

データコネクティビティに関する 知識の重要性

WHITE PAPER

The logo for DataDirect Technologies features the company name in a blue sans-serif font. The word "Data" is in a lighter blue, and "Direct" is in a darker blue. A series of small grey dots forms a curved path above the "Direct" portion. Below the main name, the word "TECHNOLOGIES" is written in a smaller, all-caps, grey sans-serif font.

DataDirectTM
TECHNOLOGIES

要約

データコネクティビティは、アプリケーションによるデータベースへのアクセスを可能にするミドルウェアで、IT インフラストラクチャを構成する非常に重要なコンポーネントの1つですが、見過ごされることも少なくありません。ODBC ドライバ、JDBC ドライバ、ADO.NET プロバイダなどのデータベースコネクティビティコンポーネントは、アプリケーションのパフォーマンス、信頼性、および移植性に大きな影響を及ぼす可能性があります。コンポーネントの選択の誤りは、開発コストの増大、導入の遅延、収益機会の喪失を通じて最終的な収益に直接影響します。実際、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に対する最も大きな障害の1つは、欠陥のある(または最適ではない)データコネクティビティに起因する、アプリケーションとデータベース間のボトルネックです。

このホワイトペーパーでは、データコネクティビティの概要とデータコネクティビティがアプリケーションに及ぼす影響の詳細について説明するとともに、選択可能なさまざまなソリューションについて検討します。

目次

重要ポイント	2
はじめに	3
データコネクティビティについて	3
データコネクティビティとは.....	3
データコネクティビティの重要性	4
アプリケーション開発への影響	4
ビジネス上の検討事項	7
選択可能なデータコネクティビティ	8
ベンダが提供するドライバ/プロバイダ	8
ネイティブコネクティビティ.....	13
オープンソースのドライバ/プロバイダ	13
サードパーティのデータコネクティビティの評価	15
包括性	15
実績	15
技術的リーダーシップ	16
結論	17
Appendix A – Database and Platform Support by Product Line	1

重要ポイント

- ODBC ドライバ、JDBC ドライバ、ADO.NET プロバイダなどのデータベースコネクティビティコンポーネントは、アプリケーションをデータベースに接続するミドルウェアです。データベースとの通信はすべて、データコネクティビティコンポーネント経由で行われます。
- データコネクティビティの品質は、機能、パフォーマンス、スケーラビリティ、セキュリティ、信頼性など、アプリケーションの主な特性に大きな影響を及ぼします。
- データコネクティビティの選択は、開発、テスト、および顧客サポートのコストに大きな影響を及ぼします。
- 複数のデータベースや複数のデータベースバージョンに接続する必要があるアプリケーション開発者は、データベースに依存しないデータベースコネクティビティコンポーネントを選択する必要があります。
- アプリケーション開発者は、データコネクティビティを評価する際に、ベンダが提供するドライバ、ネイティブ/独自仕様のコネクティビティ、オープンソースドライバ、およびサードパーティが商用目的で開発したコンポーネントの4つの中から選択できます。
- データベースベンダから提供されるソリューションは、相互運用性と移植性を制限するように設計されているため、本質的に劣っています。データベースベンダは、コンポーネントの研究開発またはテストに多くのリソースを投入することはありません。そのため、データベースベンダが提供するコンポーネントには、品質、パフォーマンス、および機能のサポートの点でも制限があります。
- サードパーティコンポーネントは、重要なアプリケーションおよび複数のデータベースをサポートするアプリケーションにとって最良の選択肢です。
- サードパーティコンポーネントを選択する場合、アプリケーション開発者は以下に示す3つの主要な要因について検討する必要があります。
 - 包括性 – 製品がサポートする環境と機能の幅広さはどの程度か
 - 経験 – ベンダおよびその製品がどの程度の実績があるか
 - 技術的リーダーシップ – ベンダはこの分野におけるリーダーであるか

はじめに

現在、大部分のソフトウェアアプリケーションは、情報の格納、アクセス、および管理をサポートするために、リレーショナルデータベースに依存しています。顧客、注文、ユーザ、在庫、および財務に関する情報を格納するデータベースは、ERP、CRM、ファイナンシャル/ビジネスインテリジェンスなど、データベースに接続するエンタープライズシステムと同等の重要性を持つようになっています。

しかし、アプリケーションとデータベース間の接続についてはどうでしょうか。

ますます複雑化するアプリケーションを数多くのデータベースバックエンドに対して配置する必要性が増大する中で、多くのアプリケーション開発者は、データ接続に必要な不可欠なデータコネクティビティの重要性を見過ごしています。

このホワイトペーパーでは、アプリケーションの全体的なパフォーマンス、スケーラビリティ、相互運用性、信頼性、開発およびメンテナンスのコストと複雑さという点において、データコネクティビティが果たす役割について説明します。次に、選択可能なさまざまなデータコネクティビティについて検討し、これらのデータコネクティビティを評価する際の注意事項を提示します。

データコネクティビティについて

データコネクティビティとは

データコネクティビティには、ソフトウェアアプリケーションをデータベースに接続する際に使用されるすべての方法が組み込まれています。データコネクティビティと最も頻繁に関連付けられるのは、ODBC ドライバ、JDBC ドライバ、Microsoft .NET Framework の ADO.NET データプロバイダなどの標準ベースのコンポーネントです。

