サービス指向アーキテクチャでの Oracle Service Bus クラスタと IBM WebSphere MQ クラスタの統合

Oracle ホワイト・ペーパー
2009 年 1 月更新

ORACLE

サービス指向アーキテクチャでの Oracle Service Bus クラスタと IBM WebSphere MQ クラスタの統合

Oracle Corporation 発行「ntegrating an Oracle Service Bus Cluster with an IBM WebSphere MQ Cluster in a SOA」の翻訳版です。

サービス指向アーキテクチャでの Oracle Service Bus クラスタと IBM WebSphere MQ クラスタの統合

はじめに

サービス指向アーキテクチャは、おもにビジネス・データの非同期交換需要の高まりにともない、業界パラダイムとして広く受け入れられるようになりました。ただし、メッセージング・ミドルウェアは組織によって異なる場合があるため、Oracle 製品ファミリーと各ミドルウェア・ソリューションとの統合方法の提供が必要です。とくに、ユーザーが Oracle Service Bus/IBM WebSphere MQ 統合システムを使用して、メッセージングの相互運用性、ロードバランシング、高可用性、高パフォーマンス、フェイルオーバー、（要求メッセージに関する）Exactly-Once メッセージ・サービス品質に対応できるようにする必要があります。

このホワイト・ペーパーでは、IBM WebSphere MQ のトランザクション拡張クライアントを使用して、Oracle Service Bus（リリース 2.1 以降）と WebSphere MQ（リリース 5.3 以降）のクラスタ環境を設定し、構成する方法について説明します。とくに、これらの製品間の分散トランザクション要求/応答メッセージ通信を中心に説明します。ここに提供されている例を使用すれば、ユーザーは Oracle Service Bus クラスタ・ドメイン内のロードバランシングとフェイルオーバーに加え、WebSphere MQ クラスタ内のロードバランシングを実現できます。

このホワイト・ペーパーでは、WebSphere MQ のトランザクション拡張クライアントを使用した Oracle Service Bus と WebSphere MQ のクラスタ環境の設定および構成方法について説明します。とくに、これらの製品間の分散トランザクション要求/応答メッセージ通信を中心に説明します。

メッセージ・インフラストラクチャの選択

Oracle Service Bus の Java Message Service（JMS）は、JMS 1.1 仕様の Oracle WebLogic 実装です。これはエンタープライズ・クラスのメッセージ・システムで、JMS 1.1 仕様を完全サポートしているだけでなく、標準 JMS API に多数の拡張機能を提供します。Oracle Service Bus の JMS 機能は Oracle WebLogic Server プラットフォームと緊密に統合されているため、ユーザーが Oracle Service Bus 管理コンソールで監視可能な非常にセキュアな Java エンタープライズ・アプリケーションを構築できます。Oracle Service Bus の JMS は、拡張アーキテクチャ（XA）トランザクションを完全サポートしているのに加え、クラスタリングとサーバー移行機能により高可用性を実現します。また、サード・パーティのメッセージング・ベンダーとのシームレスな相互運用性も備わっています。

統合システムの実装決定時には、Oracle Service Bus/WebSphere MQ メッセージ集合を使用する場合と、Oracle Service Bus/Oracle WebLogic JMS を使用する場合のトレードオフを比較検討する必要があります。一般に、WebSphere MQ と Oracle Service Bus を使用するおもな理由としては、すでに配置済みで簡単に交換できないレガシー・メッセージ・システムに対応していることが挙げられます。

Oracle Corporation 発行「ntegrating an Oracle Service Bus Cluster with an IBM WebSphere MQ Cluster in a SOA」の翻訳版です。

一般に、WebSphere MQ と Oracle Service Bus を使用するおもな理由としては、すでに配置済みで簡単に交換できないレガシー・メッセージ・システムに対応していることが挙げられます。一方、Oracle Service Bus/Oracle WebLogic JMS を使用する理由としては、使用や構成、監視が簡単で、ほかの JMS ベンダー製品以上のパフォーマンスと機能を発揮することが挙げられます。

一方、Oracle Service Bus/Oracle WebLogic JMS を使用する理由としては、使用や構成、監視が簡単で、ほかの JMS ベンダー製品以上のパフォーマンスと機能を発揮することが挙げられます。Oracle Service Bus/Oracle WebLogic JMS を使用する場合、次のような利点があります。

- Oracle WebLogic Server の必須構成要素として実行されます。これに対し、外部 JMS プロバイダの場合は、別のツール・セットを使用して構成、起動、停止、監視する必要があります。
- Oracle WebLogic Server の組込みの Java Naming and Directory Interface（JNDI）やクラスタリング・サポートと統合されているため、ファイルの集合や Lightweight Directory Access Protocol（LDAP）サーバーのいずれにおいても、別の JNDI インフラストラクチャを構成して管理する必要がありません。
- JMS 仕様に厳密に準拠しているため、アプリケーションの移植可能な記述ができます。
- 優れたパフォーマンスとスケーラビリティを実現します。
- JMS に対してより効率的なトランザクション管理を提供します。外部 JMS プロバイダを使用する場合は、Oracle Service Bus と外部 JMS プロバイダ間の 2 フェーズ・コミットが Oracle Service Bus と外部 JMS プロバイダの両方に追加で発生します。

Oracle Service Bus と WebSphere MQ の統合

統合システムを使用する目的は、WebSphere MQ メッセージ・システムと Oracle Service Bus を、WebSphere MQ の JMS インタフェースを使用して接続することです。そうすれば、ユーザーは通常、WebSphere MQ のネイティブ・プロトコルをビジネス・プロセスに使用できるようになります。分散トランザクションの整合性、信頼性、スケーラビリティ、フェイルオーバーを統合システムで実現し、Oracle Service Bus と WebSphere MQ アーキテクチャのクラスタ環境を最適化する必要があります。

設計と構成

このホワイト・ペーパーの次項では、Oracle Service Bus/WebSphere MQのクラスタ間通信に直接関連する機能の概要について説明します。[1]

