信リソースを介してメッセージの送受信のため、送受信先障害を検出します。

通常、送受信ターゲットとなるキューマネージャへの再接続を試みます。

ターゲットマシンは接続されると、キューマネージャはそのロケーションからその確立済みのチャネルがすでに存在していると認識します。その結果、新たな接続は拒否されます。

AdoptNewMCA を構成することにより、新たな接続を確立して、同ネットワーク・アドレスおよび同キューマネージャより同じ確立済みのチャネルがすでに存在している場合は

既存のチャネルを終了し、新しいインスタンスで置換えるようキューマネージャに通知することができます。つまり、チャネルインスタンスは新たな接続によって「切替 (adopt)」されます。

QM.INI またはそれに相当するもの (z/OS では XPARMS を使用) の channels スタンプのさまざまな構成パラメータで、乗換の振舞いを多岐にわたって指定することができます。この方法で切替される可能性があるチャネルタイプのリストの切替を定義します。

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

ADOPTMCA=YES/NO (z/OS の場合)

「切替られた」チャネルを確立するまで即時終了の間の待ち時間のインターバルを設定します。

AdoptNewMCATimeout=60

チャネル名はオプションで指定する必要はありますが、さらにネットワーク・アドレスおよびリモートキューマネージャの確立済みのチャネルであることを指定することができます。

これにより、ネットワーク上の別のマシンからの不正な接続、すでに確立済みのチャネルを乗取ることを防ぐことができます。

「切替られた」接続は新たな接続間で一致する必要がある属性を設定します。

AdoptNewMCACheck=QM,ADDRESS,NAME,ALL

ADOPTCHK=NONE/QMNAME/NETADDR/ALL (z/OS の場合)

パフォーマンス

- 考慮すべき問題
 - ▶ トランSMissionキューでの到着速度は?
 - ▶ 要求される応答時間は?
 - ▶ バッチ送言の所要時間は?
 - ▶ チャネルの稼働率は?
- 答は環境によって異なります
 - ▶ アプリケーションの使用状況
 - ▶ オペレーティングシステムハードウェア
 - ▶ ネットワークの帯域幅
 - ▶ メッセージサイズ

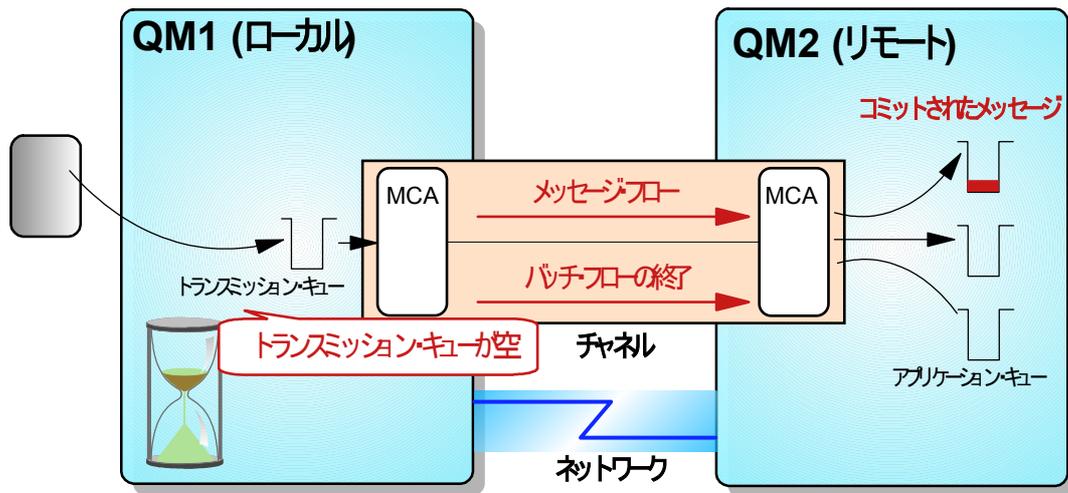


パフォーマンスを考慮した構成

- バッチ送言のため、バッチインターバルを使用する
 - ▶ 到着速度が完全なバッチの送言速度よりも遅い場合
 - ▶ 待ち時間が長くなる場合があるが、トータルのスループットは向上
- 常にチャネルを稼働状態にするため、切断インターバルを高設定
 - ▶ チャネルの稼働停止再稼働の回数を削減
 - ▶ ネットワークが「アクティブセッション」単位課金されない場合、またMaxChannelsのリミットを越えない限り
- 3つのシナリオ
- トラストドバインディングを使用
- 「Capacity Planning Guides」を参照のこと
 - ▶ SupportPac MP00
 - ▶ SupportPac MP16



バッチインターナル



■ メッセージのバッチの最大継続時間

DEF CHL BATCHINT(100)

