

Oracle Database 19c : Advanced Queuing

技術概要/2019年12月

免責事項

本文書には、ソフトウェアや印刷物など、いかなる形式のものも含め、オラクルの独占的な所有物である占有情報が含まれます。この機密文書へのアクセスと使用は、締結および遵守に同意した Oracle Software License and Service Agreement の諸条件に従うものとします。本文書と本文書に含まれる情報は、オラクルの事前の書面による同意なしに、公開、複製、再作成、またはオラクルの外部に配布することはできません。本文書は、ライセンス契約の一部ではありません。また、オラクル、オラクルの子会社または関連会社との契約に組み込むことはできません。

本書は情報提供のみを目的としており、記載した製品機能の実装およびアップグレードの計画を支援することのみを意図しています。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料になさらないでください。本書に記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。

製品アーキテクチャの性質により、コードが大幅に不安定化するリスクなしに、本書に記載されているすべての機能を安全に含めることができない場合があります。

目次

免責事項.....	2
概要.....	4
メッセージ・キューイング.....	4
Oracle Advanced Queuing.....	4
Oracle AQのコンポーネント.....	5
Oracle AQの機能.....	6
トランザクション・サポート.....	6
サービス品質.....	6
キュー・モデル.....	7
セキュリティ.....	7
メッセージの伝播.....	8
メッセージの変換.....	8
ルールベースのメッセージ・ルーティング.....	8
Oracle AQのデプロイメント.....	8
Oracle WebLogic Serverとの統合.....	9
ベスト・プラクティス.....	9
パフォーマンス監視.....	10
結論.....	11
Oracle Advanced Queuing – 簡単なチュートリアル.....	12

はじめに

ITマネージャーは、スケーラブルで強力な信頼性のあるプラットフォーム上でさまざまなシステムやテクノロジーを統合して、リアルタイムの情報の流れを実現できる標準ベースのエンタープライズ・メッセージ交換インフラストラクチャを必要としています。Oracle Database 19cでは、Oracle Advanced Queuing (Oracle AQ) によってそのようなエンタープライズ・メッセージ交換インフラストラクチャを実現しています。Oracle AQは、分散アプリケーションでの業務プロセス・ワークフローの自動化における主要コンポーネントです。Oracle AQを使用することにより業務では、別個のハイエンド・メッセージ指向ミドルウェア製品を必要とせずに、Oracle Databaseを利用してエンタープライズのメッセージングのニーズを満たすことができます。Oracleデータベース内のすべてのデータを管理できるだけでなく、信頼性と可用性が極めて高いスケーラブルな単一のOracle Databaseで、さまざまなシステムへのメッセージを使用してデータの流れやデータ交換も管理できます。Oracle AQにより、メッセージ・キューイング機能がデータベース内にネイティブに実装され、容易な管理性、高可用性、高いパフォーマンスとセキュリティを活用できるようになります。Oracle AQでは、Point-to-Pointキューとパブリッシュ/サブスクライブ・キュー、永続および非永続メッセージング、およびメッセージ順序の優先順位付けがサポートされるため、柔軟性と強力なメッセージング機能がアプリケーションにもたらされます。Oracle AQとの間のインターフェースには、PL/SQL、JMS 1.1、JDBC、ODP.NET、Python、Node.js、C/C++ (Oracle Cloud Infrastructureの場合) などがあります。

このホワイト・ペーパーでは、エンタープライズ・メッセージ交換インフラストラクチャの一般的な要件に焦点を当てているほか、Oracle Databaseで使用できるアドバンスド・キューイング・テクノロジーによって、分散環境の業務ワークフローをどのように自動化できるかを説明しています。また、Oracle AQの高度なメッセージング、ルーティング、および伝播機能の一部について紹介し、ビジネス企業でOracle Databaseのデータベース統合メッセージング機能を利用することにより、インフラストラクチャの投資利益率を最大化し、堅牢で高度にスケーラブルな分散アプリケーションを構築して優れたサービス品質でユーザーに提供する方法を示します。

Oracle Database Release 12cのOracle AQでは、AQシャード・キューが導入されました。これによりJMS機能には、高度に最適化されたスケーラブルなデータベース・キューイングが実装されます。特にAQシャード・キューなど、機能の詳細については、技術概要『スケーラブルで高パフォーマンスのアドバンスド・キューイングを実現するAQシャード・キュー』を参照してください。このホワイト・ペーパーでは、アドバンスド・キューのより一般的な側面について説明します。

メッセージ・キューイング

メッセージ・キューイング・インフラストラクチャは、さまざまな、そして多くの場合分散されたアプリケーションでの情報共有と情報統合を可能にします。プロデューサ・アプリケーションがメッセージをキューに送信 (エンキュー) し、コンシューマ・アプリケーションがそこからメッセージを受信 (デキュー) します。プロデューサとコンシューマは、キューを使用して非同期式にやり取りします。この“分離”がメッセージ・キューイングの中核です。このパターンはマイクロサービスの場合に非常に役立つことが明らかになっていますが、これはマイクロサービスが、バインドされたコンテキストと疎結合されたサービス指向アーキテクチャとして定義されるからです。Oracle AQにより、トランザクション・メッセージングでのマイクロサービスのビルドとデプロイが簡素化されます。

メッセージは通常、重要なビジネス・イベントを表すため、基盤となるメッセージ交換インフラストラクチャには、必然的に一定の特性が必要となります。メッセージの作成、消費、伝播は、最高水準の整合性をもって処理される必要があります。メッセージは、エンタープライズ・スタックのいかなるコンポーネント障害に対しても保護される必要があります。あらゆる場合においてリカバリ可能でなければなりません。また、メッセージの内容と属性は、標準的なインターフェースを使用して容易に検索できなければなりません。さらに、インフラストラクチャは、システムのパフォーマンス、可用性、信頼性を妥協することのないスケーラビリティを備えている必要があります。

