

Oracleホワイト・ペーパー

2013年7月

Oracle WebLogic Suite 12c (12.1.2) テクニカル・ホワイト・ペーパー

はじめに	1
ライセンス	1
Oracle WebLogic Suiteテクノロジーの概要	3
最新の開発プラットフォーム	4
Java EE標準のサポート	5
Webサービス	6
WebSocket	6
Oracle TopLinkとTopLink Data Service	7
統合開発環境 (IDE) のサポート	8
軽量のアプリケーション・サーバー	9
Oracle WebLogic ServerのMavenプラグイン	10
Oracle WebLogic ServerのAntタスク	11
開発環境とテスト環境の自動化	11
アプリケーションのクラス・ロード	11
ネイティブなクラウド管理	13
インストールとパッチ適用の統合	13
WebLogic管理フレームワーク	14
Managed Coherence Server	18
Oracle HTTP Serverの管理	20
Elastic JMS	20
Oracle Enterprise Manager Cloud Control	21
ミッション・クリティカルなクラウド・プラットフォーム	22
競合他社に対する優位性の実証	22
ワーク・マネージャとサーバー・パフォーマンス	23
Oracle WebLogic Serverのクラスタリング	24
サーバーとサービスの移行	25
ローリング・アップグレード	26
動的クラスタ	26
Oracle WebLogic ServerとOracle Coherenceの統合	28
JMSメッセージング	29

Active GridLink for RAC	30
Database 12cの統合	31
Exalogic Elastic Cloudソフトウェアの最適化	33
Oracle WebLogic Suite 12.1.2へのアップグレード	34
まとめ	35

はじめに

Oracle WebLogic Suite 12.1.2は、Oracle WebLogic Server製品ラインを代表するエディションの重要な新リリースです。Oracle WebLogic Suiteは、Webサーバー層、アプリケーション・サーバー層、データ・グリッド・テクノロジー層にまたがるオンプレミスのクラウド・インフラストラクチャを構築するための統合ソリューションです。従来型の環境とクラウド環境にまたがってアプリケーションを開発およびデプロイするのに最適なアプリケーション・サーバーであるOracle WebLogic Server、さらにOracle Coherence Enterprise Edition、Oracle Web Tier、Oracle TopLink、Oracle Java Mission Control、Oracle Java Flight Recorder、およびその他のコンポーネント・テクノロジーが含まれています。

Oracle WebLogic Suite製品は、重要なビジネス動向とビジネス要件に基づいて開発されています。企業ならびにCIOは、ITコストの削減、新製品や改良品の迅速な導入、および生産性の向上をますます重要視するようになってきています。長年にわたって、オラクルのお客様は、これらの要件を満たすためにOracle WebLogic Suiteを使用してきました。オラクルのお客様は、アプリケーションの迅速な開発、アプリケーション・デプロイメント用リソースのオンデマンド・プロビジョニング、管理の効率化、およびモバイル・クライアントとデスクトップ・クライアントの両方での高いアプリケーション・サービス・レベルを実現するため、アプリケーション・インフラストラクチャとしてOracle WebLogic Suiteを使用しています。実際に、これらのお客様は、それぞれ独自のプライベート・クラウド環境を構築しています。クラウド・コンピューティングは、パブリックにホストされるデプロイメントとプライベートの社内デプロイメントの両方のインフラストラクチャを構築するためのアプローチを形式化します。IT組織では、高い生産性、管理の効率化、および高いアプリケーション・サービス・レベルのすべてを低い所有コストで実現する、クラウド・コンピューティング・プラットフォーム・ソリューションの構成要素を提供するアプリケーション・インフラストラクチャ・テクノロジーに期待を寄せています。

Oracle WebLogic Suite 12.1.2は、これらのビジネス要件を満たすために、以前のリリースの機能に基づいて構築されています。既存の機能と新機能、および製品統合機能により、高い生産性を発揮する最新のアプリケーション開発プラットフォーム、低い所有コストを実現する、ネイティブなクラウド管理機能を備えた効率的な統合管理フレームワーク、および多様なクライアントでアプリケーションのパフォーマンス、可用性、およびスケーラビリティを最大化するミッション・クリティカルなクラウド・プラットフォームを提供します。これらの機能については、このドキュメントの以降の項で説明します。

ライセンス

Fusion Middleware 12.1.2では、オラクルは、単一の統合インフラストラクチャをOracle WebLogic Suiteとしてライセンスを付与し、提供しています。Oracle WebLogic Suiteには、次の主要なサーバー側ランタイム・コンポーネントが含まれています。

- Oracle Web Tier – フロント・エンドのOracle WebLogic ServerのデプロイメントにWebサーバー層を作成するための、Oracle HTTP Server、Oracle iPlanet、およびWebサーバーのプラグインが含まれています。

- Oracle WebLogic Server - 従来型の環境とクラウド環境にまたがったアプリケーションを開発およびデプロイするのに最適なアプリケーション・サーバーです。Oracle WebLogic Suiteには、開発ツールと管理ツール、クラスタリング、Active GridLink for RACなどの、Oracle WebLogic Server のすべての機能を使用する権限が含まれています。
- Oracle Coherence Enterprise Edition – 業界をリードする予測可能性と信頼性を備えた、スケールアウトのインメモリ・データ・グリッドです。
- Oracle Java SE Suite – Oracle JRockitで初めて導入された新機能とオラクルによる組込みJDKサポートを備えた、市場シェア・トップのOracle JDK (HotSpot) です。Oracle JDKは、Oracle WebLogic ServerとOracle Coherenceを実行するためのJava VMを提供します。Oracle WebLogic Suiteでは、他のJDK実装もサポートしています。

また、Oracle WebLogic Suiteには、次のコンポーネントも含まれています。

- Oracle TopLink – EclipseLink Java Persistence APIのリファレンス実装に基づいており、バージョン12.1.2でTopLink Data Serviceなどの新機能が追加されています。
- さまざまなIDE - Oracle WebLogic Serverアプリケーションの開発に使用する、業界をリードするEclipseベースのIDEであるOracle Enterprise Pack for Eclipse、Oracle Fusion Middlewareアプリケーションの開発に使用する、戦略的なIDEであるOracle JDeveloper、最新のJava EE APIをサポートする、オープンソースIDEであるOracle NetBeans。
- Oracle Application Development Framework – MVCベースのランタイム・フレームワークであり、Oracle JDeveloperまたはOracle Enterprise Pack for Eclipseを使用してOracle WebLogic Serverで実行するアプリケーションを構築する際に使用します。
- Oracle iAS Enterprise Edition – Oracle Containers for Java (OC4J)のライセンス権限と、Oracle Forms、Oracle Reportsなどの関連するコンポーネント・テクノロジーが含まれています。

Oracle WebLogic Suite 12.1.2のライセンスは、ライセンスを追加することで、Oracle Coherence Grid Edition Option と Oracle Enterprise Manager Management Pack (Oracle WebLogic Server Management Pack Enterprise EditionとCloud Management Pack for Oracle Fusion Middleware) に対応するように拡張できます。Oracle WebLogic Suiteのライセンス情報の詳細については、Oracle Fusion Middleware 12.1.2のライセンス情報ドキュメントを参照してください。

Oracle WebLogic Suite 12.1.2に含まれ、ライセンス付与されている製品コンポーネントには、物理的にパッケージ化されて、一般的な使用方法に合わせて個別のインストール・プログラムで提供されるものがあります。たとえば、主要なJavaアプリケーション・インフラストラクチャ・テクノロジーであるOracle WebLogic Server (Oracle TopLinkを含む) とOracle Coherenceは、同じインストール・プログラムにパッケージ化され、システムに作成されたJDKインストールにインストールされます。Web層テクノロジーは個別にパッケージ化され、通常は個々のシステムにインストールされます。Oracle WebLogic Suite製品コンポーネントのインストール手順の詳細については、Oracle Fusion Middleware 12.1.2の該当する製品ドキュメントを参照してください。

Oracle WebLogic Suiteテクノロジーの概要

次の図に、Oracle WebLogic Suite本番環境の一般的なデプロイメント・トポロジについて概要を示します。

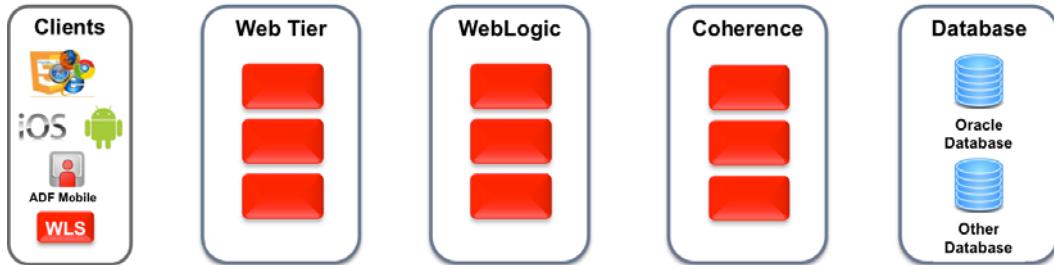


図1：Oracle WebLogic Suite本番環境の一般的なデプロイメント・トポロジ

このデプロイメント・トポロジには、ブラウザ・クライアント、ネイティブなモバイル・クライアント、Oracle WebLogic Server上にデプロイされたアプリケーションにアクセスするOracle WebLogic Server Javaクライアントなどのクライアントが含まれています。通常、これらのクライアントは、上記でWeb層として示されている、別々の一連のWebサーバー・システムにホストされているWebサーバー・プロキシを介して、Oracle WebLogic Serverの管理対象サーバーとクラスタにアクセスします。Webサーバーで実行されているWebサーバー・プラグインは、上記のトポロジのWebLogic層で実行されているOracle WebLogic Serverアプリケーションにクライアントのリクエストを転送します。

これらのアプリケーションは一般的に、データベースに保存されているデータにアクセスしてこれらのデータを更新します。通常はOracle Databaseですが、他のサード・パーティ・データベースもサポートされています。アプリケーション処理に使用されるアプリケーション・データは、WebLogic 層内のメモリにJavaオブジェクトの形式で保存されます。データ集中型のアプリケーションのパフォーマンスとスケーラビリティを向上させるためには、Oracle WebLogic Serverのインメモリ・アプリケーション・データを別のOracle Coherenceデータ・グリッド層に移動して、分散キャッシング、イベント、分析、およびコンピュータ処理の各サービスをOracle WebLogic Serverアプリケーションで利用できるようにすると効果があります。Oracle WebLogic ServerとOracle Coherenceの両方が、データベース層と効率的に統合するための独自の機能を備えており、特にOracle DatabaseとOracle Database Real Application Clustersで効果的です。

この概要では、Web層で提供されるセキュリティとロードバランシング機能、WebLogic層で提供される開発、管理、パフォーマンス、および信頼性の各機能、およびCoherence層で提供される組込みの信頼性とフェイルオーバーを利用した卓越した拡張性などの、インフラストラクチャの詳細な機能については省略しています。これらの機能については、このドキュメントの全体を通してさらに詳しく説明していきます。オラクルでは、Oracle WebLogic Suiteは、Webサーバー層、アプリケーション・サーバー層、およびデータ・グリッド層にまたがって業界をリードするアプリケーション・インフラストラクチャ実装を提供するものと確信しています。

Oracle WebLogic Suite 12.1.2では、これらの機能をさらに次のレベルに引き上げています。Oracle WebLogic Suite 12.1.2は、次のものを提供することで、Webサーバー層、アプリケーション・サーバー層、およびデータ・グリッド層にまたがるクラウド・アプリケーション・インフラストラクチャの実装を簡素化します。

- Maven統合の重要な更新、新しいWebSocketプロトコルのサポート、企業データにRESTfulインターフェースを提供してアプリケーションの開発を簡素化するTopLink Data Serviceなどの、アプリケーション開発の生産性を高める最新の開発プラットフォーム。
- Webサーバー層、アプリケーション・サーバー層、およびデータ・グリッド層にまたがって実績のある一貫したツールを使用し、効率的なクラウド・インフラストラクチャ管理を実現する、ネイティブなクラウド管理機能。Oracle Universal Installation (OUI)テクノロジーとOPatchテクノロジーの標準化により、クラウド・プロビジョニング用のインストールとパッチ適用の統合フレームワークを実現します。WebLogic管理フレームワークの機能拡張により、実績のあるWebLogic管理機能を一貫して使用し、Webサーバー層とデータ・グリッド層を管理できます。また、オラクルでは、Elastic JMSなどの拡張機能により、WebLogic管理フレームワークで継続して機能拡張を行っています。
- Webサーバー層、アプリケーション・サーバー層、およびデータ・グリッド層にまたがって、クラウド・アプリケーションの信頼性、可用性、スケーラビリティ、およびパフォーマンスにおけるもっとも厳しい要件に対応する、ミッション・クリティカルなクラウド・プラットフォーム。動的クラスタ、Oracle Database 12c統合、およびExalogicの最適化が、アプリケーションに提供されるサービス・レベルを強化するために追加された新機能の例として挙げられます。