[1] Oracle WebLogic ServerクラスタとIBM WebSphere MQキュー・マネージャ・クラスタの詳細については、付録を参照してください。

Oracle Corporation 発行「ntegrating an Oracle Service Bus Cluster with an IBM WebSphere MQ Cluster in a SOA」の翻訳版です。

分散トランザクションの整合性

WebSphere MQ のリリース 5.3 以降では、外部同期点コーディネータとあわせて XA トランザクション管理がサポートされています。このサポートは、JMS クライアントと C クライアントの両方を対象としています。WebSphere MQ のトランザクション拡張クライアントでは、外部同期点コーディネータの制御により、アプリケーションがほかのローカル・リソース・マネージャと一緒の作業単位内に加えられます。2003 年 2 月に WebSphere MQ のトランザクション拡張クライアントがリリースされるまで、これはローカルの MQ サーバーを使用することによってのみ可能でした。価格設定上、WebSphere MQ のトランザクション拡張クライアントは MQ サーバーと同様の扱いとなります。一方、現行の非トランザクション・クライアントは引き続き無料で提供されます。

WebSphere MQのトランザクション拡張クライアントが各 Oracle Service Bus クラスタ・サーバーと一緒に配置されている場合は、Oracle Service Bus と WebSphere MQ のリモート・サーバー間で分散トランザクション・メッセージをやり取りできます。

WebSphere MQ のトランザクション拡張クライアントが各 Oracle Service Bus クラスタ・サーバーと一緒に配置されている場合は、Oracle Service Bus と WebSphere MQ のリモート・サーバー間で分散トランザクション・メッセージをやり取りできます。

用語	説明
リソース・マネージャ	アプリケーションからアクセスと更新が可能なリソースを所有し、管理するコンピュータ・サブシステム（IBM WebSphere MQ キュー・マネージャ - キューがリソースに該当; IBM DB2 データベース - 表がリソースに該当）。
作業単位	アプリケーションが 1 つまたは複数のリソース・マネージャのリソースを更新する場合はたいてい、全更新が 1 つのグループとして正常に完了することが不可欠です。そうでないと更新はいずれも完了しません。このようにして完了する更新を"作業単位"または"トランザクション"内での発生と呼びます。
同期点コーディネータ	同期点とは、作業単位内の全更新がコミットまたは取消しされた時点を指します。作業単位を管理するコンピュータ・サブシステムを同期点コーディネータと呼びます。
拡張アーキテクチャ（XA）	トランザクション・マネージャが作業単位を管理するには、リソース・マネージャ用に設計されたインタフェースが必要です。そのようなインタフェースとしては、X/Open Company Limited（現社名：Open Group）の発行する XA インタフェースがあります。

Oracle Service Bus による分散トランザクションのサポート

Java トランザクション API を Oracle WebLogic Server 経由で分散すると、Oracle Service Bus JMS クラスタと WebSphere MQ クラスタ間での透過的な要求/応答メッセージのやり取りが可能になります。

WebSphere MQ クラスタの設定

リモート・メッセージ・フォワーダとして機能する WebSphere MQ クラスタ・ノードが停止して再起動する場合、MQ のクラスタ設定を使用すれば、Oracle Service Bus JMS クラスタと WebSphere MQ クラスタ間での要求/応答メッセージの相互運用で連続性と透過性を維持できます。これは、共有キューのローカル参照を保持するキュー・マネージャが残存するためです。引き続き Oracle Service Bus で、要求キューにメッセージを配置して、応答キューからメッセージを取得できます。また、ビジネス・アプリケーションで要求キューからメッセージを取得して、応答キューにメッセージを配置できます。

ただしこれは、共有要求/応答キューのローカル参照を保持するキュー・マネージャが停止した場合にはあてはまりません。その場合、メッセージをキューに配置することはできますが、キューからメッセージを取得することはできません。

透過的な要求/応答メッセージの設定と構成

一般に、インストールと構成の詳細は、オペレーティング・システム、マシン数、Oracle Service Bus と WebSphere MQ の希望クラスタ・サイズによって異なります。本番環境では、各 Oracle Service Bus クラスタ・サーバーは通常別々のマシンにホストされます。さらに、各 WebSphere MQ クラスタ・ノードも別々のハードウェア・システムにホストされる場合があります。

次項では、Oracle Service Bus JMS クラスタと WebSphere MQ クラスタ間で要求/応答メッセージを透過的にやり取りするための設定と構成の例を紹介します。この例では最小限のハードウェアを使用します。2 つのシステムを配置して、各システムで Microsoft Windows オペレーティング・システムを実行します。

次項では、Oracle Service Bus JMS クラスタと WebSphere MQ クラスタ間で要求/応答メッセージを透過的にやり取りするための設定と構成の例を紹介します。この例では最小限のハードウェアを使用します。2 つのシステムを配置して、各システムで Microsoft Windows オペレーティング・システムを実行します。1 番目のマシンでは、Oracle Service Bus ドメインと WebSphere MQ トランザクション拡張クライアントをホストします。2 番目のマシンでは、WebSphere MQ サーバーのクラスタをホストします。

Oracle Service Bus ドメインには、1 つの管理サーバーと、2 つの管理対象サーバーから成るクラスタが含まれます。WebSphere MQ クラスタには、2 つのノードが含まれます。以下の例は、本番要件に合わせて外挿できる一般的な構成手順の概要を示しています。

WebSphere MQ サーバーと WebSphere MQ トランザクション拡張クライアントのインストール例

WebSphere MQ サーバーと WebSphere MQ トランザクション拡張クライアントをインストールして構成するには、まず WebSphere MQ 5.3 または 6.0 をインストールします。インストール時には、ステップ・バイ・ステップのインストール手順に従い、必要に応じて付録に引用した IBM WebSphere MQ インフォメーション・センターを参照してください。