▶ 回復可能メッセージはBATCHINT の後で送信



バッチインターナル(説明)

トランザクションキーが空の次の回復可能メッセージのバッチ送信が終了します。

これは、必要以上に長い継続時間およびディスク・アクティビティーの原因となります。

たとえば、アプリケーションがトランザクションキューに継続メッセージを3つ書き送ります。

2番目のメッセージが書き送られる前に、チャネルが1番目のメッセージを検索することができます。

その結果、チャネルはバッチを完了し、2番目および3番目のメッセージは2回目のバッチによって処理されます。

バッチが若くして遅延 (例えばBATCHINT (100) をつかえる) だけで、3つのすべてのメッセージが半分のディスク・アクティビティーで、1回のバッチで送信できることがあります。

ただし、少なくとも100ミリ秒の遅延があるため、メッセージの送信に多少時間がかかると考えることにしてください。

BATCHINT は非継続メッセージNPMSPD (FAST) に影響を与えることはありません。これらのメッセージは、同時に書き送られるためです。

アプリケーションは、通常、スピードが重視されるデータベース接続は非継続メッセージとして、また、さまざまなスピードが重視されるデータベース接続は継続メッセージとして送信します。

BATCHINT の実際の効果は、MQ メッセージング・ネットワークおよびMQ メッセージング・プロファイルに大きく左右されます。

マシン上のチャネル数が多いほど、節約可能なコストは大きくなります。

BATCHINT に大きなリットがあるかどうかを判断するのに役立つ1つの方法は、効果的なバッチサイズを調べることです。

DISPLAY CHSTATUS コマンドを発行し、メッセージ数をバッチ数割ります。

この数値が非常に小さい場合 (1 から 2 の間)、チャネルは1つ2つのメッセージとコミットを行う必要ありませんが、これは非効率的であり、BATCHINT のリットが得られるでしょう。

これは1つの目安ですが、例えばすべてのメッセージが非継続の場合はBATCHINT は全役に立たないことにしてください。

もう一つ調べる方法は、チャネルプロセスのディスク使用率です。

ディスク使用率が高い場合、BATCHINT によりコストを削減し、全体のスループットを増やす可能性が高くなります。

N
O
T
E
S



シナリオ1

- プリロード済みのトランスマッションキューの大量送
 - ▶ 大きなバッチサイズを設定する (デフォルト値50 が適当)
 - あまり大きなメッセージを使用しない場合(再送リスト)
 - ▶ BatchInt は0



© 2003 IBM Corporation

シナリオ1 (説明)

最新のウォーマスチューニングのシナリオは、プリロード済みのトランスマッションキューでのメッセージの大量送です。

このシナリオでは、メッセージを大量のバッチとして送付することで、1メッセージ当たりのバッチ処理終了に要するコストを減らそうとしています。

N

そのため大きなバッチサイズを使用して (この場合 デフォルト値の50 が適当)、バッチインターバルを0に設定します。

O

このシナリオでは、すべてのメッセージはトランスマッションキューにプリロードされているため、バッチをオープンしておく必要はありません。

T

メッセージ当たりのバッチ処理終了に要するコストを少なくするため

E

1つのバッチで送付しようとするメッセージ数、ネットワーク接続途中で断れた場合バッチを全再送するため必要コストとのウォーマス上のバランスを取ることが必要です。

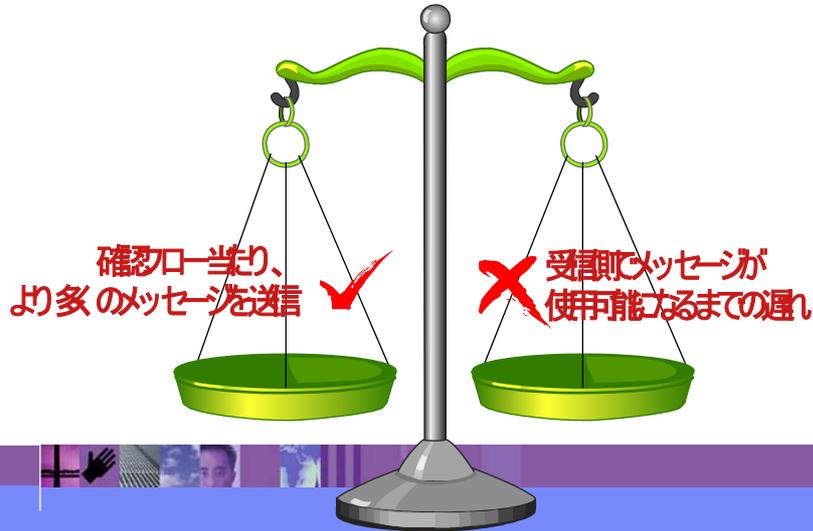
S

非常に大きなメッセージを送付する場合は、バッチサイズを50より小さくしてこのバランスをとる必要のあるでしょう。

© 2003 IBM Corporation

シナリオ2

- 遅延処理のための少量の転送
 - ▶ 正確な数値挿着率による
 - ▶ 大きなバッチの実装を図
 - ▶ 適切な送信の遅延時間を BatchInt で設定 例 1 分
 - ▶ あまり大きな値を設定しないこと: 作業単位の時間が長くなるため。



