

データベース移行で押さえておくべきこと

アーキテクチャーと機能の違い

技術を知る



- | | | | | |
|-----------------------------------|--|--------------------------------|---------------------------------|---------------------------------------|
| <input type="checkbox"/> 導入／環境設定 | <input checked="" type="checkbox"/> 移行 | <input type="checkbox"/> 性能 | <input type="checkbox"/> チューニング | <input type="checkbox"/> バックアップ／リカバリー |
| <input type="checkbox"/> 冗長化／負荷分散 | <input type="checkbox"/> 監視 | <input type="checkbox"/> データ連携 | <input type="checkbox"/> 災害対策 | <input type="checkbox"/> 豆知識 |

近年、基幹システムの老朽化や複雑化、あるいは、運用保守コストの負担といった課題を解決するため、モダナイゼーションの必要性が高まっています。そのなかで、データ資産の活用を考慮しつつ、データベースをどのように移行するのかが重要なポイントになります。その際、運用保守コストの削減や、ベンダーロックインの回避、エコシステムの構築を理由に、商用データベースからオープンソースソフトウェア（以降、OSSと略す）系のデータベースへの移行を検討されるケースが増えています。

この記事では、Oracle Database から PostgreSQL への移行を取り上げ、押さえておくべきポイントとして、「アーキテクチャーと機能の違い」をテーマに説明します。

なお、対象となる Oracle Database のバージョンは 10g 以降、PostgreSQL のバージョンは 10 以降を想定しています。

1. アーキテクチャーの基本的な違い

データベース移行の際に意識すべき主なアーキテクチャーの違いについて説明します。

表のファイル構造

PostgreSQL は Oracle Database と同じく、オブジェクトリレーションナルデータベース管理システム（ORDBMS）であり、どちらも表（テーブル）のデータを別々の物理ファイルに保持します。ただし、PostgreSQL の物理ファイルの構成は Oracle Database と異なります。

移行の際に注意すべきポイントは、PostgreSQL は 1 つのデータファイルに複数の表のデータを格納できないことです。

Oracle Database

表や索引などのオブジェクトは、データファイルと呼ばれる物理ファイルに格納します。データファイルは、表領域（テーブル空間）に対して割り当てます。Oracle Database では以下のように表を格納することができます。

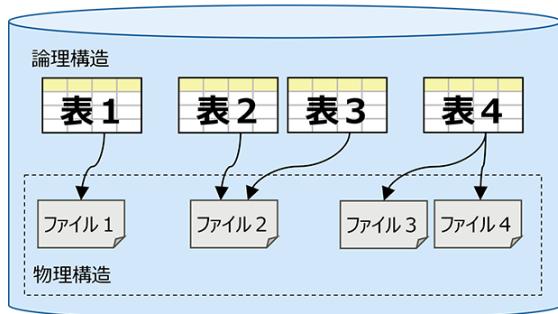
- 1 つの表を、複数のデータファイルに分割して格納できる（データファイル名を指定できる）。
- 1 つのデータファイルに複数の表のデータを格納できる。

PostgreSQL

1 つの表に対して、1 つ以上の物理ファイルを作成して格納します。ファイルの配置先は固定のディレクトリー配下です。ただし、表領域を使用することで、PostgreSQL は任意のディレクトリーにファイルを格納できます。ファイルの作成は PostgreSQL がコントロールするため、以下のように Oracle Database と異なる点があります。

- 1 つの表を、複数のデータファイルに分割して格納できる（データファイル名は指定できない）。
- 1 つのデータファイルに複数の表のデータを格納することはできない。

Oracle Database



PostgreSQL

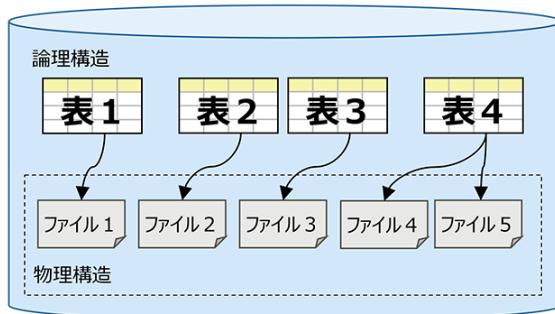


図1：表のファイル構造（概要）

同時実行制御

データベースのパフォーマンスを向上させるためには、データへの同時アクセスの管理が不可欠です。つまり、同一のオブジェクトデータへの過度のロックを防ぎながら柔軟なアクセス制御を行うことです。PostgreSQLは、Oracle Databaseと同様に読み取り一貫性を保ちながら同時実行性を実現していますが、採用している仕組みが異なります。