次に、WebSphere MQ にキュー・マネージャを 2 つ（または 3 つ以上）作成します。キュー・マネージャをクラスタ・メンバーとして設定すると、分散キューイングに比べて利点が得られます。クラスタリングを使用しない場合は、キュー・マネージャは独立して機能し、分散キューイングを使用して通信します。その場合、キュー・マネージャ間でメッセージを送信する必要が生じると、リモート・キュー・マネージャへの送信キューとチャネルの両方を定義する必要があります。

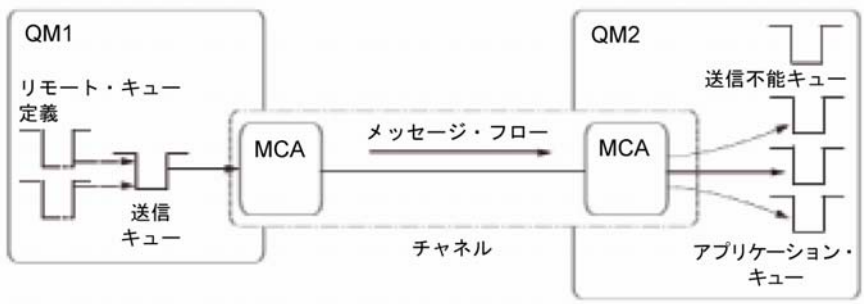

図1：分散キューイングに必要なコンポーネント

チャネル、リモート・キュー、各送信先の送信キューを明示的に定義することなく、同一クラスタ内の全キュー・マネージャ間でメッセージを送信できます

しかし、キュー・マネージャが 1 つのクラスタとしてグループ化されている場合は、クラスタ内の全キュー・マネージャ間でホストするキューを共有できます。チャネル、リモート・キュー、各送信先の送信キューを明示的に定義することなく、同一クラスタ内の全キュー・マネージャ間でメッセージを送信できます。クラスタ内の各キュー・マネージャにつき 1 つの送信キューがあり、それを使用してクラスタ内のどのキュー・マネージャへもメッセージを送信できます。クラスタ内の各キュー・マネージャでは、メッセージを受信するクラスタ受信チャネルを 1 つと、キュー・マネージャを登録し、クラスタの情報を交換するクラスタ送信チャネルを 1 つ定義するのみで済みます。図 2 は、キュー・マネージャの小さいクラスタを示しています。

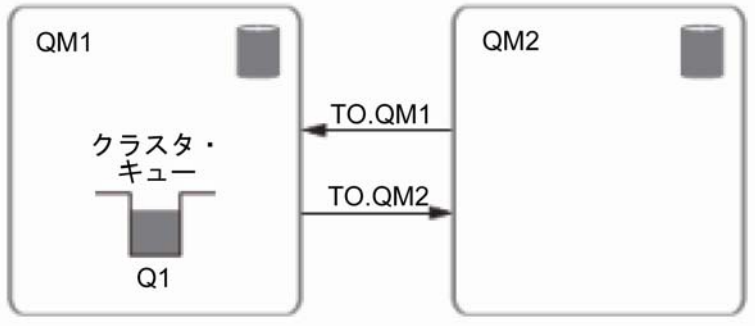

図2：2 つのキュー・マネージャで構成される小さいクラスタ

次に、クラスタを作成し、キュー・マネージャをクラスタ・ノードとして追加します。その後、両方のキュー・マネージャをホスト・リポジトリとして指定します。

完全リポジトリ・キュー・マネージャである QM1 と QM2 は、クラスタ内のキュー・マネージャ情報のリポジトリをホストします。（これらのリポジトリは、図 2 では網掛けされた円柱で表示されています）。QM1 は、クラスタ内のほかの全キュー・マネージャにアクセスできるキューをホストします。このキューは、クラスタ・キューと呼ばれています（クラスタ・キューは、図 2 では網掛けされたキューで表示されています）。

分散キューイングと同様に、アプリケーションでは MQPUT コールを使用して、任意のキュー・マネージャのクラスタ・キューにメッセージを配置します。また、MQGET コールを使用して、ローカル・キュー・マネージャのクラスタ・キューからメッセージを取得します。

クラスタ内にキュー・マネージャを取り込むとき、キュー・マネージャ間のシステム・チャネルの作成を伴います。各キュー・マネージャでは、メッセージを受信できる TO.qmgr というチャネルの受信側が定義されています。これをクラスタ受信チャネルといいます。クラスタ受信チャネルは、分散キューイングで使用される受信チャネルと似ていますが、このチャネルの場合はメッセージに加え、クラスタの情報を伝達します。

また、各キュー・マネージャでは、完全リポジトリ・キュー・マネージャのいずれかのクラスタ受信チャネルに接続するチャネルの送信側についても定義されています。これをクラスタ送信チャネルといいます。図 2 では、QM1 と QM2 に、TO.QM2 に接続するクラスタ送信チャネルがあります。QM2 には、TO.QM1 に接続するクラスタ送信チャネルがあります。クラスタ送信チャネルは、分散キューイングで使用される送信チャネルと似ていますが、このチャネルの場合はメッセージに加え、クラスタの情報を伝達します。

チャネルのクラスタ受信側とクラスタ送信側の両方を定義すると、チャネルが自動的に起動します。

チャネルのクラスタ受信側とクラスタ送信側の両方を定義すると、チャネルが自動的に起動します。

Windows システムでは、キュー・マネージャとクラスタの作成とクラスタ内へのキュー・マネージャの取込みは、WebSphere MQ Explorer インタフェースか、または…/WebSphere MQ/bin ディレクトリにあるコマンドライン・インタフェース（CLI）ツール（runmqsc）を使用して実行できます。UNIX システムでは、CLI ツール（runmqsc）を使用します。