© 2003 IBM Corporation

シナリオ2 (説明)

NOTES

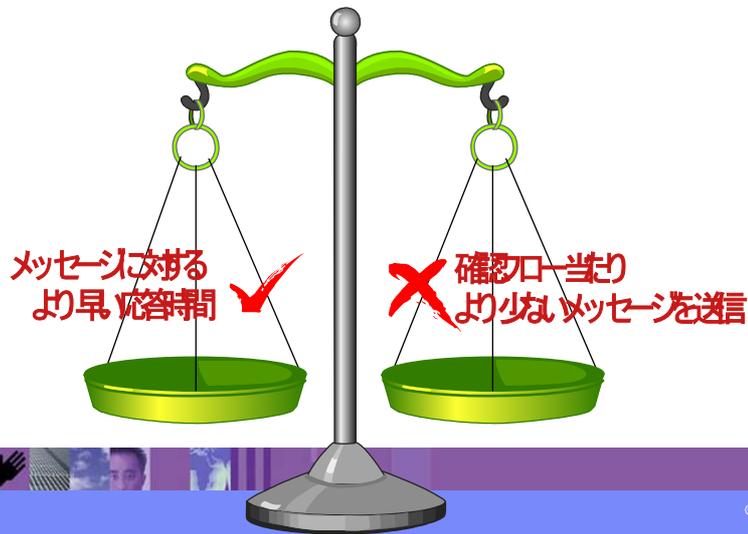
2番目のパフォーマンスチューニングのシナリオは、遅延処理(deferred processing) のためにメッセージを少量ずつ転送する方法です。このシナリオでは、メッセージのバッチ処理サイズを適切に大きくすることで、メッセージ当たりのバッチ処理終了に要するコストを減らそうとしています。これはバッチインターバルを使用します。メッセージがチャネルがメッセージを処理する速度よりも遅く到着してきます。多量な時間バッチをオープンしておくことで、より多くのメッセージを一回のバッチで処理することができます。1回のバッチで何個のメッセージを送信できるかということ、回復可能メッセージは送信できる時間が多くなる(バッチ完了するまでターゲットキューでミットされないため、少なくともバッチインターバル分は遅くなる)という事実の間パフォーマンス上のバランスを取ることが必要です。送信時間問題がないのであれば、バッチインターバルの値あまり大きく設定しないようご注意ください。大きすぎる値を設定すると、作業単位(Unit of Work) の時間が長くなってしまいます。

© 2003 IBM Corporation

シナリオ3

■ 同期応答

- ▶ 応答時間が重要な場合 BatchInt を使用しないほうが良い
- ▶ 連続リクエストが多い場合 バッチサイズを小さくする
- ▶ 非連続メッセージフローの場合 NPMSPEED(FAST) および BatchInt に 0 を使用



© 2003 IBM Corporation

シナリオ3 (説明)

3番目のパフォーマンスチューニングのシナリオは、応答時間が重要な同期要求応答モデルです。

これはシナリオ2と同様の状況ですが、1回のバッチ内のメッセージ数は、より早い応答時間に重点が置かれている点でシナリオ2と異なります。

N
O
T
E
S

連続メッセージのボリュームが多い場合は、より小さなバッチサイズを使用し、バッチインタバルを全使用するか、

またはメッセージを確実タイムリーに使用可能にするために数という非常に小さな値を使用するかを検討することになります。

非連続メッセージがある場合、非連続メッセージスピードまたは NPMSPEED に FAST を使用し、バッチインタバルに 0

での値を設定します。これは非連続メッセージが高帯域チャネルを介して配信され即時使用可能になります。バッチサイズ値を使用しますが、応答時間に影響はありません。

以下の数値を参考してください。

1秒当たりの非連続メッセージが30と予想される場合 BatchInt=2000 (2秒)、および適切なバッチサイズを50にするために BatchSize=50 を設定します。

© 2003 IBM Corporation

トラステッド・バインディング

- 単一プロセスとして稼働
 - ▶ MCA およびローカルキューマネージャーエージェント
 - ▶ MCA MQI コマンドに対する IPC オーバーヘッドがない
 - 非連続メッセージの場合影響大
- キューマネージャー ini ファイルで構成
 - ▶ channels スタンプ