移行の際に注意すべきポイントは、PostgreSQLは更新または削除操作を実施したタイミングでは以前の行データが削除されないため、特に更新が多いシステムでは物理ファイルが肥大化し、パフォーマンスが劣化する可能性がある点です。

Oracle Database

DML（データ操作言語）によりデータを変更する場合は、以前のデータを UNDO 表領域という専用の領域に保存します。削除された領域は再利用することができます。あるセッションによるデータの変更がコミットされていない時に、他のセッションが同じデータを参照する場合は、UNDO レコードの情報（参照時にコミット済である情報）を参照します。

PostgreSQL

同時実行制御を追記型アーキテクチャーにより実現しています。データを更新または削除する場合は、変更対象の行データに削除マークを付け、末尾に新しいデータを追加します。この時、以前の行データは削除されず、ロールバックに必要な情報として残されます。あるセッションによるデータの変更がコミットされていない時に、他のセッションが同じ情報を参照する場合は、以前の行データを参照します。

トランザクション

PostgreSQLは、Oracle Databaseと同様にトランザクションをサポートしていますが、自動コミットおよびエラー時の動作に違いがあります。

移行の際に注意すべきポイントは、Oracle Databaseは文単位で処理をコミットするのに対し、PostgreSQLはトランザクション単位で処理をコミットする点です。移行時にアプリケーションなどの修正が必要となる場合もあるため、この違いに注意してください。

Oracle Database

- 自動コミット

DDL（データ定義言語）では、自動コミットを行います。一方、DMLでは自動コミットを行いません。

- エラー時の動作

トランザクション内でエラーが発生しても、最後に COMMIT が実行されていれば、正常に実行できた DML の処理を確定します。

PostgreSQL

- **自動コミット**

DDL では、自動コミットを行いません。一方、DML では明示的に指定しない限り、自動コミットを行います。

- **エラー時の動作**

トランザクション内でエラーが発生すると、最後に COMMIT が実行されても、ROLLBACK を実行した場合と同じ処理を行います。

排他制御（ロック機能）

PostgreSQL は、Oracle Database と同様に排他制御をサポートしていますが、排他制御の動作について、いくつか異なる部分があります。

移行の際に注意すべきポイントは、PostgreSQL はアプリケーション側で適切な設定をしない限り、ロックを取得できない場合は、ロックが取得できるまで待ち続ける点です。移行時にアプリケーションの修正が必要となる場合もあるため、この違いに注意してください。

Oracle Database

- **ロックの単位と明示的なロックの取得方法**

表ロックと行ロックの 2 種類があります。アプリケーションは以下の SQL 文を使用して明示的なロックを取得します。

- 表ロックは LOCK TABLE 文で指定します。
- 行ロックは SELECT 文に FOR UPDATE 句または FOR SHARE 句を指定します。

- **DDL 実行時のロック**

DDL 実行時には、自動的に適切なロックを取得します。ロックを取得できない場合、エラーが発生します。

PostgreSQL

- **ロックの単位と明示的なロックの取得方法**

ロックの単位と明示的なロックの取得方法は、Oracle Database と同じです。

- **DDL 実行時のロック**

DDL 実行時には、自動的に適切なロックを取得します。ロックを取得できない場合、ロックが取得できるまで待ちます。

データストレージ容量の拡張

Oracle Database と PostgreSQL では、ストレージ容量を拡張する方法に違いがあります。PostgreSQL で容量を拡張する場合、拡張したい表などの全てのデータをコピーする必要があるため、データベース管理者は容量拡張を計画的に実施する必要があります。

Oracle Database

データベース管理者は表領域のサイズを増やすことでデータストレージ容量を拡張できます。

PostgreSQL

データベース管理者は表領域のサイズを増やすことができません。容量を拡張する場合は、新しいディスクに表領域を作成し、拡張したい表や索引などを移動することで対応します。そのため、移行先データベースシステムを設計する際には、データ容量をすぐに拡張する必要のないよう、以下のことを考慮することをお勧めします。

- 適切なディスク容量と表領域の構成を決定し、パーティションを使用するかどうかを検討し、適切なバキューム計画を検討します。
- 本番環境に移行した後にデータ容量を拡張する必要がある場合は、データのコピー時間を適切に考慮して、計画的に拡張を実施します。

更新履歴情報（REDO ログ）

Oracle Database と PostgreSQL では、更新履歴情報の名前が異なります。Oracle Database では更新履歴情報を記録するファイルは REDO ログファイルと呼びますが、PostgreSQL では先行書き込みログ（WAL）と呼びます。名前は違いますが、PostgreSQL の WAL は Oracle Database の REDO ログファイルと同様の機能で使用されます。