このドキュメントの以降の項では、これらの各領域におけるOracle WebLogic Suite 12.1.2の機能について、Oracle WebLogic Suite 12.1.2で追加された新機能に重点を置いて説明します。このドキュメントでは、まず、Oracle WebLogic Suiteの最新の開発プラットフォーム機能について、Oracle WebLogic Serverアプリケーションの開発に焦点をあてて説明します。このドキュメントの次の項では、Oracle WebLogic Suiteのネイティブなクラウド管理機能について、Oracle WebLogic Serverの管理機能に焦点をあてて説明しますが、これらの機能を適用して一般的なOracle WebLogic Suiteデプロイメント・アーキテクチャのWeb層とCoherence層を管理する方法についても説明します。次の項では、Oracle WebLogic Suiteのミッション・クリティカルなクラウド・プラットフォームで提供されるサービス・レベルについて説明し、これらのサービス・レベルを提供する方法について説明します。

最新の開発プラットフォーム

オラクルでは、クラウド・アプリケーションなどのアプリケーションの開発を簡素化するとともにアプリケーション開発者の生産性を高める最新の開発プラットフォームを提供するために、Oracle WebLogic Suiteを大幅に改良しました。おもな開発機能は、以下のとおりです。

Java EE標準のサポート

生産性を最大限に高めるとともに、ミドルウェア・インフラストラクチャの使用に伴うリスクを軽減するため、標準を完全に採用していることが重要です。また、標準を採用しているため、開発者の能力を活用したり、相互運用性をサポートしたりでき、ベンダー独自のテクノロジーに縛られることはありません。Oracle WebLogic Server 12.1.1以降、Oracle WebLogic ServerはJava Enterprise Edition Version 6 (Java EE 6) に基づいて構築されており、Java EE 6をサポートしています。

Java EE 6では、開発者の生産性を向上する機能を提供することにもっとも重点を置いています。Java EE 6では、これを実現するために、アノテーションの使用範囲を広げ、コンポーネントの再利用と統合を簡素化し、特定レベルの機能を実装するのに必要なコードの量を低減し、Java EEアプリケーションをデプロイするのに必要なXMLの量を低減しています。Java EE 6で提供されている機能と簡素化によって、開発者は、Java EEインフラストラクチャの使用に注力するのではなく、ビジネス機能を実装するというアプリケーション開発作業に注力できます。Java EEの拡張機能には、次のようなものがあります。

- Servlet 3.0では、アノテーション、クライアント・リクエストの非同期処理、およびJava EE 6でのサード・パーティ・フレームワークの使用を簡素化するWebフラグメントのサポートが追加されています。
- JSF 2.0では、アノテーションのサポートが追加されており、JSFテンプレートなどの新しいページ作成の拡張機能が用意されています。テンプレートを使用すると、一般的なテンプレートからアプリケーション内に同様のページを複数作成できるため、同じルック・アンド・フィールでアプリケーションを容易かつ迅速に開発できます。
- EJB 3.1のインターフェースのないビューを使用すると、開発者はBeanクラスを使ってエンタープライズBeanを指定できます。別のビジネス・インターフェースを作成する必要はありません。また、EJB 3.1では、EJBをWARファイルに直接パッケージ化できるため、EJBを使ったアプリケーションのパッケージ化と開発を簡素化できます。
- Contexts and Dependency Injection (CDI) は、Java EE 6の新しいAPIです。CDIを使用すると、JSFアプリケーション内のJSFマネージド・ビーンをEJBで置換えできるため、Java EE 6アプリケーションのWeb層とエンタープライズ層のギャップがなくなります。これにより、Java EEアプリケーションのトランザクション・リソースにアクセスするWebアプリケーションの開発を簡素化できます。
- JPA 2.0では、オブジェクト/リレーションナル・マッピングとJava Persistence Query Languageで拡張機能が追加されており、オブジェクト・ベースの問合せを作成するための新しいクライティアAPIが用意されています。EclipseLinkは、Oracle WebLogic Serverに含まれているOracle TopLinkの基盤で、JPA 2.0のリファレンス実装です。
- Bean ValidationもJava EE 6の新しいAPIで、Java EEアプリケーションの複数のコンポーネントで検証ロジックを再利用できるようにし、複数のアプリケーション・コンポーネントで検証ロジックを複製する際の複雑な操作を排除します。
- Java API for RESTful Web Services (JAX-RS) は、Java EEアプリケーションでの軽量のRESTfulサービスの開発を標準化する新しいAPIです。

Oracle WebLogic Server 12.1.2は、Oracle WebLogic Server 12.1.1で開発されたJava EE 6アプリケーションに対して完全な互換性とサポートを提供しており、ユーザーのフィードバックを組み込んでいます。また、Oracle WebLogic Server 12.1.2（およびJava EE 6）は、以前のバージョンのOracle WebLogic Serverで開発されたJava EEアプリケーションとの互換性も提供しています。開発者の観点では、最新の標準と仕様がサポートされているため、スキルを有効に活用でき、今後のプロジェクトにも転用できます。

Webサービス

Oracle WebLogic Serverは、Java EE標準をサポートおよび拡張する、完全に標準に準拠した包括的なWebサービス・インフラストラクチャを備えています。サポートされているおもなJavaプログラミング・モデルは、開発者が使いやすいアノテーション・ベースのJAX-WS Java EE 6標準です。ただし、下位互換性を確保するため、JAX-RPCもサポートされています。サービス品質を確保するため、WS-Policy、WS-PolicyAttachment、WS-Security、WS-SecureConversation、WS-SecurityPolicy、WS-TrustなどのWS-*標準の完全なセットがサポートされています。この実装を補完するのが、Webサービスのポリシー管理とポリシー実施に対応する、Oracle Web Services Managerのきめ細かいサポートです。

WebSocket

WebSocketは、標準のHTML5クライアントでサポートされている新しいプロトコル標準で、Webアプリケーションの重要なプロトコルとして急速に普及しています。WebSocketプロトコルを使用すると、標準のクライアント（最新のブラウザ・バージョンなど）やモバイル・クライアントでアプリケーション・サーバーとの双方向の永続的な接続を作成できます。クライアントは、HTTPを使用して、WebSocket接続を確立するようにリクエストします。WebSocket接続が作成されたら、この接続をクライアントとサーバー・アプリケーション間の双方向のデータ交換に再利用できます。これは、クライアントとサーバー間でデータを転送する前にHTTPヘッダー情報を交換する必要がある、従来のHTTPのアプローチとは対照的です。

低いオーバーヘッド、双方向の永続的な接続、およびHTML5クライアントの標準化という特徴を組み合わせることで、WebSocketは新世代のアプリケーションを構築するための魅力的なプロトコルとなっています。たとえば、HTTPリクエストのフルのオーバーヘッドを生じさせずに、または新しいWebページが生成されて表示されることなく、モバイル・クライアントやデスクトップ・クライアントに軽量のデータ更新を提供するように、ゲーム・アプリケーション、株取引アプリケーション、またはチャット・インターフェースを使用するアプリケーションを開発できます。このようなアプリケーションでは、エンドユーザーのデバイスで実行されているJavaScriptクライアントでサーバーから受け取ったデータ更新を動的に更新したり、表示したりできます。エンドユーザーの観点では、標準のクライアントの応答時間が短くなります。アプリケーション・サーバーの観点では、サーバーで実行される処理が低減されるため、スケーラビリティが大幅に向上升るというメリットが得られます。このようなメリットにより、データ集中型のモバイル対応のタイプのアプリケーションをデプロイおよび拡張できるようになります。

このような新しいタイプのアプリケーションで保証されるスケーラビリティについては、疑問視されることが多いです。クライアントとサーバー間のデータ交換でオーバーヘッドが軽減されるため、スケーラビリティでメリットがあることは認識されていますが、個々のサーバー・インスタンスでの程度の数のオープンなソケット接続をサポートできるのかという疑問があります。多数の同時クライアントをサポートするように、WebSocketアーキテクチャを拡張できるのでしょうか。

当初の社内テストでは、Oracle WebLogic Server 12.1.2インスタンスで、サーバー・インスタンスあたり最大60,000のWebSocket接続を維持できることが示されました。オラクルでは、このことから、すべてのアプリケーションでサーバー・インスタンスあたり最大60,000接続まで拡張できるとは主張していません。ただし、ほとんどの場合に、革新的なWebSocketアプリケーションを提供する上で、Oracle WebLogic ServerのWebSocket接続のスケーラビリティが障壁となるとは考えていません。

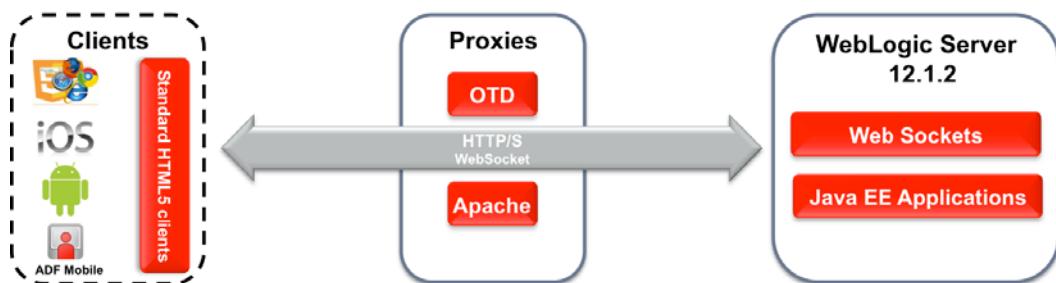


図2：Oracle WebLogic Server 12.1.2とプロキシでのWebSocketプロトコルのサポート

Oracle WebLogic Suite 12.1.2では、Web層のサポートに、Oracle WebLogic Server 12.1.2とApache Web ServerのプラグインのWebSocketプロトコルをサポートしています。また、Oracle Traffic Directorもサポートしており、Exalogicシステムで使用できます。

Oracle TopLinkとTopLink Data Service

Oracle TopLink 12.1.2は、Oracle WebLogic 12.1.2インフラストラクチャに完全に統合されています。Oracle TopLinkはオープンソースEclipse Persistence Servicesプロジェクト (EclipseLink) に基づいており、開発作業と保守作業を軽減して企業アプリケーションの機能を高める実行時機能を備えた、高度なオブジェクト永続化フレームワークです。TopLinkは、幅広いJava EEアーキテクチャとJava SEアーキテクチャで使用するために設計されています。TopLinkのもっとも人気の高い永続化サービスには、次のようなものがあります。

- リレーションナル：Java Database Connectivity (JDBC) を使用してアクセスするリレーションナル・データベースへの、標準のJava Persistence API (JPA) 仕様を使用したJavaオブジェクトの永続性を可能にします。TopLinkは、Oracle Virtual Private Database、XML DB XMLType、フラッシュバック、Oracle Databaseのストアド・プロシージャおよびファンクションに特化してサポートし、業界をリードするすべてのデータベースに高度な機能を提供します。
- TopLink Grid：大規模なクラスタへのJPAアプリケーションのスケールアップとOracle Coherence キャッシュ・オブジェクトの問合せを並列化するためのグリッドの処理能力の活用をサポートする、Oracle Coherenceとの統合を可能にします。

- XML : Java Architecture for XML Binding (JAXB) とService Data Objects (SDO) を使用した、Java オブジェクトとXMLドキュメント間の変換を可能にします。EclipseLinkで提供されているSDOソリューションである、オラクルのSOAインフラストラクチャのコア・コンポーネントは、SDO 2.1.1 のリファレンス実装です。
- NoSQL : この新しい機能により、開発者がNoSQLデータ構造にエンティティをマッピングするのを支援し、問合せとトランザクション操作をネイティブ・データベース操作に変換するのをサポートする、永続化APIを提供します。NoSQLサポートのこの最初のリリースには、MongoDBとOracle NoSQLのプラットフォーム・サポートが含まれており、他のプラットフォームに対応するカスタマイズも可能です。

HTML5クライアントまたはモバイル・クライアントにサービスを提供するアプリケーションの開発にもっとも重要なこととして、以前のバージョンで提供されていたTopLink Data Serviceのサポートが、Oracle TopLink 12.1.2でRESTfulデータ・サービスのサポートを含めるように拡張されています。TopLink Data ServiceでJPAエンティティをREST経由で容易に公開できるようになったため、JavaScriptを使用するHTML5クライアントまたはモバイル・クライアントで、Oracle WebLogic Server 経由でRESTを使用し、データベースに格納されているデータを容易に取得して操作できるようになりました。JPAアプリケーションの開発者は、RESTfulインターフェースを既存のエンティティに宣言的に公開でき、追加のプログラミングは必要ありません。JSONとXMLの両方で、カスタマイズ可能なバインディングがサポートされています。Oracle Databaseの通知のリスニング、RESTfulインターフェースの自動更新などの強力な機能も利用できます。サンプル・アプリケーションで、Oracle WebLogic Serverで提供されている新しいWebSocketプロトコル・サポートを用いてTopLink Data Serviceを使用する方法を示しています。