MQIBINDTYPE=FASTPATH

- チャネル出口のテスト時使用可
- 以前のコマンド発行は要留意

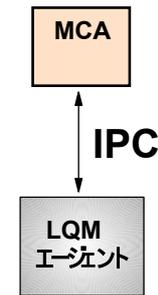
STOP CHL() MODE(FORCE)

(V5.3 以前)

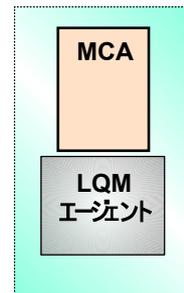
STOP CHL() MODE(TERMINATE)

V5.3

通常のチャネル



トラステッド・バインディング使用のチャネル



トラステッド・バインディング (説明)

プロダクトが連続システムで動作している環境は、デフォルトで、すべてのキューマネージャーの命令は、実際はエージェントと呼ばれるアプリケーションと別のプロセスによって実行されます。

N
O
T
E
S

各API コールは、IPC (プロセス間通信) メカニズムを介して、エージェント・プロセスに送信されます。これはアイソレーション (分離) を目指して行われます。

アプリケーションはキューマネージャー・リソースまたはメモリーを直接アクセスしません。しかし、各MQI コールはプロセスの切替であるため、パフォーマンスに影響があります。非連続メッセージの場合、コールにかかるコストの約60 %がプロセス・スイッチングに費やされます。

すべてのアプリケーションは「トラステッド」モードで実行させることができます。これは、キューマネージャー機能をアプリケーションに組み込んで直接実行することを意味します。通常のモードでアプリケーションを実行しないことを奨励します。しかし、チャネルは製品コードであるため、高速(fastpath)の「トラステッド」アプリケーションとして実行するのは適していません。

デフォルトでは、チャネルは通常のアプリケーションとして実行されますが、QM.INI ファイルのchannels スタンプでMQIBindType=FASTPATHと設定すると、すべてのチャネルがトラステッドモードで稼働します。

チャネル出口は留意が必要です。チャネル出口はMCAに組み込まれて稼働するため、キューマネージャーエージェントと同じメモリースペースに存在します。

チャネル出口をこのモードで稼働させる前に、徹底的にテストする必要があります。

V5.3 以前のリリースでSTOP MODE (FORCE) コマンドを使用する場合は注意が必要です。

また、V5.3 でSTOP MODE (TERMINATE) コマンドを使用する場合は注意が必要です。

どちらの場合も、チャネルが強制終了された場合に、クリーンアップが完全に実行されない可能性があるため、キューマネージャーが壊れたり停止したりする可能性があります。



パラメーターリファレンス

| チャネル | 説明 | |
|--|------------------------------|--|
| HBINT | ハートビートインターバル | |
| BATCHSZ | 1回のバッチのメッセージの最大数 | |
| BATCHINT | メッセージのバッチの最大総間 | |
| DISCINT | チャネル閉鎖までのインターバル | |
| SHORTRTY, SHORTTMR LONGRTY, LONGTMR | チャネル再試行回数およびタイマー | |
| MRRTY, MRTMR MREXIT, MRDATA | メッセージ再試行回数およびタイマー、出口(z/OSは除) | |

| QM.INI - channels スタンプ | XPARMS | 説明 |
|------------------------|----------|----------------------|
| AdoptNewMCA | ADOPTMCA | 切替チャネルタイプ |
| AdoptNewMCATimeout | | 切替体タイムアウト |
| AdoptNewMCACheck | ADOPTCHK | 切替パラメーターチェック |
| KeepAlive | TCPKEEP | キープライブの使用 |
| MQIBindType | | チャネルFASTPATH バインディング |

| 環境変数 | XPARMS | 説明 |
|-------------|---------|-------------|
| MQRVCBLKTO | RCVTIME | 受信待ちタイムアウト |
| MQRVCBLKMIN | RCVTMIN | 受信待ち最タイムアウト |

パラメーターリファレンス(説明)

この図は、パフォーマンスおよびリカバリ機能に影響を与えるチャネルの属性をリストアップしています。

実際のチャネル定義の一部である属性または INI ファイルなどの場所セットアップする属性があります。

チャネル定義のパラメーターの詳細については「相互通信のマニュアル」を参照してください。

分散プラットフォームでは QM.INI ファイルに含めるパラメーターは channels スタンプに存在し、Windows ではレジストリで実装されています。

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

または MQSeries のサービスを使用してください。

チャネルインジエーターのパラメーターは z/OS 上で、分散プラットフォームの QM.INI

ファイルにある同じパラメーターの一部提供されています。XPARMS は CSQ6CHIP マクロを使用してビルドされます。