エンコーディング（文字コード）

Oracle Database と PostgreSQL では、データベースに格納するデータやメッセージに対して、指定できるエンコーディングが異なります。移行元システムにおいて、PostgreSQL でサポートしていないエンコーディングを使用している場合は、PostgreSQL でサポートしているエンコーディングのいずれかに変更するよう検討する必要があります。

データベース環境設定ファイル

Oracle Database で使用している設定ファイルは、PostgreSQL では使用できません。Oracle Database のパラメーター設定や接続先設定などを確認し、PostgreSQL の設定ファイルに適用する必要があります。

スキーマ

PostgreSQL は、Oracle Database のようなスキーマの概念をサポートしていますが、機能には違いがあります。

Oracle Database は、ユーザー名と同じ名前のスキーマが自動的に作成されます。PostgreSQL は、デフォルトで"public"スキーマが作成されますが、ユーザー名と同じ名前のスキーマは自動的に作成されません。

移行の際には、自動的に作成されるスキーマやスキーマの検索順の差異を考慮し、同じ処理が行えるようにスキーマを追加で作成するなどの検討が必要です。

PostgreSQL は、search_path パラメーターでスキーマの検索順を適切に設定することが重要になります。PostgreSQL は、search_path パラメーターを使用して、検索するスキーマのリストであるスキーマ検索パスを管理します。クエリの実行時にスキーマ名が指定されていないと、PostgreSQL は検索パスで指定された順序でスキーマを検索し、最初に一致するオブジェクトがクエリの実行に使用されます。また、各ユーザーが作成したスキーマはデフォルトで作成したユーザーが所有するため、他のユーザーがアクセスできるようにするには、適切な権限を付与する必要があります。

2. データベースの観点ごとの移行難易度

データベース移行を実現するためには、「1. アーキテクチャーの基本的な違い」で述べた主な相違点に加えて、様々な観点での影響を認識し、移行の際の難易度を考慮した正しい見積りを行う必要があります。ここでは移行難易度を以下の表のとおり、0 から 3 で定義しました。移行難易度の数字が大きいほど移行コストがかかります。

移行難易度	移行難易度の説明
0	PostgreSQL に互換性があるため、変更の必要はありません。
1	PostgreSQL に機能の互換性がありますが、インターフェイスなどにいくつかの違いがあるため変更が必要になる場合があります。
2	PostgreSQL は主な機能をサポートしていますが、関数や構文などの一部に互換性がなく、移行方法の調査や検討が必要になります。
3	PostgreSQL に機能がなく、同等機能を実装するための代替方法の検討が必要になります。

ここからは、データベース特有の観点ごとに、移行難易度と調査・検討すべき点について説明します。また、参考として、OSS の PostgreSQL をエンタープライズ向けに強化した製品「FUJITSU Software Enterprise Postgres（以降、Enterprise Postgres と略す）」の情報も併せて示します。

基本アーキテクチャーに関する移行観点

項目	移行難易度	補足
エンコーディング (文字セット)	2	PostgreSQL は UNICODE と EUC をサポートしています。その他のエンコーディングを使用している場合には検討が必要です。
データベースの最大容量	0	
表の定量制限	0	
表の索引数	0	
表の索引形式	2	B-tree 以外を使用している場合には検討が必要です。
データ型	文字型	PostgreSQL は CHAR などの一部の文字型はサポートされていますが、最大サイズの指定方法に違いがあります。VARCHAR2、NVARCHAR2、CLOB、NCLOB、LONG などの一部の文字型はサポートされていないため、TEXT 型など PostgreSQL の類似の文字型の利用、あるいは、代替方法の検討が必要です。なお、拡張モジュールの orafce を利用することで、VARCHAR2、NVARCHAR2 が利用できます。その際にも、最大サイズの指定方法に違いがあるため検討が必要です。
	数値型	PostgreSQL に同じ名前の型がある場合でも、有効桁数や切り捨てなどの違いなどがあるため検討が必要です。
	日時型	PostgreSQL に同じ名前の型がある場合でも、精度、タイムゾーンの仕様、表示形式の違いがあるため、使用方法によっては検討が必要です。
	バイナリデータ型	Oracle Database と同じ型は PostgreSQL にないため、データ型の変更が必要です。
	XML	PostgreSQL も XML 型を使用できますが、Oracle Database のすべての機能をサポートしていないため、使用方法によっては検討が必要です。
	JSON	PostgreSQL も JSON 型を使用できますが、Oracle Database のすべての機能をサポートしていないため、使用方法によっては検討が必要です。
グローバリゼーションのサポート(多言語対応)	1	