データコネクティビティコンポーネントには、ソフトウェアアプリケーションによるデータベース機能の呼び出しおよび使用を可能にするデータベースの内部動作に関する特定の知識が組み込まれています。データコネクティビティコンポーネントは、IT インフラストラクチャにおいて以下の5つの重要な機能を実行します。

1. データベースにアクセスするための標準手段(API)をアプリケーションに提供します。
2. アプリケーションがデータを更新、挿入、または削除したり、データベース内のストアプロシージャを呼び出ししたりできるようにします。
3. アプリケーションがデータベースエンジンに格納されている機能やビジネスロジックを呼び出せるようにします。

4. アプリケーションがデータベースからデータを取得できるようにします。
5. データのETL(抽出、変換、および格納)のためのデータベース間通信を提供します。

データコネクティビティの重要性

データコネクティビティコンポーネントは、ソフトウェアアプリケーションとデータベース間を流れるすべての情報およびコマンドの通り道になります。データコネクティビティアーキテクチャの精巧さと品質は、IT インフラストラクチャのパフォーマンス、スケーラビリティ、信頼性、およびセキュリティに必ず影響します。

データコネクティビティの品質とパフォーマンスは、アーキテクチャの精巧さと基になるデータベースエンジンに関する深い知識に依存します。たとえば、データベースベンダのクライアントライブラリ(アプリケーションと同じサーバにインストールする必要がある追加ソフトウェア)を使用すると、ODBC ドライバの動作が遅くなることがよくあります。クライアントライブラリでは、処理に追加の手順が必要になるため、パフォーマンスとスケーラビリティが低下します。クライアントを使用しない(ワイヤプロトコルの)ODBC ドライバでは、この手順が必要ないため、パフォーマンスとスケーラビリティが劇的に向上します。同様に、基になるデータベースプロトコルに関する深い知識なしに開発されたコンポーネントでは、データベースの機能をアプリケーションで十分に活用することができません。

データコネクティビティコンポーネントは、ソフトウェアの保守性および移植性に影響する重要な要素でもあります。すべてのデータベース通信はコネクティビティコンポーネントを経由する必要があるため、コンポーネントのアーキテクチャに欠陥またはバグが含まれていると、アプリケーションの所有者がサポートを必要とする問題が発生します。たとえば、JDBC ドライバが JDBC 仕様で定められた特定の機能を正しく実装していない場合、アプリケーションが予想外の動作をすることがあります。移植性に関しては、データコネクティビティのアーキテクチャによって、データベース間、データベースバージョン間、およびプラットフォーム間の移植が簡素化されることもあれば、劇的に複雑化されることもあります。理想的には、データコネクティビティコンポーネントは、基になるデータベースインフラストラクチャの変更またはアップグレードを容易にする共通のアーキテクチャを共有すべきです。

アプリケーション開発への影響

アプリケーション開発がデータコネクティビティの選択から最も大きなメリットを得ることはほぼ間違いありません。戦略的なデータコネクティビティにより、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の向上、データベースとプラットフォームの移植性の劇的な改善、プロジェクトコストの削減、開発者の生産性の向上、機能の増加、および配置に要する時間の短縮が実現されます。

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

新しいアプリケーションを設計したり、既存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の向上を図る場合、エンジニアおよびアプリケーション開発マネージャは、データコネクティビティについて慎重に検討する必要があります。この数年間でサーバの処理能力が大幅に向上するとともに、サーバの価格が急激に下落しました。その結果、アプリケーションおよびデータベースがハードウェアの処理能力によって制限されなくなりました。つまり、ボトルネックがシステムの他の部分に移動したのです。

実際、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に対する最も大きな障害の1つは、欠陥のある(または最適ではない)データコネクティビティに起因する、アプリケーションと基になるデータベース間のボトルネックです。

1. クライアントライブラリの使用：クライアントライブラリ(クライアントシステムにインストールする必要がある追加のソフトウェア)が必要な場合、アーキテクチャが複雑になり、パフォーマンスとスケーラビリティが低下します。
2. ディスクキャッシュの使用：データコネクティビティコンポーネントがメモリ内処理ではなくディスクキャッシュを使用する場合、パフォーマンスが大幅に低下する可能性があります。
3. 冗長なデータベース通信：データコネクティビティコンポーネントがデータベースと通信する際に必要以上にネットワークを多く使用する場合、パフォーマンスが低下します。
4. Java/JDBC 環境における接続プーリングやステートメントのプーリングのサポート – データコネクティビティコンポーネントが接続プーリングまたはステートメントのプーリングをサポートしない場合、パフォーマンスが大幅に低下する可能性があります。

データベースとプラットフォームの移植性

データコネクティビティのアーキテクチャによって、データベース間、データベースのバージョン間、およびプラットフォーム間の移植が簡素化されることもあれば、劇的に複雑化されることもあります。理想的には、データコネクティビティコンポーネントは、基になるデータベースインフラストラクチャの変更またはアップグレードを容易にする共通のアーキテクチャを共有すべきです。ソフトウェア企業およびエンタープライズ IT 組織の大部分は、複数のデータベースプラットフォームをサポートする必要があります。また、サポートする各プラットフォームの複数のバージョンもサポートする必要があります。これは、無数に存在するデータベースコネクティビティ手法、ドライババージョン、およびクライアントライブラリパッケージを管理するという意味です。新しいデータベースを追加したり、同じデータベースを新しいバージョンにアップグレードする際には、開発、統合、およびテスト作業にかなりの時間を割かなければならない場合があります。