Windows オペレーティング・システムでの例は次のとおりです。

```
>runmqsc.exe QM_IR02.1416_ALSB1
ALTER QMGR REPOS(ALSB1)

DEFINE CHANNEL(TO.QM_IR02.1416_ALSB1) CHLTYPE(CLUSRCVR)
TRPTYPE(TCP) CONNAME('ir02.bea.com(1416)') CLUSTER(ALSB1)

DEFINE CHANNEL(TO.QM_IR02.1417_ALSB1) CHLTYPE(CLUSSDR)
TRPTYPE(TCP) CONNAME('ir02.bea.com(1417)') CLUSTER(ALSB1)

>runmqsc.exe QM_IR02.1417_ALSB1
ALTER QMGR REPOS(ALSB1)

DEFINE CHANNEL(TO.QM_IR02.1417_ALSB1) CHLTYPE(CLUSRCVR)
TRPTYPE(TCP) CONNAME('ir02.bea.com(1417)') CLUSTER(ALSB1)

DEFINE CHANNEL(TO.QM_IR02.1417_ALSB1) CHLTYPE(CLUSSDR)
TRPTYPE(TCP) CONNAME('ir02.bea.com(1416)') CLUSTER(ALSB1)
```

次のステップでは、WebSphere MQ Explorer インタフェースまたは runmqsc を使用して、キュー・マネージャのいずれかに 2 つのローカル・キュー（要求/応答）を作成し、それらをクラスタで共有します。

以下に例を示します。

```
DEFINE QLOCAL(REQUESTQUEUE1) CLUSTER(ALSB1)

DEFINE QLOCAL(RESPONSEQUEUE1) CLUSTER(ALSB1)
```

WebSphere MQのトランザクション拡張クライアント 5.3 は、リリース 5.3 に別のインストール・パッケージとして付属しています。これを使用すれば、別々のマシンにホストされた Oracle WebLogic Server と WebSphere MQ 間のグローバル・トランザクションが可能になります。

次に、WebSphere MQ トランザクション拡張クライアント 5.3 を 2 番目のマシンにインストールします。WebSphere MQ トランザクション拡張クライアント 5.3 は、リリース 5.3 に別のインストール・パッケージとして付属しています。これを使用すれば、別々のマシンにホストされた Oracle WebLogic Server と WebSphere MQ 間のグローバル・トランザクションが可能になります。詳細については、本書の付録の項を参照してください。

次に、WebSphere MQ トランザクション拡張クライアント 5.3 をインストールしたマシンに、Oracle Service Bus クラスタ・ドメインをインストールします。

上記の設定は若干簡素化されていますが、本番環境では各管理対象サーバーを別々のマシンにインストールして、それぞれの WebSphere MQ トランザクション拡張クライアントと一緒に配置する必要があります。

Java メッセージ・サービスの構成 - 外部 Java メッセージ・サービス・プロバイダ

JNDIを使用したJMSの構成について説明するため、このホワイト・ペーパーでは『Using Foreign JMS Providers with WebLogic Server』（http://ftpna2.bea.com/pub/downloads/jmsproviders.pdf）のドキュメント情報を使用しています。

JMS クライアントでは、JNDI を使用して、初期コネクション・ファクトリ・オブジェクトと宛先オブジェクト（キューとトピック）を参照する必要があります。そのため、JMS クライアントでは、次に示す 2 つの主要設定の実装時に、WebSphere MQ の JMS プロバイダを使用できます。

- WebSphere MQ のコマンドライン・ユーティリティを使用した WebSphere MQ の JMS オブジェクトの JNDI バインド。WebSphere MQ の JMS プロバイダでは、コネクション・ファクトリと宛先のほかに、JNDI を使用して自身の構成情報（IP アドレス、サーバーのポート、転送タイプ、宛先属性、およびそのほかの特別オプション）を保存します。
- Oracle WebLogic Server の管理コンソールを使用した WebSphere MQ の外部 JMS プロバイダの構成。

この構成により、Oracle Service Bus サーバーに次の情報が提供されます。

- **初期コンテキスト・ファクトリ** - 参照の実行用に作成される JNDI クラス名です。JNDI クラスは Java 開発キットの一部として Sun Microsystems により提供されるか、または JMS ベンダーから提供されます。
- **プロバイダ URL** - この URL は、ディレクトリ情報の場所を示す初期コンテキスト・ファクトリです。LDAP などのプロトコルを使用するサーバーや、ファイル・システム内のファイルなどを参照できます。

- **コネクション・ファクトリの JNDI 名** - これは、JNDI に保存されているコネクション・ファクトリ・オブジェクト名です。各 JMS プロバイダが作成した独自のツールにより、このオブジェクト名が作成されて JNDI に保存されます。保存後は、すべての JMS クライアント・プログラム（Oracle Service Bus と Oracle WebLogic Server など）でこのオブジェクトを参照できます。
- **宛先 JNDI 名** - このオブジェクトは、JNDI に保存されている JMS の宛先、キュー、またはトピックを示します。コネクション・ファクトリと同様に、各 JMS プロバイダによって作成された独自のツールにより、このオブジェクトが作成されて JNDI に配置されます。

Java メッセージ・サービスの構成 - WebSphere MQ の Java メッセージ・サービス

WebSphere MQ の JMS を構成するには、WebSphere MQ サーバーと WebSphere MQ トランザクション拡張クライアントのインストールされた両マシンで、WebSphere MQ の JMS オブジェクトに同一の JNDI バインドを実行します。または、共有データ構造を使用することもできます。

まずは、次のファイルで非コメント化することにより、JNDI サービス・プロバイダとその初期コンテキストの URL を選択します。…/Java/bin/JMSAdmin.config

INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.RefFSContextFactory

PROVIDER_URL=file:/C:/JNDI-Directory

次に、JMS 管理ファイル（…/Java/bin/JMSAdmin.bat）を使用して、JMS オブジェクトを JNDI ツリーにバインドします。このバインドは C:¥JNDI-Directory¥.binding ファイルに保存されます。具体的には以下のようになります。