Oracle Advanced Queuing

Oracle Advanced Queuingは、Oracle Database 19cのデータベース統合型メッセージ交換インフラストラクチャです。Oracle AQでは、Oracleデータベースの機能を利用して、メッセージが永続キューに保管されます。高可用性、スケーラビリティ、信頼性など、Oracleデータベースの持つ運用上の利点はすべて、Oracle AQのメッセージとキューに適用できます。バックアップとリカバリ、セキュリティ、管理性などの標準的なデータベース機能は、Oracle AQで使用できます。Data Guard、Oracle Real Application Clusters (Oracle RAC)、自動ストレージ管理 (ASM) などのOracleテクノロジーをOracle AQと組み合わせると、極めて高可用でスケーラブルなメッセージ・システムが実現します。お客様は、標準的な既製のサーバーとストレージを使用してOracle AQベースのメッセージ・システムを構築し、パフォーマンス、可用性、または信頼性を犠牲にすることなく線形に拡張できます。

Oracle AQのコンポーネント

Oracle AQの4つの主要コンポーネントは以下のとおりです。

1. メッセージ：メッセージは、メッセージの内容（ペイロード）で構成されます。ペイロードは、入力されたデータ（生データ）と、メッセージ属性（管理情報）を使用して指定できます。
2. メッセージ・キュー：メッセージはキューに保管されます。それらのメッセージ・キューは、さまざまなアプリケーションがメッセージ形式の“メール”を探ることができる“郵便箱”として動作します。つまり、あるアプリケーションがあるタスクのために特定のアプリケーションと通信したい場合、そのアプリケーションはメッセージ・キューにメッセージを残すことができ、受信側のアプリケーションはそのメッセージを見つけて処理できます。
3. メッセージ・インタフェース：Oracle AQは、エンキュー、デキュー、伝播処理に対応しています。これらは、一般的な標準に対応することによって既存のアプリケーションとシームレスに統合されます。Oracle AQのメッセージは、PL/SQL、C/C++、Java、Visual Basic (Oracle Objects for OLEを介して)、Python、Node.js、ODP.NETなどの一般的なアプリケーション・プログラミング・インタフェース (API) を使用して作成、問い合わせ、伝播、および消費できます。Oracle AQでは、Javaアプリケーションでメッセージ・キューイング機能を利用できるようにするJava Message Service 1.1 (JMS) APIがサポートされます。
4. メッセージ処理：Oracle AQでは、メッセージ・ペイロードまたはメッセージ属性のデータを基にしたルールベースのメッセージ・ルーティングがサポートされます。さらに、メッセージがターゲット・アプリケーションやサブスクリバに自動的に送信される前に、メッセージ変換をメッセージに適用してデータを再フォーマットできます。Oracle Database 19cでは、IBM MQおよびTIBCO/Rendezvousで、Oracleメッセージ・ゲートウェイを介して、Oracle AQのメッセージを変換することもできます。

Oracle AQのさまざまなコンポーネントによって、エンタープライズ・アプリケーション統合や分散アプリケーションに必要な機能が提供されます。以下の図1に示す典型的な統合環境では、メッセージはOracle Databaseサーバー、アプリケーション、およびユーザーの間で作成、伝播、消費されます。

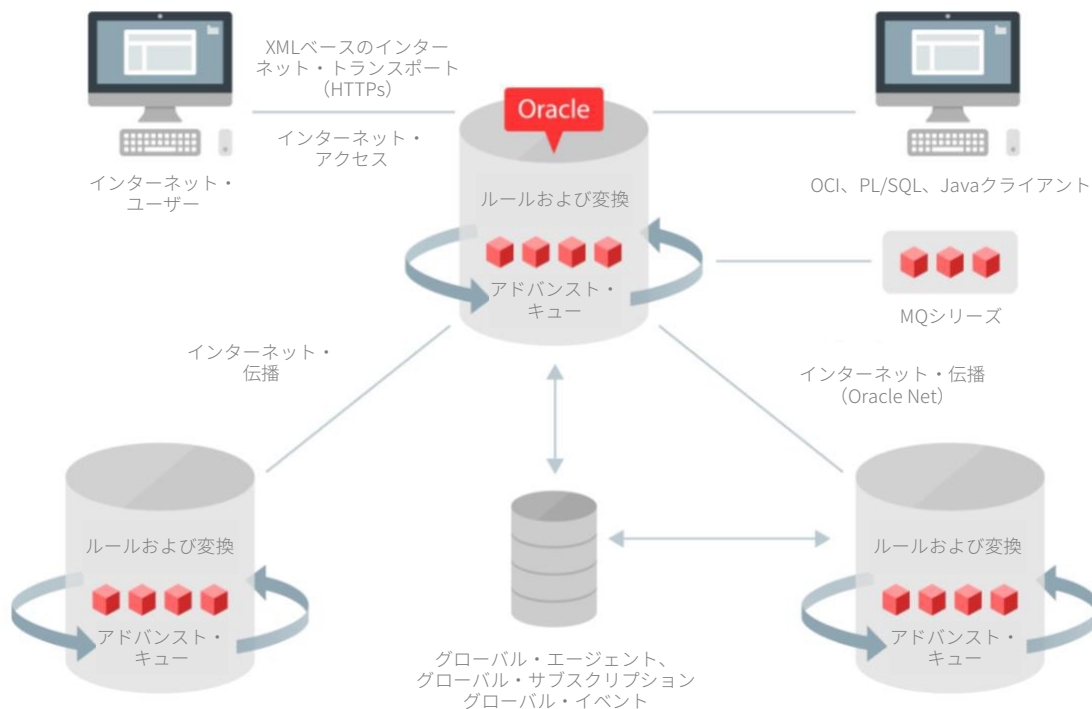


図1：Oracle AQを使用した統合アプリケーション環境

Oracle AQの機能

Oracle AQは、Oracleデータベース内の統合型メッセージ交換インフラストラクチャで、メッセージベースの分散アプリケーションを開発するための以下のような重要な機能が組み込まれています。

- トランザクション・サポート
- パブリッシュ/サブスクライブ・サポート
- サービス品質 (QoS)
- キュー・モデル
- セキュリティ
- メッセージの伝播
- メッセージの変換
- ルールベースのメッセージ・ルーティング
- メッセージの優先順位