たとえば、特定のデータベースのみと連携するように設計されたデータコネクティビティコンポーネントは、他のデータベースのみと連携するように設計されたコンポーネントとは異なる方法で BLOB/CLOB データ型(大きなバイナリオブジェクトまたはキャラクタオブジェクト)を処理します。開発者

は、サポートする必要がある新しい各データベースに対する追加のコーディングおよびテストにかなりの時間と労力を費やします。

データコネクティビティアーキテクチャを標準化および簡素化することにより、複数のデータベースバックエンドのサポートに関連するコストと複雑さが大幅に軽減されます。これは、特に独立系のソフトウェアベンダにとって優先順位の高い作業です。

プロジェクトのコスト

データコネクティビティコンポーネントの開発、統合、およびテストに費やされる時間と労力は、プロジェクトのコストに大きく影響する可能性があります。これらのコストは、開発時間やコンサルティング料金、テスト、デバッグ、および修正のコストなど、さまざまな形で顕在化する場合があります。

たとえば、JDBC ドライバがセーブポイントをサポートしていない場合に、この機能をサポートする場合、開発者は追加のコードを記述してテストする必要があります。これにより、不要な時間とコストがプロジェクトに加算されます。

さらに、選択したデータコネクティビティの信頼性とパフォーマンスに応じて、顧客サポートおよびメンテナンスのコストも増加します。

アプリケーションの信頼性

データコネクティビティのバグは、アプリケーションのビジネスロジックのバグやデータベース層のバグと同様に、アプリケーションの信頼性に悪影響を及ぼす可能性があります。ユーザは、データベースクエリから一貫性のある正確な結果を得る際、および更新、挿入、削除から信頼できる結果を得るために、アプリケーションに依存しています。

信頼性を確保するには、アプリケーションインフラストラクチャのすべての層(アプリケーション、データコネクティビティ層、およびデータベース)が完全に連携して動作する必要があります。たとえば、ODBC ドライバや JDBC ドライバのバグは、データベースが誤った結果やエラーメッセージを返したり、結果をまったく返さない原因となる場合があります。

また、一部のデータコネクティビティコンポーネントは、アプリケーションの信頼性を大幅に向上させる機能(分散トランザクションのサポートなど)を備えています。信頼性が非常に重要となる Java 環境では、開発者は JTA(Java Transaction API)をサポートするデータコネクティビティコンポーネントを選択する必要があります。

開発者の生産性

データコネクティビティの構築、統合、およびテストに要する時間が多ければ多いほど、新しい機能の開発や組織内のその他のプロジェクトに割くことができる時間が少なくなります。アプリケーション開発組織では、開発スタッフおよび QA スタッフの最適利用が求められています。そのため、マネージャは、データコネクティビティの選択がリソースの割り当てに及ぼす影響を慎重に見極める必要があります。

機能

データベースベンダは絶えず革新を続けており、自社のデータベース製品に新しい機能を追加しています。また、データコネクティビティの標準である ODBC、JDBC、および ADO.NET には、新しい機能が頻繁に追加されています。しかし、開発者は、いずれかのカテゴリの新しい機能をアプリケーションで自動的に活用することができません。開発者が使用するデータベースドライバまたはプロバイダは、新しいデータベース機能や新しいバージョンの標準をサポートする必要があります。

古いバージョンのドライバまたは ADO.NET プロバイダは、当然ながら新しい機能をサポートしていません。また、新しいドライバは、古いデータベースに対する下位互換性を備えていない場合があります。データベース機能の最新の変更と最新の規格を活用したいと考えるアプリケーション開発者は、使用するデータコネクティビティ手法に必要なサポートが含まれていることを確認する必要があります。同時に、アプリケーションが以前のデータベースバージョンとの完全な下位互換性があることも確認する必要があります。

展開に要する時間

構成に関する問題が発生すると、展開に要する時間が大幅に増加する可能性があります。利用可能なデータベースおよびデータベースバージョンが無数に存在することを考慮すれば、データコネクティビティの構成が複雑で時間のかかる作業になることも不思議ではありません。

エンジニアおよびアプリケーション開発マネージャは、配置を円滑に行えるようにするために、データコネクティビティの合理化と標準化について検討する必要があります。これにより、構成とテストが簡素化され、アプリケーションの配置に要する時間が大幅に短縮されます。

ビジネス上の検討事項

データコネクティビティの選択は明らかに技術的な意思決定ですが、その実装およびアーキテクチャには、コスト、製品の収益見込み、プロジェクトの実施、および顧客サポート/顧客満足に関するビジネス上の重要な意味合いが含まれています。

アプリケーションの開発コスト

データコネクティビティコンポーネントの品質、使い勝手、および相互運用性は、特定のソフトウェア開発プロジェクトで必要となるエンジニアリングリソースに大きく影響します。統合およびテストにかなりの開発作業を要するコンポーネントを選択すると、新しい製品またはプロジェクトの総開発コストが増加することがあります。