- ファクトリ名、キュー・マネージャ名、ホスト名、ポート、クライアント転送を示す WebSphere MQ の 2 つの XA コネクション・ファクトリ（各 WebSphere MQ サーバー・ノードに 1 つ）をバインドします。次に例を示します。

```
def xaqcf(XAQCF0) qmanager(QM_IR02.1416_ALSB1) HOSTNAME(ir02)
PORT(1416) TRANSPORT(CLIENT)

def xaqcf(XAQCF1) qmanager(QM_IR02.1417_ALSB1) HOSTNAME(ir02)
PORT(1417) TRANSPORT(CLIENT)
```

- キュー名と、そのキューへのローカル参照を保持しているキュー・マネージャー名、および永続性モード指定して、2 つの MQ キュー（要求と応答）をバインドします。もしキューからデキューするクライアントが WebSphere MQ のネイティブ・クライアントとなる場合には、TARGCLIENT（MQ）オプションを指定する必要があります。このオプションにより、クライアントでメッセージの適切な WebSphere MQ ヘッダーを取得できます。次に例を示します。

```
def Q(REQUESTQUEUE1) queue(REQUESTQUEUE1)
qmanager(QM_IR02.1416_ALSB1) PERSISTENCE(PERS) TARGCLIENT(MQ)

def Q(RESPONSEQUEUE1) queue(RESPONSEQUEUE1)
qmanager(QM_IR02.1416_ALSB1) PERSISTENCE(PERS)
```

最後に、WebSphere MQ の JMS バインドの結果を表示するため、dis ctx コマンドを実行します。

```
InitCtx> dis ctx

   Contents of InitCtx

        .bindings                              java.io.File
    a    RESPONSEQUEUE1                    com.ibm.mq.jms.MQQueue
    a    REQUESTQUEUE1                     com.ibm.mq.jms.MQQueue
    a    XAQCF0
 com.ibm.mq.jms.MQXAQueueConnectionFactory


    a    XAQCF1
 com.ibm.mq.jms.MQXAQueueConnectionFactory
    5 Object(s)
    0 Context(s)
    5 Binding(s), 4 Administered
```

Java メッセージ・サービスの構成 - Oracle Service Bus の Java メッセージ・サービス

もっとも簡単な設定は、1 台の Oracle データベースと 2 台の管理対象サーバーを一緒に配置して、Oracle Service Bus クラスタ・ドメインを本番モードで構成することです。

setDomainEnv.cmd または setDomainEnv.sh を編集します。PRE_CLASSPATH に拡張子を追加します。

次に例を示します。

```
set IBM_MQ_DIR=C:\Program Files\IBM\WebSphere MQ\Java\lib
set IBM_MQ_ETC_DIR=C:\Program
Files\IBM\Source\C4BUV6AL\Windows\MSI\Program Files\IBM\WebSphere
MQ\Java\lib
set
EXT_PRE_CLASSPATH=%IBM_MQ_DIR%\com.ibm.mqjms.jar;%IBM_MQ_DIR%\fscontext.jar;
%IBM_MQ_DIR%\com.ibm.mq.jar;%IBM_MQ_DIR%\com.ibm.mqbind.
jar;%IBM_MQ_DIR%\providerutil.jar;%IBM_MQ_ETC_DIR%\com.ibm.mqetclient.jar;
%EXT_PRE_CLASSPATH%
```

また、デモでの使用を目的として、メモリ消費量を削減するようデフォルトのメモリ設定を変更します。

次に例を示します。

```
set MEM_ARGS=-Xms256m -Xmx256m
```

管理サーバーとクラスタ（管理対象）サーバーを起動します。

関連 JMS オブジェクトをすべて別の JMS システム・モジュールに保存するには、Oracle WebLogic Server 管理コンソールを使用して、クラスタに配置する新規 JMS システム・モジュールを作成します。

この JMS システム・モジュールでは、以下を作成します。

- WebSphere MQ の 2 つのクラスタ・ノードに対応する WebSphere MQ の 2 つの XA コネクション・ファクトリ（XAQCF0 と XAQCF1）と WebSphere MQ の要求/応答キュー（REQUESTQUEUE1 と RESPONSEQUEUE1）と一緒に、外部 JMS サーバーをクラスタに配置します。
- XA コネクション・ファクトリのデフォルトのデリバリ・モードを PERSISTENT に設定します。
- プロキシ要求ゲートウエイ共通分散キュー（UDQ）をクラスタに配置します。
- プロキシ応答の UDQ をクラスタに配置します。

1 番目のプロキシ・サービスを配置するローカル・キューを作成する目的は、UDQ に配置する 2 番目のプロキシ・サービスにメッセージをルーティングすることです。これはまったくのオプションですが、これにより WebSphere MQ サーバーへの送信メッセージに関係する全クラスタ・サーバーを表示できます。

またオプションとして、Plain Old Java Object（POJO）の JMS クライアントで要求メッセージを送信してシステムをテストする場合には、管理対象サーバーのいずれか 1 台にエントリ・ポイントとして配置するローカル・キューを作成できます。1 番目のプロキシ・サービスを配置するローカル・キューを作成する目的は、UDQ に配置する 2 番目のプロキシ・サービスにメッセージをルーティングすることです。これはまったくのオプションですが、これにより WebSphere MQ サーバーへの送信メッセージに関係する全クラスタ・サーバーを表示できます。

次に、Oracle WebLogic Server の管理コンソールの JNDI パネルに、JNDI ツリーと外部 JMS サーバー・オブジェクト（WebSphere MQ のコネクション・ファクトリとキュー）すべて、新規作成した Oracle Service Bus の JMS コネクション・ファクトリとキューが表示されていることを確認します。