トランザクション・サポート

Oracle AQでは、リレーショナル・データで使用されるものと同じ基盤のインフラストラクチャをOracle Databaseで使用して、トランザクション・セマンティクスをメッセージに提供します。エンキューやデキューなどのキュー処理は原子的であり、Oracleデータベースでは、永続キューのメッセージの一貫性が保証されます。メッセージは、同じストレージ・デバイスを使用して他のデータとともに保管され、特別な設定や管理は不要です。Oracle AQでは、メッセージはどのような期間でも（必要であれば無限に）保持でき、追跡や監査の目的で使用することも、データ・マイニングや分析用のメッセージ・ウェアハウスを構築するために使用することもできます。アプリケーションは、リレーショナル・データとメッセージを単一のトランザクションで処理できます。外部の、またはサード・パーティのメッセージ・システムが使用される場合、アプリケーションでは通常、2フェーズのコミット・アルゴリズムを使用してトランザクション・セマンティクスを実現する必要がありますが、この方法は高額になる可能性があります。一方、Oracle WebLogicでは、JTAトランザクションが同じOracleデータベースに対するAQ JMS操作とJDBC操作のみで構成されている場合、1フェーズ・コミットの最適化を使用します。

サービス品質

メッセージは、さまざまな理由から永続的でなければなりません。規制の順守、業務プロセスの監査、分析などのために、メッセージは異なる期間保持される必要があります。アプリケーションは、Oracle AQのメッセージ・キューイング・インフラストラクチャに依存することで、一度限りの送信メカニズムを保証できます。ポートフォリオ管理システムと取引処理システムを統合する金融サービス・ハブは、法的要件を満たすために、クライアントの注文メッセージを保持する必要があります。パートナーまたはサード・パーティのフルフィルメント請負業者と統合して注文を処理するには、注文プロセスが完了するまで、メッセージを交換し、保持することが求められます。

アプリケーションは、メッセージ形式でデータを交換し、業務プロセス・ワークフローを処理するためにそれらのメッセージを保持する必要があります。

Oracle AQはOracleデータベースと統合されているため、ほとんどのメッセージ・システムよりも高品質のサービスを提供できます。Oracle AQのキューは、他のOracle表と同じようにバックアップできます。Oracle Data Guardにより、Oracle AQのメッセージに高可用性がもたらされます。Oracle AQは、非常に大規模な同時メッセージ・ボリュームにまで拡張できるスケーラビリティを持ちます。小規模な同時メッセージ送信は、大規模なメッセージ送信によって阻止されません。Oracle AQでは、プロデューサを制限しなくとも、極めて大規模なメッセージ・バックログを処理できます。非常に大規模な（数ギガバイトなどの）メッセージ・ペイロードを処理し、LOBストリーム・インフラストラクチャを活用します。Oracle AQのメッセージはSQLを介して表示され、Oracle AQの表は索引付けできます。

Oracle AQでは、データベースに支えられたサービス品質の永続メッセージに加えて、非永続メッセージやバッファされたメッセージに対する統合サポートが提供されます。メッセージは、場合によっては、メッセージのエンキューからデキューまでの時間によって評価される待機時間が極めて短くなければなりません。そのようなメッセージは一時的である可能性があります。つまり、保持される必要はありません。株価情報のサブスクライブには、一定間隔の更新が必要ですが、現在のほとんどの株価更新は大半のコンシューマにとって十分であるため、更新が失敗に終わっても、再試行する必要はありません。たとえば、位置を基に発行されるモバイル・クーポンは、できる限りすぐにモバイル・サブスクライブに送信される必要があります。

エラーが発生した場合は、アプリケーションは、それらのサブスクライバにクーポンを再送信するか、サブスクライバの新しい位置を基に新しいクーポンを送信するかを決定できます。そのような種類のアプリケーションでは、Oracle AQによって、待機時間が短く高パフォーマンスなメッセージ管理インフラストラクチャに、インメモリの、またはバッファされたメッセージングが提供されます。

Oracle Database 19cのOracle AQでは、永続メッセージと非永続メッセージの双方に包括的な機能が用意されています。

1. 永続メッセージング

Oracle AQでは、ネットワーク、ハードウェア、ソフトウェアの障害時でも一度限りの処理が保証されなければならない永続メッセージに、キューと基盤のキュー表が提供されます。アプリケーションは、Oracle AQのキューを使用して、外部プログラムやアプリケーション内のモジュールから同時に到着するメッセージを処理できます。Oracle AQは、さまざまなメカニズムをサポートし、メッセージが処理される順序を管理します。アプリケーションでは、エンキュー時に各メッセージの優先順位を指定でき、この優先順位を使用してメッセージが消費される順序を管理します。代わりに、エンキュー時間やコミット時間に依拠してメッセージをソートし、メッセージを消費する場合のFIFO（先入れ先出し）順位を取得することもできます。コミット時間とは、トランザクションがコミットされた時間であり、トランザクションが相互に依存している場合は特に有用です。永続的なサービス品質は、Oracle AQのデフォルトです。

2. 非永続メッセージング

高いパフォーマンスが要求されるため、Oracle AQの永続メッセージングによって提供される信頼性やトランザクション・サポートのトレードオフを厭わないアプリケーションもあります。非永続メッセージのキューには、メモリ内のメッセージが保存され、ディスクI/Oは関係しません。バッファされたキューのメモリは、SGAから割り当てられ、'streams_pool_size'パラメータを使用して管理できます。代わりに、SGA自動チューニングを使用して適切なメモリを自動的に割り当てることもできます。永続メッセージに使用できるメッセージの順序付けスキームはすべて、バッファされたメッセージにも使用できます。

Oracle AQでは、汎用APIを介した永続メッセージと非永続メッセージがサポートされ、アプリケーション・ロジックとメッセージ統合ロジックを効果的に分離するメッセージ交換インフラストラクチャが提供されます。AQキューは、Point-to-Pointやパブリッシュ・サブスクライブなどの異なるキューイング・モデルで設定し、業務アプリケーションを相互に柔軟かつ高い信頼性で通信させることができます。