製品の収益見込み

データコネクティビティは、データベース間およびデータベースバージョン間の相互運用性を決定する主な要因です。独立系ソフトウェアベンダの場合、できる限り多くのデータベースとバージョンをサポートすることが非常に重要になります。なぜなら、これを行うことにより、参入可能な市場と関連する製品の収益見込みを最大化できるからです。独立系ソフトウェアベンダは、データベースに依存しないデータコネクティビティコンポーネントを使用す

ることで、単一のコードベースを維持したまま、より多くの顧客を獲得することができます。

プロジェクトの実施スケジュール

開発者がデータコネクティビティの統合およびテストにより多くの時間をかければかけるほど、特定のプロジェクトの完了までに要する時間が長くなります。同様に、データコネクティビティに集中的に取り組んでいるエンジニアは、それと並行して新しい機能の追加または統合に取り組むことはできません。どちらの場合もアプリケーションとユーザに悪影響が及びます。

顧客サポート/顧客満足

データコネクティビティコンポーネントが正しく動作しない場合、ユーザは顧客サポートに連絡します。これにより、問題の診断および解決のための時間と作業が必要になります。これは主に独立系ベンダに当てはまりますが、カスタムアプリケーションをサポートする必要があるエンタープライズIT部門にも当てはまります。データベースベンダのドライバやプロバイダをベースとするアプリケーションは、コネクティビティコンポーネントではなくデータベース自体に重点を置くサポート/開発組織の意向に左右されます。

選択可能なデータコネクティビティ

これまでの説明により、データコネクティビティの選択がアプリケーション開発プロジェクトのコスト、品質、および移植性に大きく影響するということが明らかになりました。開発者が合理的な選択を行えるようにするために、利用可能なデータコネクティビティコンポーネントを大きく以下の4つのカテゴリに分類します。

ベンダが提供するドライバ/プロバイダ – データベースエンジンに付属するドライバまたはプロバイダです。

ネイティブのコネクティビティ – 業界標準のコネクティビティなしで開発されたクライアントライブラリベースの独自仕様のアプリケーションです。

オープンソースドライバ/プロバイダ – オープンソースプロジェクトを通じて開発および配布されるコンポーネントです。

サードパーティコンポーネント – データコネクティビティを事業とする企業によって開発および販売されるドライバとプロバイダです。

ベンダが提供するドライバ/プロバイダ

すべてのリレーショナルデータベースには、開発者がアプリケーションをデータベースに接続できるようにするためのドライバが付属しています。これらのコンポーネントは、特定ベンダのデータベースエンジンの特定のバージョン向けに設計されています。ベンダが提供するコンポーネントは、データベースエンジンを使用するのに必要な必須レベルの機能とパフォーマンスを提供します。

ベンダが提供するコンポーネントはデータベースに付属しているため、新しいアプリケーションを開発する上で最も低コストであると見なされることがよくあります。ただし、これらのコンポーネントには、プロジェクトの成功や開発およびサポートのコストに大きな影響を及ぼす可能性があるいくつかの重要な制限が含まれています。

ベンダのデータベースドライバは、データベースベンダによって月に一度の頻度で更新されます。バージョン間で一貫性のない機能がこれらのリリースに含まれていることも珍しくありません。独立系ソフトウェアベンダの場合、これによって製品リリース前の開発および QA スケジュールに大きな混乱が生じることがあります。これは開発の工数とコストが増大することを意味します。また、製品リリース後の技術サポートコストも増加します。エンタープライズ環境もほぼ同様です。頻繁に更新されるドライババージョンが複数存在する場合、開発、テスト、およびサポートのコストが増大します。

これらの制限の根本的要因は、データベースベンダは事業としてデータベースエンジンを開発および販売しているということです。データベースベンダがドライバを配布するのは、データベースに接続するための手段を開発者に提供する必要があるからです。

開発者は、この点を考慮に入れた上で、以下に示すベンダドライバの4つの特性を精査する必要があります。

- アプリケーションの移植性
- パフォーマンスとスケーラビリティ
- ソフトウェアの品質
- 機能のサポート

アプリケーションの移植性

ベンダが提供するコンポーネントは、特定のデータベースの特定のバージョンと連携するように設計されています。たとえば、Oracle では自社のデータベースエンジンの現行バージョンすべてに対応するために、その ODBC ドライバに数十種類のバージョンを用意しています。Oracle が提供するドライバに対して開発されたアプリケーションは、IBM DB2 データベースと連携出来ないだけでなく、Oracle の複数のデータベースとも連携出来ない場合があります。アプリケーションは、多くの場合、各種データベースおよびデータベースバージョンと連携させる必要があるため、アプリケーション開発者、テスト実施者、サポート担当者にとってこれは困った問題です。

アプリケーションが複数のデータベースの複数のバージョンをサポートする必要がある場合は、順列と組み合わせの数だけ管理およびバージョン問題が発生します。ソフトウェア開発者は、サポートするデータベースごとに異なるバージョンのアプリケーションを構築して管理することが必要になります。このため、追加テストや再テストの作業負担が増え、プロジェクトの完了が遅れます。開発やテストの先の段階でも、アプリケーションユーザが新しいバージョンのデータベースにアップグレードする際のサポートに関する問題が発生します。