統合開発環境（IDE）のサポート

Oracle WebLogic Suiteでは、アプリケーション開発者が幅広いIDEから選択できます。Oracle JDeveloper 12.1.2は、Oracle WebLogic Server 12.1.2アプリケーションの開発をサポートするように更新されています。コードの編集、テスト、デバッグ、プロファイリング機能を備えた完全な開発環境を提供し、Oracle Application Development Framework(Oracle ADF)をサポートしているため、Oracle WebLogic Serverでエンタープライズ級のアプリケーションを迅速に構築およびデプロイできます。

Eclipse統合開発環境を好むJava EE開発者に対しては、Oracle WebLogic SuiteはOracle Enterprise Pack for Eclipseを提供しており、Oracle WebLogic ServerでJava EEの開発とデプロイメントに必要なすべてのツールが用意されています。Oracle Enterprise Pack for Eclipse 12.1.2には、Java EE 6アプリケーション開発全体のサポート、WLSTスクリプト作成のサポート、およびこのドキュメントの以降の項で説明する、Managed Coherence Serverの新機能で提供されている新しいGrid Archive (GAR) パッケージ化 のサポートなどの、Oracle Coherenceアプリケーションの開発サポートが含まれています。

また、Oracle NetBeansのIDEでも、Oracle WebLogic Server 12.1.2アプリケーションの開発をサポートしています。Oracle NetBeansは、最新のJava EE仕様のサポートで業界をリードする、オープンソースのIDE実装です。Java EEリファレンス実装であるGlassfishオープンソースに基づいたOracle Glassfish Serverと、Oracle WebLogic Serverの両方をサポートしています。NetBeansとGlassfishを使ってJava EE 6アプリケーションを開発するよう試みたことがある開発者は、Oracle WebLogic Serverアプリケーションの開発にNetBeansを使用するという、自然な進化を体感するでしょう。

軽量のアプリケーション・サーバー

Oracle WebLogic Serverは本番環境で非常に役立つ高度なランタイム機能を備えていますが、これらのすべての機能が開発プロセスに関連するわけではありません。開発者は、開発者のエクスペリエンスをさらに"軽量"にするために、これらの高度な製品機能を隠すことができる、パッケージ化オプションと開発時使用オプションを求めています。オラクルでは、このような機能を開発者に提供するために投資を続けています。

Oracle WebLogic Server 11g以降、オラクルはおもに2種類のソフトウェア・ディストリビューションを提供しています。1つ目はOracle WebLogic ServerとOracle Coherenceの両方が含まれたフル・インストール・プログラムで、開発用と本番用の両方に使用でき、パッチ適用ツールおよびOracle WebLogic Serverで提供されている他の本番環境サポート・ツールによって完全にサポートされています。また、オラクルでは、特に開発用にzipディストリビューションを提供しています。この軽量のディストリビューションは179MBで、ソフトウェアをzipファイルからインストールしたい開発者向けに、ダウンロードとインストール・プロセスをさらに軽量で行えるようにしています。このzipディストリビューションでは、主要な開発プラットフォームとしてWindows、Linux、およびMac OS Xをサポートしています。zipディストリビューションは開発用としてのみサポートされており、本番環境ではサポートされません。また、Oracle WebLogic Server 12.1.2で提供されている新しいパッチ適用ツールを使ってパッチを適用することはできません。

開発者は、両方のタイプのインストールでサービスの起動を選択できます。EJB、JMS、およびJCAの各サービスを使わずに軽量のサーバーを起動するには、起動システム・プロパティとして-DserverType="wls"または-DserverType="wlx"を指定します。たとえば、Webアプリケーションの開発者は、不要なサービスは起動しないコンテナを使って作業するように選択できます。

Oracle WebLogic Server管理コンソールなどの内部アプリケーションは、サーバーの起動にかかる時間をさらに短縮し、リソース消費量を最小限に抑えるように、ニーズに基づいてデプロイされます。Oracle WebLogic Server管理コンソールでは、開発モードでChange Centerが無効になり、確認ダイアログが表示されないため、構成変更を迅速に行えます。

Oracle WebLogic ServerのMavenプラグイン

Apache Mavenは、ソフトウェアの開発操作とビルド操作を自動化するオープンソース・ツールです。Apache Mavenは、アプリケーションのコンパイル、デプロイメント、テストの実行などの一連の操作をMavenプロジェクトで実行するための構造化アプローチを提供します。アプリケーション開発者と組織は、Mavenを採用することで開発、ビルド、およびQAプロセスの効率性、信頼性、および生産性を向上してきました。ソフトウェア・ベンダーは、ベンダーの製品で標準のMaven操作の実行を制御するのに使用できるプラグインを提供することで、Mavenでの製品の使用を簡素化できます。Mavenプラグインの"ゴール"は、アプリケーションのコンパイルなどのMavenプロジェクトの標準機能を実現するために、製品固有の機能とAPIを動作させます。

Oracle WebLogic Server 11g以降、WebLogic Mavenプラグインは、アプリケーション・デプロイメント作業をサポートするために、WebLogic Mavenプラグインのデプロイメント・ゴールをサポートすることで提供されていました。Oracle WebLogic Server 12.1.1では、Mavenプロジェクトでアプリケーション開発に必要となる追加ゴールのサポートを含めるように、WebLogic Mavenプラグインのサポートが拡張されました。ゴールのサポートには、アプリケーションのデプロイメントの他には、ドメインの作成、サーバーの起動と停止、WLSTスクリプトの実行によるサーバー構成の変更、およびアプリケーションのコンパイルが含まれていました。Oracle WebLogic Server 12.1.1で提供されているMavenゴールのサポートでは、Mavenを使用する開発者が必要とする大部分のライフ・サイクル操作について、豊富なサポートを提供しています。

Oracle WebLogic Server 12.1.2のMavenサポートでは、Mavenサポートをさらに一歩進めています。Mavenは、Mavenリポジトリ（ユーザーのシステムのローカル・リポジトリ、リモートの共有プライベート・リポジトリ、またはMaven Central Repositoryなどのパブリック・リポジトリ）に格納されているソフトウェア・モジュールを参照することで、おもにMavenプロジェクトの実行をサポートするように設計されています。Oracle WebLogic Server 12.1.1で提供されているMavenプラグインのサポートは、リポジトリの使用ではなく、ローカルの製品インストールを参照することで機能します。Oracle WebLogic Server 12.1.2では、パッチが適用されたコンポーネントを含め、適切なOracle WebLogic Serverコンポーネントを格納するローカルのMavenリポジトリまたは共有のプライベートMavenリポジトリを設定するために使用できる、新しい同期プラグインが提供されています。Oracle WebLogic Server 12.1.2では、WebLogic Mavenプラグインの可能なゴールが、デフォルトでリポジトリ内のモジュールを参照します。

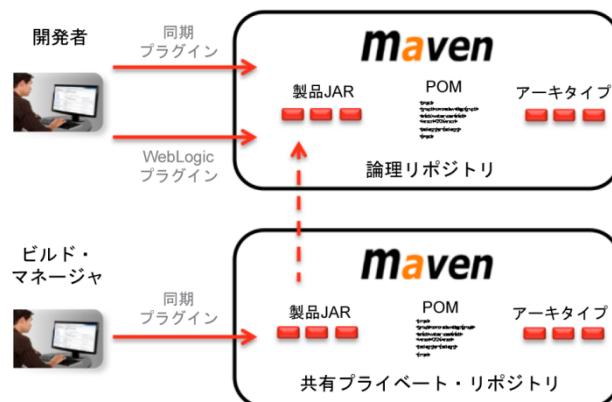


図3：新しいMaven同期プラグインを使用した、Mavenリポジトリの設定

Oracle CoherenceとOracle TopLinkでもサポートされており、Oracle Fusion Middlewareの将来のリリースでさらに広範に採用されるこのアプローチによって、組織では、複数の開発プロジェクトで一貫したソフトウェア・ベンダーとパッチ・レベルを使用でき、さまざまなソフトウェア製品で提供されているソフトウェアを参照するために一貫したアプローチを使用できます。

また、Webアプリケーション、Web/EJBアプリケーション、Webサービス、およびMDBアプリケーションの作成を容易にするために、WebLogic Mavenのアーキタイプも提供されています。

Oracle WebLogic ServerのAntタスク

Oracle WebLogic Serverでは、ライフ・サイクル、パッケージ化、コードのコンパイル、リソースの構成、およびアプリケーションの操作を実行するためのAntタスク一式を引き続き提供しています。これらのAntタスクを使用すると自動スクリプトも生成できるため、繰り返しのタスクや複雑なタスクを簡素化して人的エラーを低減できます。

開発環境とテスト環境の自動化

Oracle WebLogic Serverでは、繰り返しタスクを実行せずに済むように、開発者（および管理者）が日常業務のさまざまな側面を自動化するための一連のツールを提供しています。

たとえば、開発者は、テストと開発環境を迅速にリセットして再作成できる必要があります。これらのタスクを繰り返して実行すると、高いコストがかかる上、エラーが発生しやすくなります。ドメイン・テンプレート、ドメイン・テンプレート・ビルダー、構成ウィザード、およびWebLogic Scripting Tool (WLST) を使用すると、新しいドメインの生成、これらのドメインへのリソース (JDBCリソースやJMSリソースなど) 設定、アプリケーションのデプロイ、および管理コンソールで実行できるその他の操作を行う自動スクリプトを簡単に作成できます。このようなスクリプトは、標準のエディタで作成することも、Oracle Enterprise Pack for Eclipseを使ってIDEサポートで作成することもできます。また、管理コンソールを使用すると、カスタマイズの基盤を提供してスクリプト作成プロセスを簡素化する、再利用可能なWLSTスクリプトに操作を記録できます。

アプリケーションのクラス・ロード

クラス・ロードはアプリケーション・サーバーの複雑な領域であり、正しく理解されていないことが多い領域です。ただし、Oracle WebLogic Serverには、アプリケーションのクラス・ロードを簡単に構成するための、さまざまなメカニズムが用意されています。

まず、Oracle WebLogic Serverでは、複数のアプリケーションでライブラリを共有できるため、このようなアプリケーションの開発が簡素化されます。これは、ライブラリがアプリケーションとは異なるペースで進化する場合に役立ちます。また、アプリケーションごとにライブラリをデプロイする必要がなくなります。

次に、Oracle WebLogic Serverには、アプリケーション・レベルのライブラリが用意されています。これらのライブラリは各アプリケーションにデプロイされ、標準のJava EEクラス・ロード階層でロードされます。つまり、これらのライブラリはサーバーにデプロイされている他のアプリケーションと共有されず、各アプリケーションがライブラリの独自コピーを受け取ります。同じサーバーにデプロイされているアプリケーションで、同じライブラリの異なるバージョンを使用する必要がある場合に便利です。

また、Oracle WebLogic Serverでは、フィルタリング・クラス・ローダーのサポートにより、クラス・ロードを簡素化しています。このクラス・ローダーは、クラス自体はロードしませんが、クラスが親クラス・ローダーによってロードされるのを防ぎます。フィルタリング・クラス・ローダーを使用すると、アプリケーションによってシステム・レベルのクラスを上書きできます。これは、他のアプリケーション・サーバーでは実行するのが難しい操作です。この機能は、アプリケーションで使用されているオープンソースのソフトウェア・コンポーネントが、Oracle WebLogic Serverに組み込まれているこれらのソフトウェア・コンポーネントの異なるバージョンと競合する可能性があるユースケースで重要です。たとえば、アプリケーションで異なるバージョンのXerces、Spring、Ant、またはCommons-Loggingを使用する必要があるとします。このためには、システムのクラスパスからこれらのクラスをフィルタリングするように、フィルタリング・クラス・ローダーを構成します。代わりに、これらのクラスはアプリケーションのライブラリからバンドルされてロードされます。

Oracle WebLogic Serverには、クラス・ロードの問題を分析および解析するための、WebベースのClassloader Analysis Toolが用意されています。上記のようなケースで、アプリケーションで使用されているオープンソース・ソフトウェアとOracle WebLogic Serverのソフトウェアが競合している可能性がある場合、このツールは、このようなクラス競合の検出を支援し、競合を解決するフィルタリング・クラス・ローダー構成を提案します。オープンソース・ソフトウェア・ライブラリを使ったアプリケーション開発プロジェクトでは、クラス・ローダーの問題を早期に検出して解決できるよう、開発プロセスでこのツールを使用することを推奨します。