キュー・モデル

Oracle AQでは、Point-to-Pointとパブリッシュ・サブスクライブという2つのキュー・モデルがサポートされます。Point-to-Point、すなわち単一コンシューマ・キューは、特定のターゲットに狙いを定めます。プロデューサおよびコンシューマが、メッセージを交換する汎用キューを決定します。Point-to-Pointのメッセージは、一度だけデキューできます。一方、パブリッシュ・サブスクライブ、すなわちマルチ・コンシューマ・キューは、複数のターゲットに狙いを定めます。パブリッシュ/サブスクライブ・キューのメッセージは、複数のコンシューマによってデキュー可能です。この種類のキュー・メッセージングは、ブロードキャスト送信やマルチキャスト送信に使用できます。アプリケーションでは、コンシューマに送信するためのルールを設定でき、ルールは、メッセージのペイロード、属性、またはその両方で定義できます。サブスクライバ・アプリケーションは、デキュー時にサブスクリプション・ルールと一致するメッセージを自動的に受信できます。

パブリッシュは、異なるコンシューマやルールを意識しなくとも、引き続きメッセージを公開できます。Oracle AQでは、サブスクライバを追跡し、Oracle Call Interface (OCI) やPL/SQL通知メカニズムを使用して、サブスクライバ・アプリケーションに通知できます。そうすることで、メッセージ配信のプッシュ・モードが可能になります。

セキュリティ

Oracle AQでは、キューの管理タスクと運用タスクを分離する柔軟なセキュリティ・メカニズムがサポートされます。システムレベルのアクセス制御により、アプリケーション設計者やDBAは、すべてのキュー運用へのアクセスを制御し、特定のユーザーをキュー管理者として指名できます。キュー管理者は、データベースのあらゆるキューの管理タスクと運用タスクをどちらも実行できます。Oracle AQでは、エンキュー操作とデキュー操作に対するキューレベルのアクセス制御もサポートされます。特定のキューへのアクセスは、同じスキーマで実行されているアプリケーションのみに制限することができます。

メッセージの伝播

Oracle AQでは、同じデータベースの、またはリモート・データベースのあるキューから別のキューにメッセージを伝播できます。これによりアプリケーションは、同じデータベースや同じキューに接続していなくとも、分散環境で非同期的に相互通信できます。ソース・キューはマルチ・コンシューマ・キューであり、ターゲット・キューは、単一コンシューマ・キューまたはマルチ・コンシューマ・キューのいずれかです。ソース・キューにエンキューされたメッセージは自動的に伝播され、ターゲット・キューでデキューできます。伝播は、バックグラウンド・プロセスとして継続的に実行されるように設定することも、伝播されるメッセージがある場合のみ実行されるように設定することもできます。キューからキューへの伝播では、ソース・キューとターゲット・キューの組み合わせごとに、メッセージを伝播するための別個のジョブが作成されます。キューからdblinkへの伝播では、dblinkのすべてのターゲット・キューへの伝播で同じ伝播ジョブが使用されます。

メッセージの変換

ほとんどのB2B (Business-to-Business) アプリケーションでは、異種のアプリケーションやシステムを統合するために、異なる形式のデータを処理する必要があります。Oracle AQでは、メッセージをあるデータタイプから別のデータタイプに変換するための完全なデータ変換エンジンが提供されます。さまざまなOracleデータタイプとユーザー定義データタイプとの間のメッセージ変換がサポートされます。このような変換は、SQL式、PL/SQLファンクション、Javaで保存されたプロシージャなどで行われます。また、XSLTを使用したXML文書の変換もサポートされます。

変換によってメッセージ形式が変更されるため、あるアプリケーションによって作成されたメッセージを、別のアプリケーションが理解できるようになります。Oracle AQのメッセージ変換は、エンキュー時、デキュー時、またはキューへのサブスクライブ時にメッセージに対して自動的に適用できます。単一の変換は、メッセージの受信者数にかかわらず、メッセージをエンキューまたはデキューする際に指定する必要があります。リモート・サブスクリプションの場合は、特定のターゲット・キューに送信されるすべてのメッセージに単一の変換を指定する必要があります。メッセージの変換は、永続メッセージと非永続メッセージの双方に適用できます。変換は、スキーマ、または全データベースのエクスポートによってエクスポートされます。Oracle AQの表がエクスポートされる場合、キュー表に関連する変換もエクスポートされます。

ルールベースのメッセージ・ルーティング

Oracle AQでは、メッセージをマルチ・コンシューマ・キューの適切なサブスクライバにインテリジェントにルーティングすることや、各アプリケーションで指定されたルールに基づいて適切なキューに伝播することが可能です。ルールは、メッセージ・プロパティ、メッセージ・データ・コンテンツ、またはその両方で定義できます。メッセージのプロパティや内容を表す属性については、SQL条件式のWHERE句と似た構文でルールを表現できます。ルール・エンジンは、BITAND、CEIL、FLOOR、LENGTH、POWER、CONCAT、LOWER、UPPER、LENGTH、INSTR、SYS_CONTEXT、UIDなどの多くのSQL-92式を高速に評価できます。

Oracle AQのデプロイメント

Oracle AQは、企業のメッセージング機能を構築するためのインフラストラクチャであり、多くの業界で使用されています。Oracle AQは、シンプルでスケーラブルなワークフロー、メッセージング・ハブ、非同期処理、情報統合、アプリケーション統合、メッセージング・ウェアハウス、警告、メッセージベースの論理レプリケーションなどで利用されてきました。Oracle AQのデプロイメント例を以下で紹介します。

ある大手オンライン小売店は、Oracle AQの堅牢で信頼性のあるデータベース統合型メッセージ交換インフラストラクチャを使用して、サード・パーティ・プロバイダによってホストされていたCRMシステムを、バックエンド注文処理システムと統合しました。顧客データと注文データは、地理的に分散したサイトの2つのシステム間でほぼリアルタイムに同期されました。Oracle AQによるメッセージの永続性のおかげで、2つのシステムは永続キューを介してデータ変更を送信および受信できます。この非同期のメッセージの受け渡しにより、2つのシステムは分離され、注文処理サイトが停止している場合でも、顧客はオンライン・ストアを利用でき、注文が収集されます。Oracle AQにより、同社はOracle Databaseの信頼性とスケーラビリティを活用して、ホリデー・シーズン中の最大の通信量に対処しました。また、Oracle AQの標準ベースのインタフェースを使用して、わずか数週間で統合環境を構築しました。