さらに、データベースベンダはデータベースライセンスの販売を増やすことを目標としているため、顧客が他のベンダのデータベースエンジンに切り替えるのを阻止したいという動機が強く働きます。要するに、"ロックイン"を

作り出したいわけです。このために、データベースベンダは、データコネクティビティ標準に独自の拡張を実装する場合があります。このことが、アプリケーションの移植性をより一層複雑にしています。また、データベースベンダは、顧客が最新バージョンのデータベースにアップグレードしたくなるような戦略をとります。つまり、最新のコネクティビティコンポーネントを現在の主要なバージョンのデータベースにのみ対応できるようにして、顧客が新しいコンポーネントを利用する場合はデータベースのアップグレードを要求します。

相互運用性に不利になるような強い動機が働く場合もあります。たとえば、Microsoft SQL Server には JDBC ドライバが装備されています。しかし、Microsoft では、顧客が .NET ベースのアプリケーションを利用することを望んでいるため、Java アプリケーションのパフォーマンスと品質を向上させるための戦略を練ることはありません。自社の製品に高性能な JDBC ドライバを装備してもメリットがありません。同様に、Microsoft は Oracle 向けの .NET プロバイダを提供しています。しかし、顧客が自社の SQL Server のデータベース製品を採用することを望んでいるため、Oracle アプリケーションに対してパフォーマンスを最大化してもメリットがありません。

パフォーマンス/スケーラビリティ

データベースベンダにとって、データコネクティビティは、収益に影響を与えないため戦略的技術ではありません。データベースドライバを実装するのは、データベースに必要だからです。このように、データベースエンジン自体のパフォーマンスとスケーラビリティを向上させることには注力しますが、そのドライバやプロバイダの向上に取り組むことはあまりありません。

Oracle は自社のデータベースに ADO.NET プロバイダを装備して、Oracle をサポートする Microsoft の事業所が Oracle のコンポーネントを使用して Oracle にアクセスできるようにしています。しかし、Oracle は Java への取り組みに強い意欲を示し、顧客に対しては .NET で Java を使用することを勧めています。このように、ADO.NET プロバイダのパフォーマンスを向上させるための企業努力は最小限に抑えられています。一般的な高負荷環境の規模では言うまでもありません。高スケーラビリティ、高パフォーマンスのアプリケーション開発に従事する顧客は、特定のプラットフォームまたはデータベースの採用に利害関係のない、ベンダに中立なサードパーティのプロバイダを利用する必要があります。

ソフトウェアの品質

当然のことながら、データベースベンダは、テストや QA 作業の大半を自社のデータベースエンジンに対して投資します。データコネクティビティコンポーネントは、特定のデータベースバージョン内の特定の機能に対してのみテストされるのが一般的です。通常、これらのテストで何十万もの想定されるユースシナリオが網羅されることはありません。また、データベースベンダは、自社のドライバがコネクティビティ仕様(ODBC、JDBC など)に完全に準拠していることを確認するためのテストも行いません。

その結果、品質が低下し、バグが頻繁に発生します。低品質のためにデータベースが予想外の結果を返したり、単純に特定の機能が動作しなかったりする場合があります。データベースベンダがコネクティビティに費やすエンジ

ニアリングリソースはほんのわずかであるため、バグがすぐに修正されない場合があります。このことはプロジェクトの完了を遅らせる原因になります。

機能のサポート

データベースベンダは基になるデータベースについて深い知識があるため、最も豊富な機能を備えたデータコネクティビティコンポーネントを提供してくれるだろうと考えがちです。しかし、多くの場合、これは事実ではありません。

データコネクティビティ機能については、開発者は次のように大きく2つに分類して考える必要があります。

- データベースエンジン内の機能
- 標準(ODBC、JDBC など)でサポートされる機能

データベース内の機能については、大半のデータベースベンダが主力を注いでいる部分です。データベースに機能を追加することは、顧客にバージョンをアップグレードさせるための戦略です。しかし、多くのデータベースベンダは、他のデータベースに対して動作しないかもしれないネイティブな構文でこれらの機能を実装します。データベースベンダは、しばしば標準のデータアクセス仕様の範囲外である拡張"フック"をデータベースドライバに含めます。これにより相互運用性は著しく制限され、アプリケーションを再コーディングしない限り、他のデータベースで同様の機能を呼び出すことができなくなります。たとえば、ベンダによってストアドプロシージャを呼び出すために異なる方法が要求されたり、パラメータマーカの指定に異なる方法が利用されたりします。

一方で、データベースベンダは、最新バージョンのコネクティビティのスタンダードをサポートしていないのが通例です。実際には、データベースベンダは顧客のロックインを強化する独自の戦略を推し進めるため、通常、新しい標準を完全に採用するまでには時間がかかります。

機能がサポートされないことのわかりやすい例として、JDBC 3.0 に準拠していない Oracle の JDBC ドライバが挙げられます。つまり、このドライバはトランザクションセーブポイント、prepared statement pooling、BLOB/CLOB データの更新、Multiple Open resultSets など、重要な機能をサポートしていません。Oracle JDBC ドライバは J2EE JDBC 認定テスト (CTS) に合格していないため、J2EE と互換性がありません。したがって、Java コミュニティプロセスで要求される品質レベルを満たしておらず、アプリケーションは低品質と障害の高いリスクにさらされます。ベンダが提供する JDBC ドライバではサポートされない場合がある機能を次の表に示します。