また、アプリケーションでオープンソース・ソフトウェアを使用している場合は、これらのアプリケーションをアップグレードする際、または既存のアプリケーションを新しいバージョンのOracle WebLogic Serverにアップグレードする際にも、このツールを使用する必要があります。このようなアップグレードでは、サーバーまたはアプリケーションで使用されているオープンソース・ソフトウェア自体のバージョンがアップグレードされることがあります。そのため、アプリケーションで使用されているオープンソース・ソフトウェアのバージョンがサーバーで使用されているバージョンと異なってしまい、アップグレード前には存在しなかったクラス競合の問題が生じる可能性があります。Classloader Analysis Toolを使用すると、アップグレード・プロセスでこのような問題を特定して解決できます。

ネイティブなクラウド管理

多くのリリースを通じて、Oracle WebLogic Serverは、アプリケーション環境の管理を簡素化し、基盤となるアプリケーション・インフラストラクチャの保守コストを最終的に削減する、強力な管理ツールを提供してきました。Oracle WebLogic Suite 12.1.2のネイティブなクラウド管理とは、Oracle WebLogic Serverインフラストラクチャに組み込まれ(つまりネイティブであり)、多くの場合、Oracle WebLogic Suiteの一般的なデプロイメント・トポロジでWeb層とCoherence層にまたがって一貫した管理ツールを提供するように拡張された、管理ツールの機能強化を指しています。Oracle WebLogic Suite 12.1.2のネイティブなクラウド管理では、Webサーバー層、アプリケーション・サーバー層、およびデータ・グリッド層にまたがって共通の一貫した、効率性に優れたツールが用意されているため、大規模に拡張したクラウド・インフラストラクチャを管理できます。

インストールとパッチ適用の統合

ユーザーがクラウド・インフラストラクチャを計画および実装するにつれ、アプリケーション・インフラストラクチャ製品で使用される、基盤となるインストール・テクノロジーとパッチ適用テクノロジーがますます重要になってきています。新しいインスタンスのプロビジョニングを簡素化することがクラウド・インフラストラクチャの主目標の1つであるため、クラウド管理者は、パッチによって多数のインストールを管理および保守することを期待しています。一貫したインストール・ツールとパッチ適用ツールを使用すると、このようなクラウド環境の管理が簡素化されます。

Oracle WebLogic Suite 12.1.2より前のリリースでは、オラクルは、製品スイートでインストール・テクノロジーを一部統合していました。たとえば、Oracle TopLinkは常にOracle WebLogic Server 11gおよび12.1.1の一部としてインストールされ、Oracle WebLogic Server 11gと12.1.1のインストーラにはOracle Coherenceが含まれていました。ただし、この統合は完全ではありませんでした。特に、Oracle WebLogic Serverでは、オラクルが製品を買収する前に開発されていたインストール・テクノロジーとパッチ適用テクノロジーを使用していました。つまり、Oracle WebLogic Serverのパッチ適用テクノロジーとインストール・テクノロジーは、標準のOracle Universal InstallerとOPatchパッチ適用ツールとは異なっていました。

Oracle WebLogic Suite 12.1.2で、Oracle WebLogic Suiteのすべてのコンポーネントでインストール・テクノロジーとパッチ適用テクノロジーを統合しました。Oracle WebLogic Server、Oracle Coherence、Oracle TopLink、Oracle HTTP Server、Oracle ADF、Oracle JDeveloper、およびOracle Enterprise Pack for Eclipseはすべて、Oracle Universal InstallationとOPatchのインストール・テクノロジーとパッチ適用テクノロジーに基づいて標準化されています。つまり、クラウド管理者は、一貫したテクノロジーを使用して、Oracle WebLogic Suiteに基づいたクラウド・インフラストラクチャでインストールと保守を簡素化し、効率よく実行できます。

WebLogic管理フレームワーク

WebLogic管理フレームワークは、Oracle WebLogic Serverの一部として提供されている管理機能で構成されています。WebLogic管理フレームワークは、所有コストを低減する効率的な管理環境を提供する上で不可欠です。Oracle WebLogic Suite 12.1.2では、WebLogic管理フレームワークで提供される機能が引き続き拡張されており、Oracle HTTP ServerとOracle Coherenceの管理に対応するようフレームワークの範囲が拡張されています。

WebLogic管理フレームワークは、以降で説明する次のコンポーネントで構成されています。

- ドメイン構成（管理サーバーと管理対象サーバーを含む）
- クラスタ
- デプロイメント・インフラストラクチャ
- ノード・マネージャ
- 構成ウィザードとドメイン・テンプレート・ビルダー
- WebLogic管理コンソールとOracle Fusion Middleware Control
- WebLogic Scripting Tool
- WebLogic診断フレームワーク、Java Mission Control、Java Flight Recorder
- RESTful管理サービス

ドメイン構成（管理サーバーと管理対象サーバーを含む）

Oracle WebLogic Serverの構成と管理の基本単位はドメインです。ドメインは、一連の管理対象サーバー（Oracle WebLogic Serverインスタンス、またはアプリケーションをホストするプロセス）と管理サーバー（ドメイン内のすべてのサーバーの構成を管理）で構成されます。1つのドメインに管理サーバーは1つのみ含まれます。1つのドメインを、管理サーバーの機能と管理対象サーバーの機能を組み合わせた1つの開発サーバーで構成したり、1つの管理サーバーと複数の管理対象サーバーで構成したりできます。管理サーバーは管理対象サーバーにドメイン構成情報を配信し、標準のJMXの上に構築されたプロトコルを使用することで、構成の変更をドメイン全体に確実に伝達します。管理サーバーを使用するためのランタイム要件はありません。管理サーバーが使用できない場合は、管理対象サーバーにデプロイされているアプリケーションによって引き続き、アプリケーション・サービスが中断なしに提供されます。これらのドメインで実行される管理対象サーバーは、クラス化されている場合とされていない場合があります。

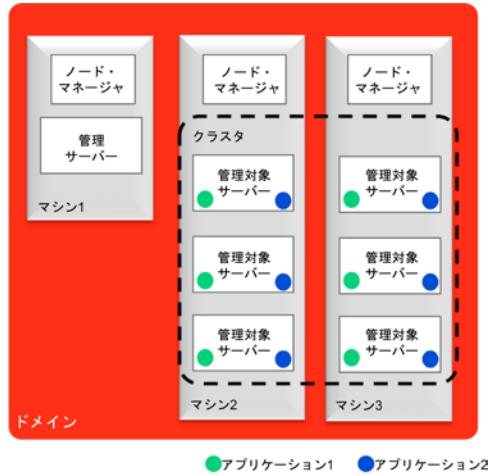


図4：管理サーバー、クラスタ化された管理対象サーバー、アプリケーション、ノード・マネージャによるドメイン構成

クラスタ

Oracle WebLogic Serverクラスタは、クライアントに対して、複数の管理対象サーバーに分散されているアプリケーションの単一の論理ビューを提供するとともに、ミッション・クリティカルなアプリケーションにスケーラビリティと可用性のメリットを提供します。ただし、クラスタは管理性のメリットも提供します。

クラスタ化されたサーバーは、同種または均一となることを目的としています。つまり、同じバージョンのOracle WebLogic Serverを実行しており、同じ構成が含まれておらず（このドキュメントで後述するように、いくつか例外あり）、同じアプリケーションがデプロイされている必要があります。リソースの構成やデプロイメントなどのさまざまな管理操作についてクラスタを対象にすると、管理効率を向上できます。

デプロイメント・インフラストラクチャ

Oracle WebLogic Serverのデプロイメント・インフラストラクチャは、アプリケーションをホストする管理対象サーバー・インスタンスでWAR、EAR、RAR、JARなどのデプロイメント・アーカイブを利用できるようにし、アプリケーションの実行にこれらのデプロイメント・アーカイブを準備します。クラスタ構成では、Oracle WebLogic Serverのデプロイメントによって、クラスタ内のすべてのサーバーでデプロイメントが正常に完了されます。管理対象サーバーにデプロイメント・アーカイブを配布するために、複数のステージ・モードがサポートされています。既存のクライアントへのアプリケーション・サービスを中断せずにアプリケーション・バージョンを更新する本番環境の再デプロイメントなどの、高度なデプロイメント機能もサポートされています。

ノード・マネージャ

ノード・マネージャは、WebLogic管理フレームワークのオプション・コンポーネントです。ノード・マネージャは、特定のマシンのサーバー・インスタンスで起動や停止などのサーバー・ライフ・サイクル操作を監視および制御するために設計された、軽量のプロセスです。以前のリリースのノード・マネージャの構成モデルでは、複数のドメインでノード・マネージャのインスタンスを共有していました。Oracle WebLogic Server 12.1.2では、デフォルトの構成は、各管理対象サーバー・マシンのドメインあたりノード・マネージャ・インスタンスが1つとなっています。管理サーバーはノード・マネージャを介してサーバー・インスタンスの起動と停止を制御でき、失敗したサーバー・インスタンスの自動再起動など、特定の操作を実行するようにノード・マネージャを構成できます。

構成ウィザードとドメイン・テンプレート・ビルダー

構成ウィザードは、Oracle WebLogic Serverドメイン、およびOracle WebLogic Server上に構築したOracle Fusion Middleware製品のドメインの初回作成時に使用する、GUIおよびコマンドライン・ツールです。構成ウィザードを使用すると、テンプレートに基づいてドメインが作成されます。関連するドメイン・テンプレート・ビルダーを使用すると、エンドユーザーはテンプレート自身を作成できます。そのため、ドメイン構成を一度作成すると、構成ウィザードのテンプレートでこのドメイン構成を取得して、別のマシンでこのドメインを再作成できます。このツールは、簡単に使用できる、ドメインの初回作成メカニズムを提供し、クラウド環境で複雑なドメイン再作成やレプリケーションを行うための強力な機能を提供します。

WebLogic管理コンソールとOracle Fusion Middleware Control

Oracle WebLogic Suite 12.1.2には、実行時にOracle WebLogic Serverドメインを管理するために、2つのコンソールが用意されています。Oracle WebLogic Server管理者が使用するおもなコンソールはWebLogic管理コンソールです。管理コンソールは、管理サーバーで実行されるWebアプリケーションです。このコンソールは、ドメイン構成の管理、サーバーの起動と停止、アプリケーションのデプロイメント、およびドメイン環境の監視に必要なすべての機能を備えています。また、Oracle WebLogic Suite 12.1.2は、Oracle Fusion Middleware製品の個々のドメインを管理するための戦略的なコンソールである、Oracle Fusion Middleware Controlも備えています。Fusion Middleware Controlで提供されているOracle WebLogic Server管理機能の範囲は、WebLogic管理コンソールで提供されている機能の範囲とまだ同等ではありませんが、サーバーの監視、ライフ・サイクルの管理、アプリケーションのデプロイメント、一般的な構成要件などの広い領域に対応しています。

WebLogic Scripting Tool (WLST)

WebLogic Scripting ToolはPythonベースのスクリプト作成環境であり、Oracle WebLogic Server管理のすべての側面を自動化するために使用できます。Oracle WebLogic Server管理を自動化したり、必要な構成をレプリケートするためにWLSTスクリプトを使って"空"のドメインを変更することを許可したり、すべてのターゲット・ドメインで一貫した構成変更を行うために複数のドメインにまたがって管理タスクの均一なセットを実行したりする際に、よく使用されているツールです。開発に関する前項で述べたように、WLSTスクリプトの作成は、手動で行うこととも、WebLogic管理コンソールで実行した操作を記録して行うこととも、Oracle Enterprise pack for Eclipseでスクリプトを作成して行うこともできます。

WebLogic診断フレームワーク、Java Mission Control、Java Flight Recorder

WebLogic診断フレームワーク (WLDF) は、サーバーとアプリケーションのパフォーマンスに関連するメトリックを収集および表示するために設計された、監視と診断のフレームワークです。WLDFを使用すると、サーバーとアプリケーションの実行で生成された診断データを作成、収集、保存、および分析できます。メトリックおよび定義したルールに基づいて通知を送信するように、WLDFの監視を構成できます。WLDFのデータを管理コンソールでグラフィカルに表示して確認できるため、管理者はサーバーの重要なメトリックを監視できます。

また、WLDFは、Oracle JDKに関する2つの機能Java Mission ControlおよびJava Flight Recorderと連携して動作するように設計されています。Java Flight Recorderは、WLDFで生成されたイベント・データなど、ここ数分間に実行されたサーバー・プロセスに関する診断情報をローリング・インメモリ・バッファ内に保持するように設計されています。Flight Recorderのバッファは、Java Mission ControlのGUIを使ってインシデント後の分析用にいつでもディスクに保存できるため、サーバーで何が発生したために問題のインシデントに至ったのかを分析できます。WLDFとFlight Recorderは独自の診断機能を備えており、定期的に繰り返し発生するが、分析目的で再現するのが難しい問題に特に役立ちます。WLDFとFlight Recorderを使用すると、たとえば、問題に関連しているサーバーのメトリックに基づいて監視を構成できます。WLDFの通知の発行によって、保持されているFlight Recorder/バッファを含め、WLDFの診断スナップショットの取得をトリガーできます。このアプローチで、イベント発生時にサーバーで何が発生したのか、およびサーバーで何が発生したためにイベントに至ったのか、詳しい情報が自動的に取得されるため、問題の根本的な原因を解明するのに役立ちます。