欧州のある金融サービス機関は、同社のグローバルITインフラストラクチャを統合するための中核プラットフォームとして、Oracle AQを実装しました。Oracle AQによって提供されるエンタープライズ・メッセージング機能を使用して、ロンドン、ニューヨーク、シンガポール、香港、および東京のハブを接続しました。ハブの中核アプリケーションでは、XMLメッセージを使用して金融トランザクションと他の情報が交換されます。

情報の機密性から、顧客はどのようなソフトウェアまたはハードウェア・コンポーネントに障害や不具合が発生した場合でも、メッセージ損失のない、100%信頼のおけるメッセージングを求めています。メッセージは作成された順序で送信され、各ハブのディザスタ・リカバリ（DR）ロケーションでも使用できなければなりません。この顧客は、各ハブでマルチ・コンシューマ・キューと永続的メッセージングを使用しました。さらに、Oracle Data Guardを使用して、メッセージがDRロケーションに同期式にコピーされました。メッセージは、Oracle AQの伝播機能を使用してローカル・ハブからリモート・ハブに伝播され、適切なローカル固有のメッセージ変換が伝播先のハブに適用されました。Oracle AQとOracle Databaseによって、信頼性のある堅牢でスケーラブルなメッセージ交換インフラストラクチャが提供され、高いスループットのメッセージングを保証するという極めて厳しい顧客の要件を満たすことができました。

アドバンスド・キューイングは、非常に多く利用されているOracle Databaseの機能です。Oracle AQは、Oracle Databaseそのもので、およびその他のオラクル製品で、インフラストラクチャとして広く使用されています。Oracle AQは、データベース変更通知、データベース・アラート、イベントベースのジョブとルールベースのジョブ連鎖を含むデータベース・スケジューラ、高速アプリケーション通知（FAN）、Oracle Data Pumpで使用されています。Oracle AQを使用するオラクル製品にはほかに、Oracle Enterprise Manager、Oracle Audit Vault、Oracle Data Vault、Oracle WebLogic Application Server、Oracle E-Business Suite、Oracle Fusion Applications、Oracle Eloqua、Oracle Retailなどがあります。

Oracle WebLogic Serverとの統合

Oracle WebLogic Serverアプリケーションは、WebLogic Serverのリソース（Web Apps、Enterprise Java Beans、Message-Driven Beans）またはスタンドアロン・クライアントのいずれかを使用し、JMS APIを介してOracle AQと相互作用します。Oracle AQのJMSは、WebLogic JMS Foreign Serverフレームワークを使用します。データベース、JDBCドライバ、およびデータソースへの必要なリファレンスは、このフレームワークの一部として構成されます。WebLogic Serverのインストールには、必要なクラスがすべて含まれています。WebLogic Serverアプリケーションとスタンドアロン・クライアントは、標準のWebLogic JNDIコンテキストを使用して、Oracle AQのJMSコネクション・ファクトリとJMSデスティネーションを検索します。また、標準Java EE APIを使用して、Oracle AQのJMSをロードして起動します。

ベスト・プラクティス

Oracle AQは、ユーザー表を使用してキューを実装します。JMSシャード・キューでは、システムによってパーティション化された表が自動的に管理されます。Oracle AQの非シャード・キューの表では、索引構成表（IOT）とOracleデータベースの索引が使用されます。エンキューやデキューなどのOracle AQの操作によって、対応するデータベース・アクティビティが生成されます。基盤となるデータベースの処理パフォーマンスは、Oracle AQの全体的なパフォーマンスに著しい影響を与えます。このセクションでは、オラクルのベスト・プラクティス、推奨事項、Oracle AQメッセージ交換インフラストラクチャのパフォーマンスに最適なチューニングのヒントについて詳述します。

- ❖ JMSシャード・キューは、以下のキューを使用するOracle Advanced Queuingに適したJMS構成です。
 - 複数のOracle RACインスタンスにエンキュー元またはデキュー元があるJMSキュー
 - 高スループットのJMSキュー
 - 非シャード・キューを使用するときに過度に多くのシステム・リソースを消費するJMSキュー
 - サブスクライバ数が多いJMSキュー
- ❖ オラクルでは、とりわけ同時実行性の高いアプリケーションにおいて、AQキューの表に自動セグメント領域管理（ASSM）表領域を使用することを推奨しています。ASSM表領域を使用しない場合は、INITRANS、FREELISTS、FREELISTグループをチューニングして、Oracle AQのより良いパフォーマンスを達成する必要があります。ストレージ・パラメータは、*storage_clause*パラメータを使用して、キュー表の作成中に指定できます。キュー表が複数存在する場合は、高可用性とメンテナンスの観点から、キューごとに別個の表領域を定義することが推奨されています。
- ❖ 非シャード・キューでは、永続メッセージにおけるキュー操作のパフォーマンス特性は、基盤となるデータベース操作のパフォーマンス特性と同等です。エンキュー操作のコード・パスは、3つの索引構成表を持つ複数列表に対するSELECTとINSERTに相当します。デキュー操作のコード・パスは、複数列表でのSELECT操作とデキュー索引構成表でのDELETE操作に相当します。Oracle Real Application Clustersが使用されず、十分なストリーム・プール・メモリが存在する場合など、多くの使用例では、デキュー操作は最適化され、複数列表でのSELECT操作に相当します。最適化されたデキュー操作を活用するには、STREAMS_POOL_SIZEを増加させ、キューあたり少なくとも20 MBを割り当てます。