Oracle Service Bus コンソール（sbconsole）を使用して、次のアクションを実行します。

- ローカル・キューを JMS エンドポイントとして、1 番目の JMS プロキシ・サービスを構成します（MDB はこのローカル・キューに配置されます）。POJO JMS クライアントを使用して初期メッセージを送信している場合、これはオプションとなります。
- ゲートウェイ UDQ を JMS エンドポイントとして、2 番目の JMS プロキシ・サービスを構成します（MDB はこのゲートウェイ UDQ に配置されます）。この JMS プロキシ・サービスには、応答 UDQ への応答が必要です。
- 1 番目の JMS プロキシ・サービスから 2 番目の JMS プロキシ・サービスへのルーターを構成します。

- JMS ビジネス・サービスを WebSphere MQ 要求キューに配置し、WebSphere MQ 応答キューへの応答を要求するように構成します。このビジネス・サービスには 2 つのエンドポイント、つまり WebSphere MQ クラスタのロードバランシング用の WebSphere MQ キュー・マネージャ 2 つを設定して下さい。
- 2 番目の JMS プロキシ・サービスと JMS ビジネス・サービスとの間にルーターを使用してパイプラインを構成し、2 番目の JMS プロキシ・サービスから JMS ビジネス・サービスにメッセージをルーティングするようにします。
- メッセージ・ヘッダーをロギングする要求/応答ロギング・アクションを構成します。このロギングにより、要求/応答パイプラインを通過した要求/応答メッセージ数が表示されます。

図 3 は、Oracle Service Bus/WebSphere MQ の設定例を示しています。

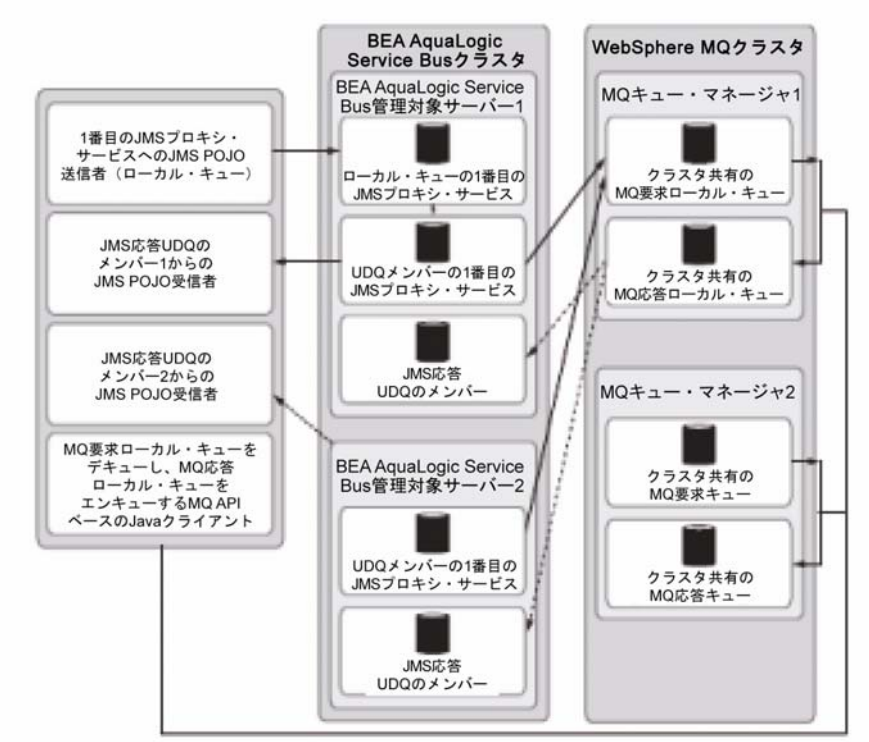

図3：テスト設定 - Oracle Service Bus クラスタ・ドメインと WebSphere MQ クラスタ間の要求/応答メッセージ通信

システムのテスト

次のステップでは、新たに統合されたシステムをテストします。Oracle Service Bus XA トランザクション・マネージャと WebSphere MQ XA リソース・マネージャで、Oracle Service Bus クラスタと WebSphere MQ クラスタ間のメッセージのやりとりを確認するには、以下の構成が必要です。

Oracle Service Bus XA トランザクション・マネージャと WebSphere MQ XA リソース・マネージャで、Oracle Service Bus クラスタと WebSphere MQ クラスタ間のメッセージを確認するには、いくつかの構成が必要です。

- WebSphere MQ と Oracle Service Bus 両方の XA コネクション・ファクトリ
- PERSISTENT デリバリ・モード
- Oracle Service Bus プロキシ・サービスに要求を送信する JMS クライアント
- 要求メッセージを受信して応答メッセージを送信する WebSphere MQ API ベースのビジネス・サービス
- 応答メッセージを受信する JMS クライアント

ビジネス・サービスは、WebSphere MQ の Java API ベースのクラスとして記述できます。このサービスは、要求キューから要求メッセージを取得し、応答キューに応答メッセージを送信します。WebSphere MQ クラスが WebSphere MQ サーバーとは別のホストで実行されている場合は、サーバー接続チャネルと対応するクライアント・チャネルを WebSphere MQ クライアントが接続する WebSphere MQ サーバーに作成します。

WebSphere MQ クラスは、環境変数の設定、WebSphere MQ リソースの初期化、要求キューの現在の深さの測定、ループ、要求キューからのメッセージの取得を処理します。また、要求相関 ID に対する各応答メッセージへの相関 ID の設定、応答キューへの応答メッセージの送信、WebSphere MQ リソースの解放をおこないます。

Oracle Service Bus の JMS 要求/応答実装で一般的なメッセージ要求/応答パターンがサポートされている場合は、WebSphere MQ クラスにより要求メッセージ ID に対する相関 ID が応答メッセージに設定されます。

POJO JMS クライアントの記述も可能です。POJO JMS クライアントは、1 番目のプロキシ・サービスに要求（XML ファイルなど）を送信します。2 つの POJO JMS クライアントを追加して、プロキシ・サービスの応答 UDQ の 2 つのメンバーをリスニングさせます。