RESTful管理サービス

Oracle WebLogic Serverの最近のリリースに含まれている新しい管理機能がRESTful管理サービスです。オラクルでは、RESTfulインターフェースの使用が、アプリケーション・データにアクセスする手段としてますます一般的になってきていることを認識しています。RESTful管理サービスの場合、このことは、RESTfulのアクセスをドメイン内のサーバー、クラスタ、アプリケーション、データソースなどのOracle WebLogic Server監視データに提供することを意味します。この機能は明示的に有効にする必要があります。また、インターフェースへのアクセスは標準の管理者ロールに基づいて制限されます。オラクルでは、アプリケーション、サーバー、およびサーバー・リソースが実行されているかどうかや、正常な状態であるかどうかを監視するのに使用できる軽量の管理インターフェースを構築する際に、この機能が役立つと確信しています。JSON、XML、HTMLの表示形式がサポートされています。

このWebLogic管理フレームワークの機能概要では、詳細を示すことを目的としていません。Oracle WebLogic Serverで提供されているネイティブな管理機能の範囲を示すことを目的としています。これらの機能は、お客様の本番環境で長年使用されることで実証されており、プライベート・クラウド環境とパブリック・クラウド環境の構築と管理にも適用されています。Oracle WebLogic Serverの上に構築され、Oracle Cloud製品全体に含まれているOracle Java Cloud Serviceでは、これらの同じ機能を利用しています。このドキュメントの次の項では、Oracle WebLogic Suite 12.1.2インフラストラクチャ全体のネイティブなクラウド管理の基盤を提供するために、Oracle WebLogic Suite 12.1.2の他のコンポーネントの管理に対応するよう、WebLogic管理フレームワークの範囲がどのように拡張されたのかを説明します。

Managed Coherence Server

Managed Coherence Serverは、Oracle CoherenceとOracle WebLogic Suite 12.1.2の新機能です。Managed Coherence Serverは、WebLogic管理フレームワークでOracle Coherenceインスタンスを管理するためのオプションを提供します。これはオプション機能であるため、WebLogic管理フレームワークを使わずにOracle Coherenceクラスタを管理するOracle Coherenceユーザーは、使用する必要はありません。ただし、Oracle WebLogic ServerとOracle Coherenceを組み合わせて使用するユーザーや、オラクルが提供する管理インフラストラクチャをOracle Coherenceに使用するユーザーは、Managed Coherence Serverを使用する必要があります。

Managed Coherence Serverの基本的な設計アプローチは、Oracle WebLogic Serverの管理対象サーバー（Java EEアプリケーションがデプロイされている管理対象サーバーまたはJava EEアプリケーションがデプロイされていない管理対象サーバー）のプロセス内でOracle Coherenceクライアントまたはサーバーを実行するというものです。この設計アプローチにより、管理対象サーバーで実行されているOracle Coherenceインスタンスでは、次の機能を含め、WebLogic管理フレームワークの利点をすべて活用できます。

- Managed Coherence Serverの構成は、構成ウィザードを使って作成でき、必要に応じてユーザーが作成したドメイン・テンプレートからレプリケートできます。

- Managed Coherence Serverが含まれているドメインは、WebLogic管理フレームワークの標準のライフ・サイクル・ツールを使って起動および停止できます。たとえば、管理コンソールまたはノード・マネージャ経由で通信するWLSTスクリプトを使って、Oracle Coherenceインスタンスとクラスタを起動できます。
- Managed Coherence Serverインスタンスを自動的に再起動するように、ノード・マネージャを構成できます。
- 管理コンソール、JMX、またはWLSTスクリプトを使って、実行しているサーバーで構成変更を実行できます。
- WebLogic管理フレームワークのデプロイメント・インフラストラクチャを使って、Oracle Coherenceアプリケーションをデプロイできます。Managed Coherence Serverインスタンスで実行するアプリケーションを準備するため、WebLogicデプロイメント・インフラストラクチャによって認識される、Grid Archive (GAR) と呼ばれる新しいタイプのデプロイメント・アーカイブが作成されています。
- 管理コンソール、JMX、Oracle Fusion Middleware Control、Oracle Enterprise Manager、およびWLSTを使って、Managed Coherence Serverとクラスタを管理できます。

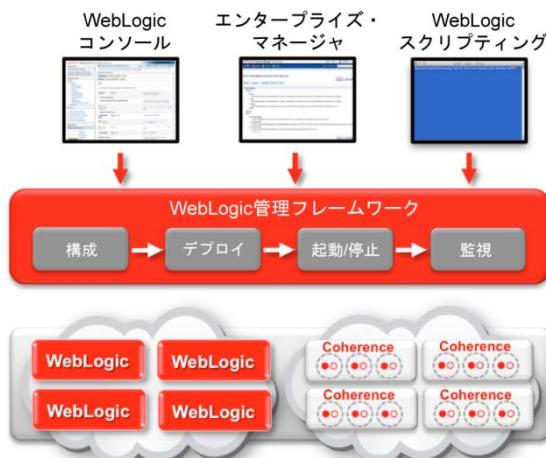


図5：Managed Coherence Server – Oracle WebLogic ServerとCoherenceの管理の統合

Managed Coherence Serverは、さまざまなOracle WebLogic Suiteトポロジをサポートします。具体的には、スタンドアロンのOracle Coherenceクラスタ（Java EEアプリケーションがデプロイされていない）や、単一のOracle Coherenceクラスタ内ですべて実行されているストレージ対応のOracle Coherenceインスタンスにアクセスする、ストレージ非対応のOracle Coherenceインスタンスが含まれたクラスタ化されたJava EEアプリケーション、または、共有のOracle Coherenceクラスタ内のデータにアクセスする複数のJava EEクラスタなどの構成があります。

管理対象サーバー内でOracle Coherenceを実行する際に生じるオーバーヘッドをユーザーが懸念している場合、オラクルは、このオーバーヘッドは最小限であると考えています。社内の測定では、管理対象サーバーのメモリ消費量の増加は約70~80MB、つまり4GBのヒープの約2%と示されています。オラクルでは、Managed Coherence ServerはOracle WebLogic Suiteのユーザーがすぐに採用する強力な機能であると考えています。

Oracle HTTP Serverの管理

以前のバージョンのOracle HTTP Serverでは、Oracle HTTP Serverインスタンスの構成と管理に別々のフレームワークを使用していました。この12.1.2リリースでは、Oracle HTTP Serverインスタンスは、次のようにWebLogic管理フレームワークを使用して管理します。

- Oracle HTTP Server構成は、"Oracle WebLogic Server ドメイン"または"スタンドアロンのドメイン"で作成できます。これらのドメイン構成は、構成ウィザードで作成できます。
- Oracle WebLogic Server ドメイン構成では、Fusion Middleware ControlまたはOracle WebLogic Server管理サーバーで実行されるWLSTスクリプトを使用した、Oracle HTTP Serverインスタンスの管理機能をサポートしています。このような構成には、Oracle WebLogic ServerとOracle CoherenceだけでなくOracle ADFとFusion Middleware Controlも含まれた、FMWインフラストラクチャの完全なインストールが必要です。Oracle HTTP Serverのプロセス制御は、ノード・マネージャを使って管理します。このタイプのドメイン構成には、Java EEアプリケーションとManaged Coherence Serverをホストする管理対象サーバーを含めることができます。
- スタンドアロンのドメイン構成は、Oracle HTTP Serverなどの（Java以外の）システム・コンポーネント用の構成です。スタンドアロンのドメイン構成オプションの使用は、Fusion Middleware Controlの使用が望ましくない場合または不要な場合に適しています。スタンドアロンのドメインは、WLSTコマンドラインとノード・マネージャを使って管理できます。

同じWebLogic管理フレームワーク・テクノロジーを使用してOracle WebLogic Server、Oracle TopLink、Oracle Coherence、およびOracle HTTP Serverを管理するため、Oracle WebLogic Suiteは、一般的なOracle WebLogic Suiteトポロジの主要な要素に実績のある一貫したネイティブな管理フレームワークを提供する、独自の機能を備えています。このようなアーキテクチャは、特に、ビジネス・ユーザーのニーズを満たすよう拡張する必要があるプライベート・クラウド・インフラストラクチャなどのインフラストラクチャにおいて、信頼性、効率性、所有コストの低減を含め、管理性で大きなメリットを提供します。

Elastic JMS

Oracle WebLogic Serverを超えて他の製品をサポートするようにWebLogic管理フレームワークを拡張する以外に、オラクルでは、Oracle WebLogic Server自体の管理効率を向上するために、WebLogic管理フレームワークを引き続き改良しています。Oracle WebLogic Server 12.1.2で行ったこのような改良の例には、JMS構成およびJMS構成の拡張を簡素化する、Elastic JMSという名前の新しい構成機能があります。

Oracle WebLogic Server 12.1.2より前のバージョンでは、個々の管理対象サーバーでのみJMSサーバーとJMSストア（ファイル・ストアまたはデータベース・ストア）の構成を対象にできました。また、サブデプロイメント、つまり問合せやトピックの対象をJMSサーバーにする場合、JMSサーバー固有の構成が必要になりました。つまり、クラスタ化されたJMSサーバー構成を作成した場合でも、これらのクラスタ内でJMSを構成するには、管理者がサーバー固有の構成手順を実行する必要がありました。管理コンソールまたはWLSTでこのような構成を作成できますが、このような構成を作成するのは、本質的に必要以上に複雑です。特に、多数の管理対象サーバー、JMSサーバー、およびサブデプロイメントに対応するようにJMS構成を拡張する場合に顕著です。

Elastic JMSは、JMSサーバーとJMSストアの両方の構成をクラスタの対象とする、新しいオプションを提供します¹。これにより、JMSクラスタ構成のプロセスが大幅に簡素化されます。管理対象サーバーがN個のクラスタでは、N個ではなく1つのJMSサーバーと1つのJMSストアのみを対象にする必要があります。クラスタ化された単一のJMSサーバーで下位デプロイメントを対象にできるため、下位デプロイメントの構成も簡素化されます。これらの構成の作成だけでなく、これらの構成の拡張も簡素化されます。クラスタ構成に管理対象サーバーを追加すると、管理対象サーバーの新しい数に合わせて、JMSサーバー、ストア、および下位デプロイメントの構成が自動的に拡張されます。その結果、JMS構成の管理効率が向上し、クラウド環境を管理するWebLogic管理フレームワークが大幅に強化されます。

Oracle Enterprise Manager Cloud Control

前の項で説明したネイティブなクラウド管理機能はすべて、Oracle Enterprise Manager Cloud Controlを使用して補完できます。Oracle Enterprise Managerは、オラクル製品のポートフォリオ全体に渡る戦略的な管理プラットフォームで、アプリケーションからディスクまで、Oracle STACKのエンド・ツー・エンドの管理を提供します。Oracle Enterprise Manager Cloud Controlでは、2つの管理パックによってOracle WebLogic Serverのサポートを提供しています。

Oracle WebLogic Server Management Pack Enterprise Editionは、複数のOracle WebLogic Server ドメイン構成の管理と監視をすべて1つの管理インターフェースでサポートすることで、独自の価値を提供します。多数のドメインに拡張することが見込まれるクラウド・インフラストラクチャでは、環境内のすべてのドメインを監視する中央コンソールを使用できるため、管理効率を大幅に向上させることができます。パフォーマンス監視機能と診断機能を使用すると、アプリケーション・サービスを監視でき、サービス・レベルを満たしていない場合は、根本的な問題をドリルダウンして特定できます。構成管理機能を使用すると、管理者は、環境全体で構成を追跡および保守し、標準を確実に遵守して、不適切な構成が原因で生じる停止時間を最小限に抑えることができます。ライフ・サイクル管理を使用すると、堅牢なクラウド・インフラストラクチャを維持する上で不可欠な、分散インストールやパッチ適用などのタスクを自動化できます。ここでは、提供される機能の範囲のみ簡潔に説明しています。

¹ Elastic JMSには、順序単位、作業単位、ストア・アンド・フォワード・サービス、サーバーとサービスの移行などの高度なJMS機能のサポートはまだ含まれていません。