- ❖ キュー表索引とIOTは、Oracle AQのバックグラウンド・プロセスで自動的に結合されます。ただし、その後も必要に応じて継続的な監視と結合が必要となります。10.2以降は、自動セグメント領域管理 (ASSM) のオンライン縮小操作が同じ目的で使用される場合があります。索引のバランスが十分に取れていれば、キュー・モニターCPUの消費量が減少し、エンキューとデキューのパフォーマンスが確実に最適化されます。
- ❖ Oracle RACを使用すると、Oracle AQで高可用性とスケーラビリティを実現できます。JMSドライバを介してのみアクセスされるキューでは、JMSシャード・キューにより、Oracle RACのパフォーマンスが最適化され、管理性が向上します。AQ PL/SQL APIを介してアクセスされるキューでは、さまざまなキューをさまざまなRACインスタンスで管理できるようにすることで、Oracle AQのパフォーマンスを向上できます。異なるインスタンス・アフィニティまたはプリファレンスを、異なるキューのキュー操作をパラレル化できるキュー表に指定できます。オラクルでは、キュー表にインスタンス・アフィニティを設定することを推奨しています。インスタンス・アフィニティを設定すると、キュー・モニターのスケジューリングや伝播のバックグラウンド処理を分散できるようになります。インスタンス・アフィニティが設定されていない場合、キュー表のアフィニティが使用可能なインスタンスの間で任意に割り当てられるため、キュー表にアクセスしているアプリケーションと、高負荷のキューを監視しているキュー・モニター・プロセスの間でpingが発生する可能性があります。
- ❖ 統計を確実に収集して、メッセージの検索に最適な問合せ計画が選択されるようにします。デフォルトでは、キュー表は統計の自動収集からロックされています。統計は、代理キュー・メッセージ・ロードを使用して収集し、ロックすることが推奨されます。
- ❖ バックグラウンド・タスクを実行するための十分なキュー・モニター・プロセスが実行されるようにします。キュー・モニター・プロセスは、他の重要なバックグラウンド・アクティビティのためにも実行されていなければなりません。複数のキュー・モニター・プロセスで負荷が共有されます。十分なキュー・モニター・プロセスがあることを確認してください。これらは自動チューニングされますが、必要に応じてデータベース・パラメータaq_tm_processesを使用することによって、強制的に最小値に設定可能です。
- ❖ 待ち時間のあるデキューは、専用サーバー・プロセスでのみ使用される必要があります。共有サーバー環境では、共有サーバー・プロセスは、待ち時間を含む呼出し時間中のデキュー操作専用です。このようなプロセスが多く存在することは、深刻なパフォーマンスやスケーラビリティの問題が発生し、その結果、共有サーバー・プロセスのデッドロックが引き起こされる可能性があります。
- ❖ キューが際限なく増え続けているように見える場合は、アプリケーションによって孤立させられたすべての永続サブスクリバを探して削除します。
- ❖ Oracle Database Release 11では、ユーザーは気付いていませんが、非シャード・キューに対する数多くのパフォーマンス最適化が導入されています。たとえば、空の索引ブロックは、空の索引ブロックと、キューおよびサブスクリバのレベルでデキューされたメッセージの数を内部で監視することにより、徐々に解放されるようになりました。以前のリリースでは、このような定期的なメンテナンスは手作業で行う必要がありました。
- ❖ その他のパフォーマンスのベスト・プラクティスには、マルチ・コンシューマ・キューでの複数のデキュー操作を1回のトランザクションで一括処理することが含まれます。このとき、メッセージの優先順位を使用していない場合はナビゲーション・モードとしてFIRST_MESSAGEではなくNEXT_MESSAGEを使用し、BROWSEモードに続いてREMOVEでデキューする場合はREMOVE_NODATAデキュー・モードを使用します。パフォーマンス上のさらなるヒントについては、Oracle AQドキュメントを参照してください。

パフォーマンスの監視

Enterprise Managerでは、Oracle AQの管理機能に加えて、Oracle AQのメトリックの監視と警告のしきい値の設定がサポートされます。これらのメトリックは、永続キューと非永続キューの両方を対象としています。長い間デキューされていない見捨てられたサブスクリバの問題を検出するためや、低速なサブスクリバを特定するために使用できます。またメトリックは、スループット、領域の使用率、キューの長さなどの程度の速さで増加しているか、または減少しているかなど、キューの健全性も監視できます。詳細については、『Oracle® Enterprise Manager Oracle Database Plug-inメトリック・リファレンス・マニュアル』を参照してください。

永続メッセージングの統計、通知統計、およびサブスクリプション管理のビューが改善されたことで、19cでシステム・パフォーマンスとトラブルシューティングを直接監視できるようになりました。自動ワークロード・リポジトリ (AWR) では、永続メッセージング操作でもっともアクティブなキューが表示されるため、Oracle AQのパフォーマンス問題を容易に診断できます。ユーザーは、AWRの2つのスナップショットに基づきレポートを生成し、キューごと、またはサブスクライバごとにエンキュー率、デキュー率、その他の統計を算出できます。加えて、Oracle AQ向けのパフォーマンス監視PL/SQLパッケージは、Support Document 1163083.1より入手可能です。