アプリケーションで使用する機能	影響を受ける部分	制限または必要な追加作業
接続プーリング	パフォーマンス	接続プーリングは、一部のデータベースベンダのドラ

		<p>イバではまだサポートされていない JDBC 2.0 の機能です。パフォーマンスにとって重要な要素です。J2EE アプリケーションサーバ外でアプリケーションを実行している場合は、独自のプールマネージャを記述する必要があります。</p>
<p>ステートメントのプーリング</p>	<p>パフォーマンス</p>	<p>ステートメントのプーリングは、パフォーマンスにとって重要な要素です。アプリケーションサーバの中には、ステートメントのプーリングをサポートしていないものもあります(SunOne など)。アプリケーションをアプリケーションサーバ外で実行する場合、開発者は実行するステートメントごとに独自のステートメントのプーリングコードを記述する必要があります。</p>
<p>分散トランザクションサポート用の JTA</p>	<p>信頼性</p>	<p>分散トランザクションは、特に複数のデータベースにアクセスするトランザクションのように、信頼性が重要な場合に不可欠です。分散トランザクションをシミュレートする場合は、非常に難しいコーディング作業を検討します。コーディングをしない場合、開発者は、ドライバで JTA がサポートされていないデータベースに対して実行される分散トランザクションを無効にする必要があります。</p>
<p>BLOB/CLOB データ型</p>	<p>一貫性と移植性</p>	<p>大きなバイナリオブジェクトまたはキャラクタオブジェクトの処理は、ドライバの機能です。データベースベンダのドライバが大きなオブジェクトをサポートしていない場合は、アプリケーションでもサポートできません。複数のデータベースに接続する必要がある開発者の場合、異なるデータベースベンダのドライバでは大きなオブジェクトを処理する際の一貫性が保たれないことから影響を受けます。これを調整するためのコーディング作業の追加を検討します。</p>
<p>Multiple Open Results</p>	<p>開発プロセスの簡素化</p>	<p>Multiple Open Results のサポートは、JDBC 3.0 と J2EE 1.4 仕様に含まれる必要な機能です。データベースベンダのドライバがこれをサポートしていない場合、開発者は確実に 1 つの resultSet だけが使用されるようにそれぞれのコードを慎重にチェックする必要があります。</p>
<p>セーブポイント</p>	<p>開発プロセスの簡素化</p>	<p>セーブポイントは、JDBC 3.0 と J2EE 1.4 における仕様要件です。セーブポイントをサポートしていないデータベースベンダのドライバを使用する場合、開発者はセーブポイントが必要になるたびにセーブポイントを設定する JDBC ステートメントを個別に記述する必要があります。</p>
<p>バイナリストリーミング</p>	<p>一貫性と移植性</p>	<p>バイナリストリーミングは、複数のデータベースにアクセスする際の一貫性に関連します。バイナリストリーミングがサポートされていない場合の対処法として、開発者はカスタムコードを記述してこれを実装す</p>

		る必要があります。
バッチクエリ	パフォーマンス、一貫性、および移植性	複数のデータベースベンダのドライバが存在し、アプリケーションで複数のデータベースにアクセスする必要がある場合、開発者は、これらの異なるドライバで実装された独自のバッチクエリ動作を処理するカスタムコードを記述します。
SQLState 処理	一貫性と移植性	複数のデータベースベンダのドライバが存在し、アプリケーションで複数のデータベースにアクセスする必要がある場合、開発者はカスタムコードを記述して SQLState 情報の一貫性を保つ必要があります。これを行わないと、エラー処理がわかりにくく一貫性のないものになります。

ネイティブコネクティビティ

以前は、多くの企業が1つのデータベースで標準化すると、オープン標準が不要となり、ネイティブのデータベースライブラリに対してアプリケーションを記述することで下位互換性とパフォーマンスの向上が得られると感じていました。しかし、ユーザ要件や予算要件の変化が激しい市場において、企業の多くはその他のデータベースのサポートや新しいバージョンへのアップグレードの必要性を感じています。これを行う場合、開発者は既存のネイティブアプリケーションを記述し直す必要があります。これには開発、テスト、メンテナンスにかかる追加コストが発生します。現在、企業の多くがデータコネクティビティに対して、相互運用性という大きな利点が得られる標準ベースのアプローチを採用するようになりました。ODBC 標準は成熟し広く採用されるようになりましたが、パフォーマンスとアーキテクチャ上の柔軟性という点で解決すべき課題を残しています。

オープンソースのドライバ/プロバイダ

データコネクティビティコンポーネントをアドホック的に生成して配布するオープンソースコミュニティがいくつかあります。他のオープンソースのプロジェクトと同様に、こうしたコンポーネントは、取り組みそのものには報酬が支払われない開発者のコミュニティによって開発され、テストされます。

通常、これらのドライバは先行投資にかかる費用をほとんどまたはまったく負担せずに利用することができ、ベンダが提供するコンポーネントの一部の制限に対処することができます。ただし、これらには利点を相殺する次のような非常に深刻な制約があります。