Oracle Cloud Management Pack for Fusion Middlewareは、Oracle WebLogic Server Management Pack Enterprise Editionのアドオンです。このアドオンを使用すると、IT管理者は、開発、QA、または本番環境で使用するためにビジネス・ユーザーがOracle WebLogic Server ドメインをプロビジョニングできる、セルフサービスのプライベート・クラウド・インフラストラクチャを作成できます。管理インターフェースが簡素化されているため、ビジネス・ユーザーは、作成したサービスのセルフサービス管理に必要な基本的な管理タスクを簡単に実行できます。クラウド・リソースの消費量を示す事前定義されたメトリックに基づいてIT組織がエンドユーザーに課金できるよう、測定機能が提供されています。クラウド・インフラストラクチャの管理戦略を作成しているユーザーは、Enterprise Manager Cloud Controlの機能を評価する必要があります。

ミッション・クリティカルなクラウド・プラットフォーム

生産性に優れた最新の開発プラットフォームと効率性に優れたネイティブなクラウド管理機能を提供することは必要不可欠ですが、クラウド・インフラストラクチャに対するビジネス・ニーズを満たすには十分ではありません。ほぼすべてのビジネスで、お客様やエンドユーザーが経験するサービスの品質が、成功を決定するもっとも重要な要素の1つとなっています。クラウド・インフラストラクチャにデプロイされるアプリケーションとサービスが、パフォーマンス、可用性、スケーラビリティ、および信頼性においてもっとも高いサービス・レベルを提供する必要があります。これらのサービス・レベルをサポートするためには、クラウド・プラットフォームの特性に頼る必要があります。Oracle WebLogic Suite 12.1.2の目標は、リスクとコストを最小限に抑えながら、もっとも高いアプリケーション・サービス・レベルを提供することです。

競合他社に対する優位性の実証

中間層でアプリケーション・インフラストラクチャのパフォーマンスが向上するため、ユーザーに最高のパフォーマンスを提供できるだけでなく、ハードウェア・リソースへの投資を削減できます。Oracle WebLogic Serverは、アプリケーション・サーバー市場でパフォーマンスを先導していることが実証されており、業界標準のSPECjEnterprise 2010ベンチマークで測定されているように、複数のカテゴリで世界記録を更新し続けています。次のグラフは、最近のおもな結果を示しています。



総合的なEjOPSの最高値：IBMと比較して3.4倍のワークロード



仮想化されたプロセッサあたりの総合的なEJOPSの最高値：IBMと比較してプロセッサあたり2.2倍のワークロード。複数のカテゴリでOracle WebLogic ServerがIBM WebSphere Application Serverを上回る（出典：<http://www.spec.org/jAppServer2010/results/>）

これらの結果は、ますます高まるパフォーマンス要件を満たすための、長年に渡る製品の最適化を反映しており、オラクルのお客様が達成した卓越したパフォーマンス・レベルとも一致しています。Oracle WebLogic Serverに組み込まれているこれらのパフォーマンス機能により、クラウド・アプリケーションでメリットが得られます。

ワーク・マネージャとサーバー・パフォーマンス

Oracle WebLogic Serverの重要な機能は、市場の他のサーバーよりも優れたパフォーマンスを発揮することだけではありません。優れたパフォーマンス機能に加えて、Oracle WebLogic Serverは、複雑で反復的なシステム・チューニング手順を実行しなくても、Oracle WebLogic Serverでユーザーが最適なパフォーマンスを達成できるようにする、自己チューニング機能も提供しています。Oracle WebLogic Serverの自己チューニング機能は、ワーク・マネージャに基づいています。

アプリケーション・サーバーを最適に実行するためには、正しいアプリケーション・リクエスト・スレッド数でアプリケーション・サーバーを構成する必要があります。リクエスト・スレッド数はハードウェア、アプリケーション、および基盤となるJVMによって大きく異なるため、運用環境で最適なアプリケーション・スループットを実現するのに必要となる正確なスレッド数を判断するのは難しいです。Oracle WebLogic Serverには1つのスレッド・プールがあり、このプールですべてのタイプの作業が実行されます。Oracle WebLogic Serverは、ワーク・マネージャを使用して、実行時のメトリックに対して定義されているポリシーに基づき、リクエストを実行するのにかかる実際の時間、プールでリクエストが送受信される速度など、作業に優先順位を設定します。

ワーク・マネージャは、管理コンソールを使ってドメイン、アプリケーション、またはモジュール・レベルで構成できます。次のさまざまなポリシーと制約を使用できます。

- Fair-share-request: リクエストを処理するために必要な平均スレッド使用時間を指定します。ワーク・マネージャは、これを使用して、さまざまなモジュールに対するリクエストに、指定の割合の比率の計算時間を確保します。
- Response-time-request: 応答時間の目標をミリ秒単位で指定します。ワーク・マネージャは、これを使用して、特定のリクエストの応答時間が長くならないように、さまざまなタイプのリクエストを調整します。

- Context-request：現在のユーザー、または現在のユーザーのグループなどのコンテキスト情報に基づいて、リクエストにリクエスト・クラスを割り当てます。これにより、追加の制約を適用できるようになります。
- Max-threads-constraint：同時実行の数が制限を下回るまで、このタイプのリクエストのスケジュールをサーバーが設定するのを防ぎます。サーバーは、fair-shareまたはresponse-timeの目標に基づいて作業のスケジュールを設定します。
- Min-threads-constraint：対象となるリクエストに対してサーバーが割り当てるスレッド数を保証し、デッドロックを回避します。
- Capacity：容量に達したら、サーバーがリクエストを拒否するようにします。

上記のポリシーと制約を組み合わせて自己チューニングに使用すると、スレッド・プールを動的に調整できます。スレッド・プールは、作業のスケジュールを設定するとともに、同時実行の制約に基づいて最適なパフォーマンスを達成するよう、スレッド数を増加または低減させます。ワーク・マネージャは、全体的なスループットを監視し、収集されたデータを使って、最適なパフォーマンスを達成するためにスレッド数を調整する必要があるかどうかを判断します。その結果、反復的な手動チューニングに必要な管理オーバーヘッドを低減しながら、最適なパフォーマンスと優れたアプリケーション・サービスを達成します。

Oracle WebLogic Serverのクラスタリング

Oracle WebLogic Serverの個々の管理対象サーバーでも高いパフォーマンスを発揮しますが、Oracle WebLogic Serverアプリケーションのほとんどは、アプリケーション・サービス・レベルを高めるために複数のサーバーにデプロイされます。

Oracle WebLogic Serverクラスタは、スループット、スケーラビリティ、および信頼性を高めるために同時に実行され、連携して動作する、複数の管理対象サーバーで構成されます。クライアントから見た場合、クラスタは単一のOracle WebLogic Serverインスタンスとして示されます。クラスタを構成するサーバー・インスタンスは、同じマシンで実行することも、異なる物理マシンまたは仮想マシンに配置することもできます。クラスタの能力を拡張するには、クラスタ内のサーバー数を増やします。

クラスタリングのおもなメリットは、次のものを提供できることです。

- スケーラビリティ：アプリケーションのパフォーマンス要件を満たすよう、Oracle WebLogic Serverクラスタにデプロイされているアプリケーションを拡張できます。多くのお客様は、2~4台の管理対象サーバーで構成されたクラスタを使用しています。オラクルでは通常、33台以上の管理対象サーバーで構成された、クラスタ化されたJava EEアプリケーションを拡張する場合は、ストレス・テストを慎重に行うことを推奨しています。ただし、クラスタの拡張には技術的な制限は特にないため、お客様は本番環境で大規模なクラスタ構成を実行しています。ほとんどのアプリケーション・パフォーマンス要件とスループット要件において、Oracle WebLogic Serverクラスタは、これらの要件を満たすように拡張できます。また、他のアプリケーション・サーバー・テクノロジーのパフォーマンス・レベルを超えて拡張できます。

- 高可用性：クラスタ内にデプロイされているアプリケーションでは、さまざまなフェイルオーバー・メカニズムにより、管理対象サーバーまたはマシンで障害が発生した場合でも、アプリケーション処理を継続して実行できます。

クラスタの主要な機能には、次のようなものがあります。

- セッション・フェイルオーバー：クラスタでは、HTTPセッション・ステートとEJBセッション・ステートが、クライアントがアクセスしているプライマリの管理対象サーバーからクラスタ内のセカンダリ・サーバーに自動的にレプリケートされます。プライマリ・サーバーで障害が発生した場合、クライアントはセカンダリ・サーバーに透過的に再接続されるため、セッション・ステートの中断や損失なしにアプリケーション処理を続行できます。
- ロードバランシング：アプリケーションを適切に拡張するためには、作業および関連するネットワーク接続をコンピュータ・リソースとネットワーク・リソースで均等に分散する必要があります。Oracle WebLogic Serverは、適切な作業割当てが行われるように、ロードバランシングを複数レベルで提供しています。

Oracle WebLogic Serverは、Servlet、JSP、EJB、RMIオブジェクト、JMS宛先、JDBCリソースなどのさまざまなJava EEアプリケーション・リソースをクラスタ化する機能を備えています。

サーバーとサービスの移行

Oracle WebLogic Serverのクラスタリングの高度な機能に、サーバーとサービスの移行があります。これは、クラスタ内にデプロイされたシングルトン・サービスに適用される、クラスタ・フェイルオーバー・テクノロジーです。

シングルトン・サービスは、常にクラスタの単一のノードにのみ存在するサービスです。この例として、プロデューサからの永続的な送信操作のたびにアプリケーション・メッセージの1つのコピーだけがストレージに保持されるようにする、JMSサービスがあります。ただし、シングルトン・サービスをホストする管理対象サーバーまたはマシンで問題が発生した場合、問題が解決するまでサービスを使用できなくなる可能性があります。Oracle WebLogic Serverで現在提供されている一連のシングルトン・サービスは、以下の広いカテゴリにグループ化できます。

- 永続ストア・サービス
- JMS関連サービス
- JTA関連サービス
- ユーザー定義のシングルトン・サービス

これらのサービスは、上記のタイプの障害シナリオの影響を受ける可能性が高いです。この点と、アプリケーション・サービス・レベルへの影響に対処するため、Oracle WebLogic Serverでは、移行と呼ばれるソリューションを提供しています。Oracle WebLogic Serverの移行とは、障害発生時に、クラスタ化されたOracle WebLogic Serverインスタンス、またはクラスタ化されたインスタンスで実行されていたサブシステム・コンポーネントを別の場所に移動するプロセスのことです。障害が発生した場合に特定の物理マシンから別の物理マシンにサーバー・インスタンス全体を移動するプロセスは、Whole Server Migration (WSM) と呼ばれます。

また、特定のサーバー・インスタンスから別のサーバー・インスタンスに影響を受けたサブシステム・サービスだけを移動するプロセスは、Service Migrationと呼ばれます。

サーバー移行全体で、移行可能なサーバーがなんらかの理由（ハング、ネットワークの接続の切断、ホスト・マシンの障害など）で使用できなくなった場合は、サーバー・インスタンスが自動的に移行されます。可能であれば、移行対象のサーバーが同じマシンで自動的に再起動されます。障害が発生したマシンで移行可能なサーバーを再起動できない場合は、別のマシンに移行されます。また、管理者は、サーバー・インスタンスの移行を手動で開始できます。

Service Migrationは、障害が発生したサブシステム・サービスのみを、使用できなくなったサーバー・インスタンスから使用可能なサーバー・インスタンスに移行するために必要なインフラストラクチャを提供します。サービスを手動で移行することは可能ですが、Automatic Service Migration (ASM) フレームワークは、シングルトン・サービスの状態を事前予防的に監視し、障害が発生したサービスをクラスタ内での使用可能なサーバー・インスタンスに自動的に移行します。このため、移行の実行時間が短くなり、これらのサービス全体の可用性が向上します。

ローリング・アップグレード

ローリング・アップグレードは、1回限りのパッチ、パッチ・セット・アップデート (PSU) 、パッチ・セット、またはマイナー・リリースを使って、実行しているOracle WebLogic Serverクラスタをアップグレードするプロセスです。クラスタまたはドメイン全体を停止させずに済むため、中断のないサービスを実現できます。クラスタのローリング・アップグレード時に、クラスタ内の各サーバーは個別にアップグレードされて再起動され、その間、クラスタ内の他のサーバーがアプリケーションを引き続きホストします。このプロセスは手動でスクリプト化できます。また、パッチ適用では、Oracle WebLogic Server Management Pack Enterprise Editionを使ってこのプロセスを自動的に統合できます。ローリング・アップグレードは、サーバー保守などの実践的な要件に直面した場合でもOracle WebLogic Serverクラスタが中断のないサービスを提供する、別のメカニズムも提供します。

動的クラスタ

動的クラスタはOracle WebLogic Server 12.1.2で提供されている重要な新しいクラスタ構成オプションで、特に、クラウド・インフラストラクチャの要件や、予測できない可能性があるアプリケーション・パフォーマンス要件を対象としています。動的クラスタでは、クラスタの初期構成を簡素化する新しい構成オプションを使用でき、アプリケーション・パフォーマンス要件を満たすようにクラスタを拡張できます。以前のOracle WebLogic Serverリリースでサポートされていた標準の"構成済みクラスタ"は、この新しいクラスタ構成オプションとともに引き続きサポートされます。