設定は以下のようになります。

- クライアントが Oracle Service Bus 経由で WebSphere MQ に要求メッセージを送信します。標準 JMS API プロトコルまたは拡張 WebLogic JMS API が使用されます。Oracle Service Bus プロキシ・サービスがクライアントのメッセージを受信し、WebSphere MQ に配置されたビジネス・サービスに送信します。
- Oracle Service Bus では、基盤となる Oracle WebLogic JMS を使用します。これは JMS 1.1 仕様をサポートするエンタープライズ・クラスのメッセージ・システムであり、標準 JMS API の重要で役立つ拡張機能を提供します。
- ビジネス・サービスでは、ネイティブの WebSphere MQ を転送プロトコルとして使用します。
- キュー・マネージャの 1 つにローカル・キューを作成し、2 番目のクラスタ・メンバーと共有することにより、WebSphere MQ クラスタ・キューを作成しておきます。2 番目のクラスタ・メンバーは、共有キューのリモート・フォワーダとなります。WebSphere MQ は、基本的にはストア・アンド・フォワードのメッセージ受渡しミドルウェア・プロトコルです。

- メッセージをクラスタ・メンバーにラウンドロビン方式で送信することにより、Oracle Service Bus からの要求をロードバランシングできます。ただし、応答メッセージの取得（クラスタで共有されている応答キューからのデキュー）は、キューへのローカル参照をもつ WebSphere MQ サーバー・ノードにアクセスすることによってのみ可能です。つまり、要求メッセージの送信時には、Oracle Service Bus で JMS 宛先リストを繰り返し使用できます。ただし、応答メッセージを受信できるのは、1 つの JMS の宛先のみです。

テスト・シナリオ 1：管理対象サーバーが有効で接続されている場合

メッセージは、ゲートウェイ UDQ に配置された 2 番目の Oracle Service Bus の JMS プロキシ・サービスに直接送信できます。ただしこれは、送信者が分散キューのメンバー間にメッセージを分散できる場合に限ります。

まずは、Oracle Service Bus クラスタの管理対象サーバーがすべて有効で接続されており、WebSphere MQ クラスタ・サーバーもまたすべて有効で接続されているシナリオをテストします。

メッセージは、ゲートウェイ UDQ に配置された 2 番目の Oracle Service Bus の JMS プロキシ・サービスに直接送信できます。ただしこれは、送信者が分散キューのメンバー間にメッセージを分散できる場合に限ります。それ以外の場合は、POJO JMS クライアントを使用して事前定義された数の要求メッセージ（XML ファイルからの読取りなど）を 1 番目の JMS プロキシ・サービスに送信します。

- 1 番目の JMS プロキシ・サービスが、2 番目の JMS プロキシ・サービスに要求メッセージをルーティングします。
- 2 番目の JMS プロキシ・サービスがゲートウェイ UDQ に配置され、応答 UDQ への応答を要求します。
- 2 番目の JMS プロキシ・サービスが、JMS ビジネス・サービスに要求メッセージをルーティングします。
- JMS ビジネス・サービスは WebSphere MQ 要求キューに配置されており、WebSphere MQ 応答キューへの応答を要求します。
- JMS ビジネス・サービスには 2 つのエンドポイントがあり、それぞれ WebSphere MQ の 2 つのキュー・マネージャの 1 つに配置されています。この 2 つのエンドポイント間でロードバランシングを実行するため、ラウンドロビン・アルゴリズムが選択されます。
- WebSphere MQ API ベースの Java クライアントが、要求キューの現在の深さを測定してメッセージを検索し、要求キューから要求メッセージを取得して応答キューに応答メッセージを送信します。Oracle Service Bus 2.1 で必要とされているとおり、この Java クライアントは要求相関 ID を byte[ ]として読み取り、byte[ ]相関 ID として応答メッセージに設定します。
- 従来のエンタープライズ・アプリケーションのメッセージ要求/応答の統合パターンがサポートされているかどうかを確認したい場合は、Oracle Service Bus 2.5 のユーザー・ドキュメントを参照してください。サポートされている場合は、Oracle Service Bus 2.5 のビジネス・サービス・アプリケーションが要求メッセージ ID を byte[ ]として読み取ります。その後、応答メッセージに byte[ ]相関 ID として設定します。

- 応答 MDB が、WebSphere MQ の応答キューでリスニングし、応答メッセージを処理して応答パイプラインに配置し、応答 UDQ に送信します。
- 2 つの POJO JMS クライアントがそれぞれ応答 UDQ の 2 つのメンバーの一方をリスニングします。その後、メッセージを受信し、相関 ID を読み取って、現在の受信メッセージ数とあわせてログとして記録します。

テスト・シナリオ 2: Oracle サーバーが有効で、1 台の WebSphere MQ サーバーが再起動された場合

MQ API をベースとした Java クライアントは、要求キューの現在の深さを測定し、メッセージをループして要求キューから応答キューに移動します。

次に、Oracle Service Bus クラスタの管理対象サーバーがすべて有効であり、WebSphere MQ クラスタ・サーバーが停止して再起動されたシナリオをテストします。このシナリオを実装するには、以下の手順を実行します。

- Oracle Service Bus 管理サーバーと 2 台の管理対象サーバーを起動します。
- クラスタの WebSphere MQ キュー・マネージャの 1 つ（ローカルの要求/応答キューが構成されているほう）を停止します。
- POJO JMS クライアントを使用して、事前定義された数のメッセージ（XML ファイル）を 1 番目の JMS プロキシ・サービスに送信します。
- 停止された WebSphere MQ キュー・マネージャを起動します。
- WebSphere MQ に送信された全メッセージが、残存している 2 番目の WebSphere MQ クラスタ・ノード経由で WebSphere MQ の要求キューに正常に配信されたことを確認します。
- WebSphere MQ API をベースとした Java クライアントは、要求キューの現在の深さを測定し、メッセージを検索して要求キューから応答キューに移動します。この Java クライアントは、要求相関 ID をバイトとして読み取り、byte[]相関 ID として応答メッセージに設定します。
- 応答 MDB が、WebSphere MQ の応答キューでリスニングし、応答メッセージを処理して応答パイプラインに配置し、プロキシ・サービスの応答 UDQ に送信します。
- 2 つの POJO JMS クライアントがそれぞれプロキシ・サービスの応答 UDQ の 2 つのメンバーの一方をリスニングします。その後、メッセージを受信し、相関 ID を読み取って、現在の受信メッセージ数とあわせてログとして記録します。