- ソフトウェアの品質
- パフォーマンスとスケーラビリティ
- 顧客サポート
- 機能のサポート

- 目に見えないコスト

ソフトウェアの品質

オープンソースプロジェクトは、ソフトウェアの構築やテストをボランティア開発者に依存します。LinuxやApacheのような大規模なオープンソースプロジェクトとは異なり、オープンソースのデータベースドライバプロジェクトでは新機能の構築やテストを行う数千人もの開発者がかかっているわけではありません。したがって、一般的に、テスト専用のリソースは存在せず、テスト計画も厳格なものではありません。

とりわけ、高度な機能については、開発に時間を費やす開発者や実装のテストに利用できるリソースを保有する開発者はごくわずかです。

おそらくより大きく影響を受けるのは、開発プロセス自体です。コネクティビティコンポーネントのオープンソース開発には、通常、すでに商業的に成功しているコンポーネントのリバースエンジニアリングが含まれています。これは、開発者は元のソースコードにアクセスできないことから、顧客がコンポーネントを導入する際に発生する問題に対応する際に困難な状況に陥る原因となります。

パフォーマンスとスケーラビリティ

データコネクティビティは複雑であり、パフォーマンスとスケーラビリティの最適化は開発において非常にリソースを多用する部分です。オープンソースプロジェクトではパフォーマンスとスケーラビリティにリソースを集中させていないため、これらのプロジェクトのパフォーマンスとスケーラビリティは商用ソフトウェアに比較すると劣るのが一般的です。

顧客サポート

オープンソースプロジェクトは、正式な、サービスレベルによる顧客サポートを提供していません。オープンソースのデータコネクティビティコンポーネントのユーザは、顧客サポートに関するインターネットの掲示板やニュースグループに頼らざるを得ません。こうした方法による顧客サポートにありがちなレスポンスタイムやサポートの品質は、ビジネスクリティカルなアプリケーションには適していません。

機能のサポート

オープンソースプロジェクトは、最も一般的に使用されているデータベースとコネクティビティ標準機能をサポートするのが通例です。高度な機能についてはサポートされる場合とされない場合がありますが、通常はサポートされません。また、新しいデータベースバージョンや新しいバージョンのデータコネクティビティ標準のサポートは、通常、リリース後かなりの時間が経ってから実現されます。

目に見えないコスト

オープンソースドライバは無料ですが、しばしばそのコストは高くつくことがあります。パフォーマンスや機能のサポートの制限に対処しようとする、開発者はそれぞれのニーズに応じてオープンソースのドライバを頻繁に修正

する必要があります。これには開発、テスト、メンテナンス、およびサポートにかかる追加コストが発生します。

サードパーティのデータコネクティビティの評価

パフォーマンス、スケーラビリティ、移植性、および品質が重要な場合は、サードパーティのデータコネクティビティコンポーネントが、ソフトウェア開発者にとって最良の選択です。サードパーティコンポーネントは、特定のデータベースプラットフォームまたはデータベースバージョンにロックインされることなく、必要な機能がサポートされる可能性が最も高いものです。

開発者とITマネージャは、サードパーティコンポーネントに分類されている中からベンダを選択する前に、ベンダを精査する必要があります。選択する前には次の3つの主要な要因について調べます。

- ベンダが提供する製品群の包括性。
- ベンダとその製品の実績。
- ベンダはその技術においてリーダか。

包括性

サードパーティのデータコネクティビティコンポーネントを選択する主な理由は、データベースに依存しないことです。データベースへの非依存性を最大限に活用するには、サードパーティのベンダによってデータベース、データベースバージョン、コネクティビティ標準、およびプラットフォームに対する広範なサポートが提供される必要があります。そうでなければ利点が失われます。データベースベンダのドライバをサードパーティ製のポイントソリューションに置き換えた場合、うまくいけば非依存性はわずかに向上します。大幅な向上につなげるには、開発者は要件が最大限に満たされるコネクティビティを提供可能なサードパーティのコンポーネントベンダを選択する必要があります。

調査のポイント

- ベンダがサポートしているデータベースおよびバージョン。
- ベンダがサポートしているプラットフォーム。
- ベンダがサポートしているどの規格に準拠しているか。
- 自社のアプリケーションで必要な機能をベンダがサポートしているか。
- 下位互換性および新しいリリースのサポートに関するベンダの方針。
- ベンダが提供するデータベースの相互運用性のレベル。

実績

データコネクティビティの選択は、アプリケーションのパフォーマンス、信頼性、およびスケーラビリティに非常に大きな影響を与える可能性があります。

す。開発者や開発マネージャは、サードパーティコンポーネントおよびその背後にある企業を精査することが重要です。最も重要なことは、厳格なテストに合格した製品が何千という環境に導入されているベンダを選択することです。

調査のポイント

- ベンダの顧客数。顧客内容。
- ベンダの製品は顧客にどのように使用されているか。
- 新製品のテスト方法。
- 製品のパフォーマンスとスケーラビリティはどのような方法で調整され最適化されているか。
- ベンダが提供する顧客サポートの種類。
- ベンダの事業年数。
- 厳格な業界標準の認定テストに連続して合格しているベンダであるかどうか。