Oracle WebLogic Server 12.1.2より前のバージョンの構成済みクラスタでは、管理者がクラスタ内の個々の管理対象サーバーに、管理対象サーバーの名前、リスニング・ポート、マシン・マッピングなどのサーバー固有の構成情報を指定する必要がありました。そのため、クラスタはサーバー構成に関して同種となることを目的としていますが、構成済みクラスタにはサーバー固有の設定を指定する必要がありました。これは、最初のクラスタ構成を作成する際だけでなく、クラスタ構成を拡張する際にも当てはまります。そのため、構成済みクラスタを拡張する作業は、単一のサーバーを構成する作業やクラスタ内のサーバーのN個のコピーをリクエストする作業よりも複雑になっていました。

動的クラスタによって、これが変化しました。動的クラスタを使用すると、ユーザーは、サーバー・テンプレートに定義されているサーバー構成に基づいてクラスタ構成を作成できます。このような定義されている構成を使用して、ユーザーはクラスタ内に作成する管理対象サーバーの数を指定します。Oracle WebLogic Server構成では、クラスタ内で必要な数の管理対象サーバーを作成するために、一意の管理対象サーバー名、マシン・マッピング、およびリスニング・ポートが自動的に生成されます。クラスタに管理対象サーバーを追加して構成を拡張する場合、ユーザーは、クラスタ内の管理対象サーバーの数という1つのパラメータを変更するだけで、追加の管理対象サーバーを構成内に作成して起動できるようになります。

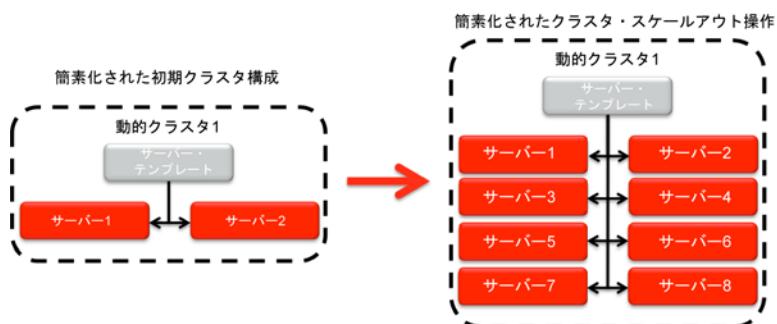


図6：動的クラスタ - クラウド環境の柔軟な拡張

動的クラスタ構成の作成と拡張は、管理コンソール、Oracle Fusion Middleware Control、WLSTなどの標準ツールを使って簡単に実行できます。動的クラスタでは、同じ基盤となるランタイム実装が使用され、標準の"構成済み"クラスタと同じサービス・レベルが提供されます。動的クラスタは新しいクラス構成オプションであり、Oracle WebLogic Serverクラスタの新しい実装ではありません。

動的クラスタはManaged Coherence ServerおよびElastic JMSと連携して動作するように設計されています。ただし、動的クラスタでManaged Coherence Server構成とElastic JMS構成を設定および拡張できます。ただし、動的クラスタのElastic JMSのサポートには、順序単位、作業単位、ストア・アンド・フォワード・サービス、サーバーとサービスの移行などの高度なJMS機能のサポートはまだ含まれていません。これらの高度なJMS機能が必要なユーザーを除き、オラクルでは、クラスタの初期構成の簡素化を動的クラスタ固有の拡張性と組み合わせることで、この機能がOracle WebLogic Serverユーザー、特に新しいクラウド・インフラストラクチャを作成するユーザーに広く使用されるようになると確信しています。

Oracle WebLogic ServerとOracle Coherenceの統合

Oracle Coherenceは、頻繁に使用されるデータに信頼性とスケーラビリティに優れた方法で高速にアクセスできるようにすることで、アプリケーションを計画どおりに拡張できるようにする、業界をリードするインメモリ・データ・グリッド・ソリューションです。Oracle Coherenceの直線的なスケーラビリティで重要なのは、データ・グリッド内のノード全体でデータをパーティション化することです。つまり、各ノードは、外部ユーザーに対し、一貫性のある単一のクラスタ・キャッシュとして表されるデータの完全なセットをキャッシュしません。Oracle CoherenceをOracle WebLogic Serverアプリケーションとともに使用すると、アプリケーションのパフォーマンス、スループット、およびスケーラビリティを向上できます。

Oracle Coherenceで提供されているAPIを使用すると、開発者は強力な方法でアプリケーションを作成したり、データ・グリッドのメリットを活用したりできます。Oracle WebLogic Suiteユーザーは、Oracle Coherence Enterprise Editionで提供されているすべてのAPIを使用できます。ただし、提供されているカスタム・アプリケーションの開発サポートに加えて、オラクルでは、アプリケーションのプログラミングなしにOracle Coherenceを利用する2つのメカニズムをOracle WebLogic Serverユーザーに用意しています。この2つのメカニズムは次のとおりです。

- Coherence*Web : Oracle Coherence内のHTTPセッション管理のサポート
- TopLink Grid : TopLinkを使用してOracle Coherence内にJPAエンティティを格納する、JPAアプリケーションのサポート

Coherence*Web

Coherence*Webは、アプリケーション・サーバー・インフラストラクチャからセッション管理をオフロードすることで、HTTPセッション・ステートを使用するJava EE Webアプリケーションのパフォーマンスを拡張および向上するように設計されています。アプリケーション・サーバーからセッション・ステート管理をオフロードするのが適切な、さまざまなシナリオがあります。多数のセッションを使用するアプリケーションでは、セッション・データがアプリケーション・サーバーのメモリ使用で競合する可能性がある場合や、セッション・データを保持するために作成された多数のヒープでガベージ・コレクションの待機時間などの他のトレードオフが生じる可能性がある場合に、アプリケーションのスケーラビリティをすばやく制限できます。このような場合、Coherence*Webを使用して、HTTPセッション・ステートのレプリケーションを含め、HTTPセッション・ステートをデータ・グリッドに格納できます。これにより、高可用性ソリューションでセッション・データにきわめて高速にアクセスできるとともに、グリッド用に設計された作業にデータ・グリッドを適切に利用できるようになります。Coherence*Webは、複数のアプリケーションでセッション・データを共有する必要がある場合も、適切に機能します。

TopLink Grid

JPA実装で提供される一般的なローカル・キャッシュ・インフラストラクチャを使用せずに、Oracle WebLogic Server内のTopLink Grid実装を使用すると、TopLinkベースのJPAアプリケーションで Oracle CoherenceをJPAエンティティの高パフォーマンス共有キャッシュとしてシームレスに利用できます。JPAプログラミング・モデルは、開発者がJPAアプリケーションをグリッドにすばやく簡単に配置できるようにする、数多くのアノテーションによって強化されています。その結果、JPAアプリケーションのパフォーマンスとスケーラビリティが大幅に向かっています。

上記のCoherence*Web機能とTopLink機能をManaged Coherence Serverと組み合わせて、WebLogic 管理フレームワークを使用すると、アプリケーション・サーバー・テクノロジーとデータ・グリッド・テクノロジーを業界で比類ない高いレベルで統合できます。この統合により、ユーザーは、Oracle WebLogic SuiteアプリケーションおよびOracle WebLogic Suiteの上に構築されたミッション・クリティカルなクラウド・インフラストラクチャで、Oracle Coherenceのスケーラビリティ、信頼性、およびパフォーマンスのメリットを簡単かつ効率的に利用できます。

JMSメッセージング

Oracle WebLogic Server JMSは、もっとも要求の厳しいメッセージング・アプリケーションに必要なすべての機能を提供するために設計および開発されています。JMS仕様に加えて、Oracle WebLogic Server JMSは、次のような高付加価値の信頼性に優れたメッセージング機能を組み込んでいます。

- 順序単位：正しくない順序でメッセージが処理されると、アプリケーションの整合性に深刻な影響が及ぶ可能性があります。この機能は、すべてのメッセージが、宛先に配置されている順序で連続して処理されることを保証します。
- 作業単位：順序単位機能で提供されるよりもさらに制限されたグループがアプリケーションに必要な場合、作業単位を使用して、メッセージ・コンシューマがメッセージのコレクションを1つの単位として処理できるよう、メッセージを1つの単位としてグループ化できます。作業単位内でスコープ設定されているすべてのメッセージが使用できるようになるまで、受信者は作業単位内のメッセージを処理できません。
- メッセージングの保存と転送：この構成により、フレームワークで信頼性の課題に対処することで、メッセージベースのアプリケーションの開発で複雑さが大幅に軽減されます。関連するメッセージ・プロバイダが停止している場合でもメッセージを1回のみ送信するオプションをメッセージ・クライアントに設定する必要があります。メッセージングの保存と転送機能は、送信先のシステムが使用できないときに、システムが使用可能になってメッセージが転送されるまで、システムに送信されるメッセージを保存します。
- Distributed Destination：Distributed Destinationは、単一の論理的な宛先である場合にクライアントからアクセスできる、一連の宛先（Distributed QueueまたはDistributed Topic）のことです。Distributed Destinationは通常、高可用性環境または一連のコンピュータ・リソースでメッセージ処理を負荷分散する必要がある環境で使用されます。

- JMSサーバーとサービスの移行：Oracle WebLogic Serverの移行とは、障害発生時に、クラスタ化された管理対象サーバー、またはクラスタ化された管理対象サーバーで実行されていたサブシステム・コンポーネントを別の場所に移動するプロセスのことです。このドキュメントの前の項で詳しく説明したように、サーバーとサービスの移行はJMS構成に高可用性を提供します。

Oracle WebLogic Server JMSも、Oracle Databaseに基づいて、Oracle AQ JMS実装のサポートを備えています。Oracle AQ JMSは、Oracle Databaseの可用性機能と信頼性機能を利用してエンタープライズ・クラスのメッセージング・サービスを提供します。Oracle AQ JMSでも、他のOracle AQ JMSアプリケーションとの相互運用が可能で、JTAトランザクションが完全にサポートされています。

Active GridLink for RAC

Oracle WebLogic Serverマルチ・データソースは、Oracle Database RACシステムでの使用が長年サポートされており、本番環境でのメリットが実証されているため、今後も引き続きサポートされる予定です。

ただし、Oracle WebLogic Server 11gより、Oracle WebLogic ServerとOracle Database RACの統合用の新しい機能が提供されており、この新しい機能はActive GridLink for RACと呼ばれています。

Oracle WebLogic Suiteで提供されているActive GridLink for RACは、Oracle Database RACシステムとの統合に推奨されているメカニズムです。Active GridLink for RACは、構成を簡素化し、Oracle Database RACの接続を最適化します。そのため、集中型のデータベース処理が伴うエンタープライズJavaアプリケーションでスループットが向上し、応答時間が短くなります。

Oracle Active GridLink for RACにより、GridLinkデータソースで次の機能を利用できます。

- ランタイム接続ロードバランシング – GridLinkデータソースは負荷を認識し、RACノードのワークロードに関して事前予防的に通知を受け取って、負荷のもっとも小さいノードに接続を分散できます。そのため、RAC環境の使用率がさらに向上し、アプリケーション・パフォーマンスが全体的に向上します。
- XAアフィニティ – グローバル・トランザクションのトランザクションベースのアフィニティにより、Oracle RACクラスタで実行されるグローバル・トランザクションのすべてのデータベース操作が同じOracle RACインスタンスに確実に送られます。XAトランザクションの最初の接続リクエストはランタイム接続ロードバランシングを使って負荷分散されて、アフィニティ・コンテキストが割り当てられます。後続の接続リクエストはすべて、最初の接続のアフィニティ・コンテキストを使って同じOracle RACインスタンスにルーティングされます。
- RACクラスタ全体の單一データソース - RACクラスタが大規模で、数多くのさまざまなドメインが含まれている場合、GridLinkデータソースによって構成を大幅に簡素化できます。
- 高速接続フェイルオーバー - RACクラスタのイベントベースの通知変更であり、データベース接続をスムーズにリカバリします。これにより、GridLinkデータソース内の接続プールに有効な接続が含まれるため、接続のポーリングやテストを行わずに済みます。
- Oracle Single Client Access Name (Oracle SCAN) のサポート - Oracle Database 12cのお客様からよく質問を受ける機能です。Oracle RACノードを追加または削除する場合に、SCANアドレスが含まれたGridLinkデータソースは変更が不要です。

Oracle WebLogic Serverのほとんどのお客様はOracle Databaseを使用しています。Active GridLink for RACなどの統合機能により、お客様では、オラクルが提供している製品間の統合を利用して、アプリケーションのパフォーマンス、可用性、および管理性を向上できます。