結論

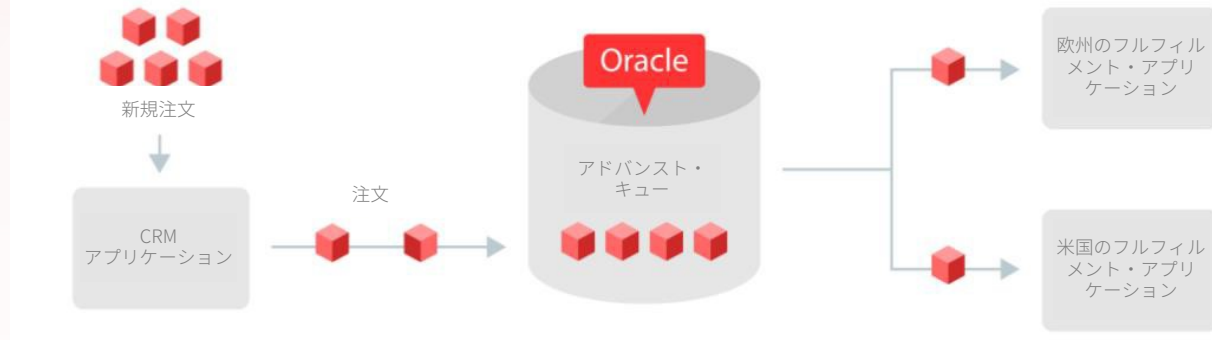
Oracle Databaseに組み込まれているOracle Advanced Queuingにより、データセンター内のさまざまなテクノロジーやアプリケーションを標準化し、統合するための堅牢なプラットフォームが提供されます。企業は、Oracle AQのエンタープライズ・メッセージ交換インフラストラクチャを活用して、極めてスケーラブルで信頼性のある分散アプリケーションを構築できます。差別化されたサービス品質、自動メッセージ変換、メッセージ伝播をはじめとするOracle AQの強力な機能により、企業は柔軟性のある強力なメッセージング・プラットフォームを設計するために必要な手段を手に入れます。データベースと統合されたAdvanced Queuingにより、成長し続けるスケーラブルで高可用性なビジネスで、重要な情報のリアルタイムで円滑な流れが実現し、管理作業が削減され、生産性が向上します。

Oracle Advanced Queuing – 簡単なチュートリアル

以下は、Oracleアドバンスド・キューイング（Oracle AQ）の構成方法と使用方法に関する短いチュートリアルです。このセクションでは、Oracle AQの基本的な機能と使いやすさについて説明します。

ある電子製品ストアは、注文のフルフィルメントのために、顧客の注文をオンライン・ストアから米国または欧州の適切なウェアハウスにルーティングする必要があります。

中央のCRMアプリケーションは、注文を顧客情報とともに収集し、



注文データをOracle Databaseに保存します。注文の詳細は、ユーザー定義タイプのメッセージとしてOracle AQにエンキューされます。次に、別のウェアハウスのフルフィルメント・アプリケーションが、注文メッセージをデキューし（キューから自動的に削除）、顧客の注文を処理します。CRMアプリケーション、欧州のフルフィルメント・アプリケーション、および米国のフルフィルメント・アプリケーションは、非同期式に連携しており、他のアプリケーションが停止している場合も機能する必要があります。たとえば、欧州のサイトが停止している場合でも、引き続きオンライン・ストアはCRMアプリケーションを介して新規の顧客注文を処理し、米国のウェアハウスは米国地域の注文を処理する必要があります。

このチュートリアルでは、Oracle AQのメッセージ交換インフラストラクチャを設定および使用するために必要な手順について説明します。

1. Oracle AQ管理者アカウントの構成

Oracle AQ管理者ユーザー（'aq_admin'）が、キューイング・インフラストラクチャを作成し、所有します。キューイング・インフラストラクチャの作成と管理が可能なロールAQ_ADMINISTRATOR_ROLEを、'aq_admin'ユーザーに付与する必要があります。

```
--管理者アカウントaq_adminを作成する
CREATE USER aq_admin IDENTIFIED BY aq_admin
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp;
ALTER USER aq_admin QUOTA UNLIMITED ON users;
--ロールをaq_adminに付与する
GRANT aq_administrator_role TO aq_admin;
GRANT connect TO aq_admin;
GRANT create type TO aq_admin;
```

2. 注文メッセージ・ペイロードと注文キューの設定

以下の手順は、aq_adminユーザーとして実行する必要があります。

- メッセージの内容すなわちペイロードを作成します。

```
CREATE TYPE orders_message_type AS OBJECT (  
order_id          NUMBER(15),  
Product_code     VARCHAR2(10),  
Customer_id      VARCHAR2(10),  
order_details    VARCHAR2(4000),  
price            NUMBER(4,2),  
region_code      VARCHAR2(100));
```

- キュー表とキューを作成します

ペイロードを作成した後に、キューイング・インフラストラクチャを作成できます。キューは、同じペイロード・タイプの複数のキューを保持できるキュー表を使用して実装されます。以下で、キュー表'orders_qt'とキュー'orders_msg_queue'を作成します。

```
DBMS_AQADM.CREATE_QUEUE_TABLE (  
queue_table => 'aq_admin.orders_qt',  
queue_payload_type =>  
'aq_admin.orders_message_type');  
DBMS_AQADM.CREATE_QUEUE (  
queue_name => 'orders_msg_queue',  
queue_table => 'aq_admin.orders_msg_qt',  
queue_type => DBMS_AQADM.NORMAL_QUEUE,  
max_retries => 0,  
retry_delay => 0,  
retention_time => 1209600,  
dependency_tracking => FALSE,  
comment => 'Test Object Type Queue',  
auto_commit => FALSE);
```

- キューを起動します

```
DBMS_AQADM.START_QUEUE('orders_msg_queue');
```

3. Oracle AQユーザー・アカウントの構成

Oracle AQユーザー ('aq_user') は、上記の手順で作成されたキューイング・インフラストラクチャにアクセスします。以下で、'aq_user'アカウントを作成し、必要な権限を付与します。

```
--aq_userユーザー・アカウントを作成する  
CREATE USER aq_user IDENTIFIED BY aq_user DEFAULT  
TABLESPACE users TEMPORARY TABLESPACE temp;  
--ロールをaq_userに付与する  
GRANT aq_user_role TO aq_user;
```

```
-aq_userにEXECUTE on message_typeを付与する  
GRANT EXECUTE ON message_type TO aq_user;
```

```
DBMS_AQADM.GRANT_QUEUE_PRIVILEGE(  
privilege => 'ALL',  
queue_name => 'aq_admin.orders_msg_queue',  
grantee => 'aq_user',  
grant_option => FALSE);
```