技術的リーダーシップ

データコネクティビティは、重要かつ複雑です。優れたデータコネクティビティは、コネクティビティのスタンダード、および基になるデータベースエンジンとその通信に使用するプロトコルについて深い知識を持っているかどうかにかかっています。開発者や開発マネージャは、データベースベンダのトップ企業と強力なビジネス関係があり、主要な標準団体と有益なつながりを持つベンダを選択する必要があります。これにより、ベンダが製品の完成品をすぐに商品化できる知識とビジネス関係を持っていることが保証されます。また、ベンダが、データコネクティビティ分野の専門のエンジニア、テスト実施者、顧客サポート担当者、および開発マネージャを置いていることも重要です。

調査のポイント

- ベンダと Oracle、Microsoft、および IBM のビジネス関係について。これらのビジネス関係の種類。
- ベンダが新しいデータベースのリリース、新機能、およびアーキテクチャ上の変更に関する情報を入手する方法と時期。
- ベンダが参加している標準団体。標準委員会でベンダが果たしている役割。
- ベンダがデータコネクティビティ標準の変更および修正に関する情報を入手する方法と時期。
- データコネクティビティ担当の代表的なスタッフおよびその実績。
- ベンダのエンジニアスタッフの数。
- ベンダの年間の研究開発費。

- ベンダの顧客サポート担当者の養成方法。
- ベンダが顧客に対してトレーニングを提供しているかどうか。

結論

データコネクティビティは、アプリケーション開発を成功に導くために重要ですが、見過ごされがちなコンポーネントです。データコネクティビティの選択は、アプリケーションのパフォーマンス、スケーラビリティ、移植性、および信頼性に大きな影響を及ぼす可能性があります。さらに、この選択は、アプリケーションの開発およびテストに必要な時間とリソース、ひいてはプロジェクト費用と実施スケジュールに多大な影響を与えます。アプリケーション開発者やマネージャは、開発のできるだけ早い段階でコネクティビティの選択についてじっくり検討してください。

主要なデータベースベンダはすべて自社の製品にドライバを装備していますが、これらのコンポーネントには大きな制約があります。データベースベンダはデータベースの販売に注力するため、中核をなすデータベース開発からリソースを転用することを最小限に抑えながら、この目標を実現するコネクティビティコンポーネントを提供しています。ベンダが提供するコンポーネントは特定のデータベースの特定のバージョン用に開発、テストされたものであり、複数のデータベース間の切り替えが難しくなるよう意図的に設計されています。データベースベンダがコネクティビティ分野に多くの投資をすることはないため、通常、ベンダが提供するコンポーネントは重要なアプリケーションに必要な品質、パフォーマンス、およびスケーラビリティに欠けています。

アプリケーション開発者やマネージャの最良の選択は、サードパーティコンポーネントです。購入する場合は、包括的な製品群と長年にわたる販売実績を持ち、データコネクティビティ市場において強いリーダーシップをとっているベンダを利用してください。

Appendix A – Database and Platform Support by Product Line

Product Line	Oracle	IBM DB2	Microsoft SQL Server	Sybase	Informix	Progress
DataDirect Connect for ODBC	Windows, Linux HP-UX IPF, HP-UX PA Risc, Solaris, AIX,	Windows, Linux HP-UX PA Risc, Solaris, AIX	Windows, Linux HP-UX IPF, HP-UX PA Risc, Solaris, AIX, Irix, Tru64, Mac	Windows, Linux HP-UX IPF, HP-UX PA Risc, Solaris, AIX, Irix, Tru64, Mac	Windows, Linux HP-UX IPF, HP-UX PA Risc, Solaris, AIX, Irix, Tru64	Windows, Linux, HP-UX PA Risc, Solaris, AIX (Available via DataDirect SequeLink)
DataDirect Connect64 for ODBC	Windows, HP-UX IPF, AIX	Windows, HP-UX IPF, AIX	Windows, HP-UX IPF, AIX	Windows, HP-UX IPF, AIX	Windows, HP-UX IPF, AIX	
DataDirect Connect for JDBC	All Operating Platforms	Window, Unix, Linux, z/OS, OS/390, AS/400	All Operating Platforms	All Operating Platforms	All Operating Platforms	Windows, HP-UX, AIX, Tru64, Solaris (Available via DataDirect SequeLink)
DataDirect Connect for ADO.NET	Windows (98, 2000, ME, XP, Server 2003)	Windows (98, 2000, ME, XP, Server 2003)	Windows (98, 2000, ME, XP, Server 2003)	Available via DataDirect SequeLink	Available via DataDirect SequeLink	

詳細情報

DataDirect Technologies is the leading provider of components for connecting software to data, providing standards-based technology that ensures consistent behavior and performance across diverse environments such as J2EE, .NET, Web, and client/server. With the most comprehensive support for ODBC, JDBC, ADO, ADO.NET and XML, DataDirect Technologies provides the easiest experience connecting software to data.

datadirect.com

datadirect.co.jp

Worldwide Sales

Belgium (French).....	0800 12 045
Belgium (Dutch)	0800 12 046
France	0800 911 454
Germany	0800 181 78 76
Japan	0120.20.9613
Netherlands	0800 022 0524
United Kingdom	0800 169 19 07
United States	800 876 3101