Database 12cの統合

オラクルでは、先日、業界をリードするデータベース・テクノロジーの最新バージョンであり、Oracle WebLogic Suiteアプリケーションとの組合せでもっともよく使用されるデータベースである、Oracle Database 12cをリリースしました。オラクルでは、アプリケーション・データをホストするデータベース・テクノロジーとの統合がきわめて重要であることを認識しており、エンタープライズ・データベース・テクノロジーで提供される機能を利用することで実現できるお客様の価値を理解しています。この理由から、オラクルは、Oracle DatabaseとOracle WebLogic Serverの統合を引き続き強化しています。

具体的には、これらの製品の上に構築されるクラウド・インフラストラクチャの可用性、パフォーマンス、およびマルチテナント機能を強化するため、Oracle WebLogic Server 12.1.2とOracle Database 12cを統合しました。Oracle WebLogic Suiteのほとんどのお客様は、最終的にOracle Database 12cを使用することになります。Oracle WebLogic Suiteへの投資は、新しいデータベース機能を次の領域で利用するインフラストラクチャへの投資です。

- アプリケーション継続性：アプリケーション環境でよく発生する問題は、データベース接続が切断されるというものです。接続の切断は、ネットワークの一時的な停止や、RACクラスタ内の複数のノードで発生したフェイルオーバーなど、一時的な問題が原因で発生する可能性があります。原因が一時的なものであり、接続を再確立できる場合でも、このような接続の切断のために例外が発生し、アプリケーションで処理する必要が生じたり、エンドユーザーに影響が及んだりするため、アプリケーションの信頼性が低下します。Oracle WebLogic Server 12cでは、アプリケーション継続性を透過的に利用して、GridLinkと汎用データソースの両方でこの問題に対処します。データベース接続が一時に切断された場合でも、エンドユーザーにアプリケーション・サービスを引き続き提供します。アプリケーション継続性によって、切断された接続が自動的に再作成され、読み取り処理と書き込み処理を含め、処理中のデータベース・リクエストが再実行されます。これらの処理はアプリケーションやエンドユーザーに対して透過的に実行され、開発者がプログラミングを行うことなく、エンドユーザーに信頼性の高いアプリケーション・サービスを提供できます。
- データベース常駐接続プーリング：アプリケーション・サーバーとデータベース間に専用の接続を維持すると、データベース・リソースが消費されます。多数のドメイン、クラスタ、およびサーバーが最終的に作成されてデータベース・インスタンスに接続されるクラウド環境では、接続管理に伴うリソース消費のために、データベース層で拡張の問題が発生する可能性があります。拡張の問題は、多くの接続が通常はアイドル状態になっている場合でも発生する可能性があります。

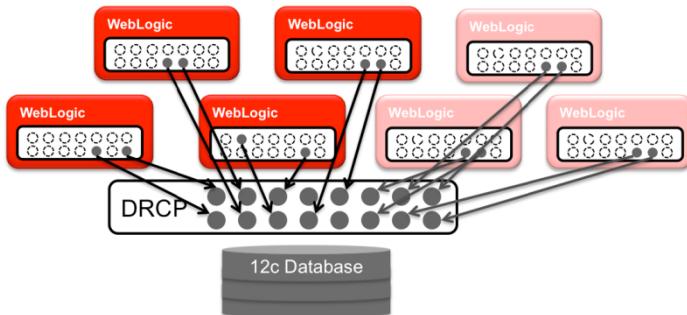


図7：データベース常駐接続プーリングによる接続共有とリソースの効率利用

データベース常駐接続プーリングは、複数のアプリケーション・インスタンスによって共有されているデータベース層に再利用可能な接続プールを作成することで、この問題に対処します。Oracle WebLogic Server 12.1.2は、データベース常駐接続プーリングを使用およびサポートするように統合されています。アプリケーション・サーバー接続は、クローズされると、共有接続プールに返されます。これにより、アプリケーション・サーバー・インスタンスが追加されるため、データベース層の拡張性が向上するとともに、新しいクラウド・サービスをクラウド環境にプロビジョニングするにつれて、クラウド・インフラストラクチャの拡張性が向上します。

- プラガブル・データベース：Oracle Database 12cでは、特定のテナント専用にできる単一の仮想データベースを表す、プラガブル・データベースの概念が導入されています。マルチテナント・コンテナ・データベースでは、複数のプラガブル・データベースをホストできるため、専用のプラガブル・データベースを使ってテナントのデータ分離を維持しながら、クラウド環境でデータベース統合を実現したり、データベース層で管理効率を大幅に向上させたりできます。Oracle WebLogic Server 12.1.2では、WebLogic Serverアプリケーションでプラガブル・データベースの使用をサポートしているだけでなく、プラガブル・データベース機能を利用してマルチテナント・アプリケーションもサポートしています。Oracle WebLogic Server構成内の単一のデータソースを使用して、マルチテナント・アプリケーションでは、テナントのIDに基づいてテナント固有のプラガブル・データベースを動的に切替えできるため、単一のマルチテナント・アプリケーションによってアクセスされているテナント・データを分離できます。データベース層でプラガブル・データベースの効率性を利用したいと考えている組織や、クラウド・アプリケーションでマルチテナント機能を利用したいと考えている組織では、Oracle WebLogic Suite 12.1.2を使用すると、組織の要件を満たすように統合を行えます。

- グローバル・データ・サービス：グローバル・データ・サービスは、マルチテナント環境でデータベース・サービスをグローバルに一元管理するための、Oracle Database 12cの機能です。サイトのワークフローと可用性に基づいて、アプリケーションによって作成およびアクセスされるサービスをサイト全体で移行できます。移行シナリオとフェイルオーバー・シナリオをサポートするためにサイト全体でデータをレプリケートするには、Oracle Data GuardまたはOracle Golden Gateを使用します。さらに多くのサービスを追加するにつれ、グローバル・データ・サービス環境にRACクラスタを追加し、アプリケーションが利用できる可用性に優れた分散データベース・クラウドを構築すると、スケーラビリティを実現できます。Oracle WebLogic Server 12.1.2は、Active GridLink for RACを使ったOracle Database 12cグローバル・データ・サービスへの直接統合を提供します。GridLinkデータソース構成では、サービスにアクセスするプライマリとセカンダリの領域を指定します。サービスが移行されると、Oracle RAC Fast Application Notification (FAN) のイベントがGridLinkデータソースによって認識されて、移行されたサービスへの接続が自動的に再確立されます。この自動移行機能によって、Oracle WebLogic Suiteユーザーはグローバル・データ・サービスを十分に活用できます。

Oracle Database 12cは、強力な新機能を数多く備えており、オラクルでは、この最新バージョンのデータベースが企業のお客様に広く導入されることを期待しています。Oracle WebLogic Suite 12.1.2は、クラウド・インフラストラクチャの可用性、マルチテナント、およびスケーラビリティを強化する、付加価値の高い重要な統合機能を提供します。

Exalogic Elastic Cloudソフトウェアの最適化

Oracle Exalogic Elastic Cloudは、比類のないスケーラビリティ、パフォーマンス、および管理性を実現する、セキュアでミッション・クリティカルなプライベート・クラウドの基盤を企業に提供するために設計されています。Oracle Exalogic Elastic Cloudソフトウェアには、オラクルの次世代アプリケーションの開発に使用した、不可欠なJava基盤であるOracle WebLogic Suite内のコア製品で行われた数多くの最適化と機能強化が含まれています。

Oracle Exalogic Elastic Cloudソフトウェアには、Oracle WebLogic Suiteで行われた一連の機能強化が含まれているため、Exalogicハードウェアで実行すると最適なパフォーマンスを実現できます。ソフトウェアの最適化により、ソフトウェアをExalogicの高密度コンピューティング・ノードと高速ネットワークInfiniBandスイッチで実行するときに関連する、パフォーマンスの制限に対処しています。これらのソフトウェア最適化では、このハイエンド・ハードウェアのメリットを最大限に活用することで、ハードウェアとソフトウェアのバランスが取れたエンジニアド・システムを実現しています。

これらの最適化の例には、次のようなものがあります。

- スケーラビリティ、スループット、応答性の向上：Oracle WebLogic Serverのネットワーキング、リクエスト処理、およびスレッド管理のメカニズムが改善されているため、内部InfiniBandネットワーク・ファブリック経由で接続されているExalogicコンピュータ・ノードのスケーラビリティとパフォーマンスを向上できます。
- セッション・レプリケーションの卓越したパフォーマンス：ネットワーク経由でのパラレル接続に管理対象サーバー間で使用される広いInfiniBand帯域幅を利用するように、Oracle WebLogic Serverのセッション・レプリケーション・メカニズムが改善されています。

- Active GridLink for RAC：データベース・インターラクションの信頼性を高めるよう、Oracle WebLogic ServerとOracle Database RACの統合が最適化されています。
- Oracle Coherenceのメッセージ・バス：Oracle CoherenceとInfinibandメッセージ・バスの統合により、非常に高いパフォーマンスを提供し、Oracle Coherenceインスタンス全体でRemote Direct Memory Access (RDMA) を使った待機時間の短いメッセージングを実現しています。
- ExalogicとExadata間の通信の応答時間の短縮：ExalogicシステムをExadataシステムに直接接続している場合に、InfiniBandを用い、SDPと呼ばれる"ネイティブ"なInfiniBandネットワーク・プロトコルを使用して、Exadata上のOracle RACデータベースとやり取りを行います。このため、データベースの呼出しで、リクエストからレスポンスまでの待機時間が短くなります。データベースから多くの結果セットを転送する場合に、パフォーマンスがもっとも顕著に向上します。実際の効果として、アプリケーションがクライアント・リクエストにさらに高速に応答できるため、エンタープライズJavaアプリケーションでパフォーマンスが全体的に向上します。
- Oracle WebLogic Suite 12.1.2の新機能は、ExalogicのJMSサブシステムの最適化機能です。社内パフォーマンス・ベンチマークで、ExalogicでJMS集中型アプリケーションを実行する場合に、CPU処理能力がExalogicストレージ・サブシステムのI/O能力を上回ることが確認されていました。そのため、ディスクに保持されるJMSメッセージでシステムにボトルネックが生じていました。Oracle WebLogic Server 12.1.2はExalogicシステムの最適化機能を備えており、JMSメッセージはディスクに保持される前に圧縮されるため、ストレージ処理のボトルネックが軽減され、JMS集中型アプリケーションのパフォーマンスが最大3倍向上します。

ハードウェアとソフトウェアを組み合わせることで、Oracle WebLogic Suite 12cで実行されているJavaベース・アプリケーションのパフォーマンスが大幅に向上し、Exalogicはクラウド・アプリケーションでミッション・クリティカルなサービスを実行するのに理想的なプラットフォームとなっています。

Oracle WebLogic Suite 12.1.2へのアップグレード

Oracle WebLogic Serverユーザーは、Oracle WebLogic Serverの以前のリリースからOracle WebLogic Server 12.1.2にアプリケーションをアップグレードする際に、互換性のレベルが非常に高いことを実感することでしょう。アプリケーション互換性、ドメイン互換性、およびバージョン間の相互運用性を提供することで、スムーズなアップグレード・パスを確保しています。アップグレードの計画時には、製品ドキュメントを参照してください。

Oracle WebLogic ServerへのOracle Containers for Javaアプリケーションの移行をサポートするOracle SmartUpgradeが、Oracle Containers for JavaアプリケーションからOracle WebLogic Server 12.1.2への直接アップグレードをサポートするように更新されています。

まとめ

Oracle WebLogic Suite 12.1.2は、Oracle WebLogic Server製品ラインを代表するエディションの最新リリースです。Oracle WebLogic Server、Oracle Coherence Enterprise Edition、Oracle Web Tier、Oracle TopLink、Oracle Java Mission Control、Flight Recorder、およびその他のコンポーネント・テクノロジーの最新バージョンが含まれています。最新の開発プラットフォーム、ネイティブなクラウド管理、およびミッション・クリティカルなクラウド・プラットフォームを提供するため、Oracle WebLogic Suite 12.1.2は以前のリリースの機能に基づいて構築されており、クラウド・インフラストラクチャを計画しているお客様に理想的なアプリケーション・プラットフォームとなっています。お客様による製品の利用とフィードバックをお待ちしています。



Oracle WebLogic Suiteテクニカル・
ホワイト・ペーパー

2013年7月

著者：Will Lyons

共著者：クラウド・アプリケーション
製品管理とマーケティング・チーム

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



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2013, 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、ここに記載される内容は予告なく変更されることがあります。本文書は一切間違いないことを保証するものではなく、さらに、口述による明示または法律による暗示を問わず、特定の目的に対する商品性もしくは適合性についての默示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社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。本文書はオラクル社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は米国Oracle Corporation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。

Hardware and Software, Engineered to Work Together