テスト・シナリオ 3 : Oracle サーバーが有効で、WebSphere MQ サーバーがすべて再起動された場合

次に、Oracle Service Bus クラスタの管理対象サーバーがすべて有効であり、WebSphere MQ クラスタが停止して再起動されたシナリオをテストします。このシナリオを実装するには、以下の手順を実行します。

- Oracle Service Bus 管理サーバーと 2 台の管理対象サーバーを起動します。
- リモート・マシンの WebSphere MQ サービスを停止します。両方の管理対象サーバーが、切断関連の例外を繰り返し受信することが予測されます。

- POJO JMS クライアントを使用して、事前定義された数のメッセージ（XML ファイル）を 1 番目の JMS プロキシ・サービスに送信します。
- WebSphere MQ クラスタを起動します。
- WebSphere MQ に送信された全メッセージが、Oracle Service Bus により WebSphere MQ の要求キューに正常に配信されたことを確認します。

結論

このホワイト・ペーパーでは、Oracle Service Bus クラスタと WebSphere MQ キュー・マネージャ・クラスタ間のメッセージ通信の構成例について説明しました。この例は、こうしたシステムを構成する最適な方法を見つける最初の試みとしてご使用ください。実環境での WebSphere MQ クラスタのインストールについては、フィールド・エンジニアの説明に従ってください（Oracle で再現できるようにするためです）。その場合の構成がこのホワイト・ペーパーで説明した WebSphere MQ クラスタのタイプと大幅に異なっている場合は、追加情報が本書に記載されます。

Oracle Service Bus 2.1 以降では、ユーザーが WebSphere MQ クラスタで共有する要求/応答キューを構成し、各 WebSphere MQ サーバーの XA コネクション・ファクトリを分離して、メッセージを WebSphere MQ サーバーにラウンドロビン法で送信できます。この方法では、Oracle Service Bus によって WebSphere MQ のロードバランシングが実行されます。

ユーザーは、要求メッセージが別々のサーバーに配信されたことを確認できます。クラスタ要求キューにローカル参照をもつ WebSphere MQ サーバーにメッセージが到着すると、メッセージはそこにとどまります。メッセージがリモート・フォワーダに到着すると、リモート・フォワーダがメッセージをただちにクラスタ要求キューへのローカル参照をもつサーバーに転送します。ローカル・キュー・マネージャからクラスタ・キューを参照すると、クラスタに送信された全メッセージが表示されます。

Oracle Service Bus の JMS サービスが応答メッセージを取得できるのは 1 つの宛先（ローカル・キュー・マネージャのクラスタ応答キュー）からのみです。

要求/応答 XA トランザクションの Oracle Service Bus/WebSphere MQ クラスタ・シナリオでは、WebSphere MQ クラスタ・サーバーの 1 つが停止して再起動されても、メッセージが失われることはありません。

要求/応答 XA トランザクションによる Oracle Service Bus/WebSphere MQ クラスタ・シナリオでは、WebSphere MQ クラスタ・サーバーの 1 つが停止して再起動されても、メッセージが失われることはありません。実際、停止したサーバーがリモート・フォワーダである場合は、クラスタ要求キューのローカル・キュー・マネージャにメッセージが引き続き到着するため、動作はスムーズに継続されます。さらに、応答メッセージがクラスタ応答キューから取得されます。WebSphere MQ クラスタ全体が停止して再起動されても、メッセージが失われることはありません。

付録：参考資料および関連ドキュメント

- **『Messaging (JMS) for Oracle WebLogic Server 10.3』**
  e-docs.bea.com/wls/docs103/messaging.html
- **『Configure Foreign Servers』**
  e-docs.bea.com/wls/docs103/ConsoleHelp/taskhelp/jms_modules/foreign_servers/ConfigureForeignServers.html
- **『Accessing Foreign Server Providers』**
  e-docs.bea.com/wls/docs103/jms_admin/advance_config.html#1075917
- **『FAQ:Integrating Remote JMS Providers』**
  e-docs.bea.com/wls/docs81/faq/interop.html
- **『Using Foreign JMS Providers with Oracle WebLogic Server』**
  ftpna2.bea.com/pub/downloads/jmsproviders.pdf
- **『Using Oracle WebLogic Server Clusters』**
  e-docs.bea.com/wls/docs103/clust**er/index.html**
- **『Communications in a Cluster』**
  e-docs.bea.com/wls/docs9103/cluster/features.html
- **『IBM WebSphere MQ Information Center』**
  publib.boulder.ibm.com/infocenter/wmqv6/v6r0/index.jsp
- **『WebSphere MQ:Information Roadmap for All Users』**
  www-128.ibm.com/developerworks/websphere/zones/businessintegration/roadmaps/wmq/
- **『WebSphere MQ Queue Manager Clusters』**
  publib.boulder.ibm.com/infocenter/wmqv6/v6r0/index.jsp
- **『IBM WebSphere MQ Extended Transactional Client』**
  www-306.ibm.com/software/integration/wmq/transclient.html
- **『Meet the Experts:Mark Taylor on WebSphere MQ』**
  www-128.ibm.com/developerworks/websphere/library/techarticles/0302_taylor/taylor.html

サービス指向アーキテクチャでの Oracle Service Bus クラスタと IBM WebSphere MQ クラスタの統合
2009 年 1 月更新

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

海外からのお問い合わせ窓口：
電話：+1.650.506.7000
ファクシミリ：+1.650.506.7200
www.oracle.com