4. 注文キューに対するサブスクリプション

注文キューには2つのサブスクリプションがあります。一方は米国内で作成される注文用、もう一方は欧州で作成される注文用です。

2つの注文タイプは、orders_message_typeのregion_codeによって区別されます。

-- サブスクライバを追加するには管理者権限が必要

```
DBMS_AQADM.ADD_SUBSCRIBER(  
Queue_name => 'aq_admin.orders_msg_queue',  
Subscriber => 'US_ORDERS',  
Rule => 'tab.user_data.region_code = ^"USA"');
```

```
DBMS_AQADM.ADD_SUBSCRIBER(  
Queue_name => 'aq_admin.orders_msg_queue',  
Subscriber => 'EUROPE_ORDERS',  
Rule => 'tab.user_data.region_code =  
^"EUROPE"', Transformation =>  
'aq_admin.Dollar_to_Euro');
```

5. メッセージ変換の作成（任意）

メッセージ変換をOracle AQキューのメッセージに自動的に適用できます。以下のコードは、通貨をドルからユーロに変換する例です。注文メッセージのpriceフィールドはドルで指定されます。欧州のウェアハウスがメッセージをデキューすると、priceフィールドが自動的にユーロに変更されます。以下に簡単な例を示します。

```
CREATE FUNCTION  
Fn_Dollars_to_Euro(src aq_admin.orders_msg_type)  
Returns aq_admin.orders_msg_type AS  
Target aq_admin.orders_msg_type;  
  
BEGIN  
Target := aqadmin.orders_msg_type(src.order_id,  
src.product_code, src.customer_id,  
src.order_details, src.price*.5,  
src.region_code);
```



```
END;
```

```
DBMS_TRANSFORM.CREATE_TRANSFORMATION(  
schema          => 'AQ_ADMIN',  
name            => 'DOLLAR_TO_EURO',  
from_schema     => 'AQ_ADMIN',  
from_type       => 'ORDERS_MSG_TYPE',  
to_schema       => 'AQ_ADMIN',  
to_type         => 'ORDERS_MSG_TYPE',  
transformation => 'AQ_ADMIN.Fn_Dollars_to_Euro(source.user_data)');
```

6. キューの操作 - メッセージのエンキューとデキュー

以下の手順は、aq_userユーザーとして実行する必要があります。CRMアプリケーションは、注文メッセージを注文キューにエンキューし、その後フルフィルメント・アプリケーションが、その注文メッセージをデキューします。

- メッセージのエンキュー - DBMS_AQ.ENQUEUEプロシージャを使用して、新しい注文をorders_queueにエンキューします。注文の価格はドルで指定されます。

```
DECLARE  
enqueue_options dbms_aq.enqueue_options_t;  
message_properties  
dbms_aq.message_properties_t;  
message_handle RAW(16);  
message aq_admin.orders_message_type;  
message_id NUMBER;  
BEGIN  
message := AQ_ADMIN.MESSAGE_TYPE (1, 325, 49,  
'詳細：デジタルカメラ。ブランド：ABC。モデル：XYX', 23.2, 'EUROPE');  
-- エンキュー・オプションVISIBILITYのデフォルトは  
-- ON_COMMIT。メッセージは遅延と有効期限が  
-- ない  
message_properties.CORRELATION := message.order_id;  
  
DBMS_AQ.ENQUEUE (  
queue_name => 'aq_admin.orders_msg_queue',  
enqueue_options => enqueue_options,  
message_properties => message_properties,  
payload => message,  
msgid => message_handle);
```

```
COMMIT;
```

```
END;
```

- メッセージのデキュー - この例では、欧州のウェアハウスが欧州の注文に関連するメッセージをどのようにデキューするかを示しています。DBMS_AQ.DEQUEUEプロシージャを使用して、キューからメッセージを読み取るか、デキューします。価格は、デキューの前に自動的にユーロに変換されます。

```
DECLARE
```

```
dequeue_options dbms_aq.dequeue_options_t;
```

```
message_properties
```

```
dbms_aq.message_properties_t;
```

```
message_handle RAW(16);
```

```
message aq_admin.orders_message_type;
```

```
BEGIN
```

```
-- dequeue_optionsのデフォルト
```

```
-- Europe_Ordersサブスクライバのためにデキュー
```

```
-- Dollar_to_Euroの変換は
```

```
-- 自動的に適用される
```

```
dequeue_options.consumer_name :=
```

```
'EUROPE_ORDERS';
```

```
-- 即座の可視性を設定
```

```
dequeue_options.VISIBILITY :=
```

```
DBMS_AQ.IMMEDIATE;
```

```
DBMS_AQ.DEQUEUE (
```

```
queue_name => 'aq_admin.orders_msg_queue',
```

```
dequeue_options => dequeue_options,
```

```
message_properties => message_properties,
```

```
payload => message,
```

```
msgid => message_handle);
```

```
dbms_output.put_line('+ ----- +');
```

```
dbms_output.put_line('| MESSAGE PAYLOAD |');
```

```
dbms_output.put_line('+ ----- +');
```

```
dbms_output.put_line('- Order ID := ' ||
```

```
message.order_id);
```

```
dbms_output.put_line('- Customer ID:= ' ||
```

```
message.customer_id);  
dbms_output.put_line('- Product Code:= ' ||  
message.product_code);  
  
dbms_output.put_line('- Order Details := ' ||  
message.order_details);  
bms_output.put_line('- Price in Euros := ' ||  
message.price);  
COMMIT;  
END;
```

ORACLE CORPORATION

Worldwide Headquarters

500 Oracle Parkway, Redwood Shores, CA 94065 USA

海外からのお問い合わせ窓口


電話 + 1.650.506.7000+ 1.800.ORACLE1

FAX + 1.650.506.7200

oracle.com

CONNECT WITH US

+1.800.ORACLE1までご連絡いただくか、[oracle.com](https://www.oracle.com)をご覧ください。北米以外の地域では、[oracle.com/contact](https://www.oracle.com/contact)で最寄りの営業所をご確認いただけます。

 blogs.oracle.com/oracle

 facebook.com/oracle

 twitter.com/oracle

Integrated Cloud Applications & Platform Services

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXは、The Open Groupの登録商標です。0920

ホワイト・ペーパー **Oracle Database 19c : Advanced Queuing**

2019年12月

著者 : Sanjay Goil

共著者 : Mahesh Girkar, James Stamos, Amit Ketkar, Amitabha Samanta

ORACLE®