性能に関する移行観点

項目	移行難易度	補足
メモリーチューニング	2	PostgreSQL に最適な設定に変更する必要があります。
SQL 文チューニング	2	PostgreSQL は拡張モジュールの pg_hint_plan を利用することでヒント句を使用できますが、Oracle Database と同等でないため、チューニングの際に検討が必要です。
マテリアライズド・ビュー	2	PostgreSQL はマテリアライズド・ビューを使用できますが、リフレッシュ方法、反映タイミングが異なります。高速リフレッシュができないため、マテリアライズド・ビューのサイズが大きい場合は検討が必要です。
検索処理の並列化	1	
インメモリカラムナ	3	<p>PostgreSQL に機能はありません。</p> <p>【参考】</p> <p>Enterprise Postgres では Vertical Clustered Index 機能が利用できます。</p>

可用性に関する移行観点

項目	移行難易度	補足
高可用性	2	<p>Oracle Database の Real Application Clusters などを使用したデータベースサーバーの冗長化構成に対して、PostgreSQL はアクティブ / スタンバイ構成を導入します。スタンバイは複数台で構成して、参照利用することができます。また、別途クラスタソフトウェアや Pgpool-II などを導入する検討が必要です。</p> <p>【参考】</p> <p>Enterprise Postgres ではデータベース多重化機能が利用できます。</p>
災害対策	2	<p>PostgreSQL はストリーミングレプリケーション機能により遠隔地に複製データを送ることが可能です。現行システムで Oracle Data Guard や Oracle GoldenGate を利用している場合は、機能差を調査し対策を検討する必要があります。</p> <p>【参考】</p> <p>Enterprise Postgres では、運用センターおよび災害対策センターでストリーミングレプリケーションを構成したときにでも更新ログ (WAL) の順序性を保証できるよう独自に強化されています。</p>

運用性に関する移行観点

項目	移行難易度	補足
運用管理ツール	2	PostgreSQL に対応した様々なツールが利用可能です。
DB 構成の変更 (カラム、索引の追加)	0	
索引領域の再編成	1	PostgreSQL は REINDEX 文を利用することで、索引領域の再編成ができます。

項目	移行難易度	補足
高速ローダー (SQL*Loader)	3	<p>PostgreSQL に機能はありません。COPY コマンドを利用すること、および、索引をデータ挿入後に行うことでロード時間のある程度の改善は可能です。</p> <p>【参考】 Enterprise Postgres は独自の高速ローダーが利用できます。</p>
データレプリケーション	1	
異種データベース間連携	2	PostgreSQL は Oracle Database とデータ連携が可能です。その他のデータベースとの連携については、連携方法からの検討が必要です。
データベース・リンク	2	PostgreSQL は別インスタンスとデータ連携できます。
パッチの適用	1	PostgreSQL もクラスタ環境であればローリングアップデートが可能です。

セキュリティに関する移行観点

項目	移行難易度	補足
透過的データ暗号化	3	<p>PostgreSQL に機能はありません。代替策として、アプリケーションでの暗号化 API を利用することで格納データの暗号化までは可能です。また、ストレージの暗号化機能の利用も可能ですが、部分的な対応になります。</p> <p>【参考】 Enterprise Postgres には Oracle Database と同等の透過的データ暗号化機能があります。</p>
データ・マスキング	3	<p>PostgreSQL に機能はありません。代替策としてアプリケーションなどでの対策が必要です。ただし、データベースからデータを取り出したときにはマスキングされていないため、十分な検討が必要になります。</p> <p>【参考】 Enterprise Postgres に Oracle Database と同等の機能があります。</p>
行レベルのアクセス制御	1	
セキュリティ監査	2	<p>拡張モジュールの pgAudit を利用することで対応が可能です。</p> <p>【参考】 Enterprise Postgres は監査ログ機能が利用できます。</p>

アプリケーション開発に関する移行観点

項目	移行難易度	補足
SQL 規格への準拠	2	PostgreSQL は SQL:2016 の主な機能をサポートしています。
オプティマイザ	2	PostgreSQL はオプティマイザ機能をサポートしていますが、クエリヒントや SQL の実行計画管理の一部機能に互換性がありません。PostgreSQL として SQL が最適化されるよう検討が必要です。
インターフェイス	埋め込み SQL (C 言語)	1
	ODBC	1
	JDBC	2
	.NET Framework	2
ツール	対話型 SQL 実行ツール	2
	GUI ツール	2
	開発環境ツール	2
ストアド・プロシージャ、ストアド・ファンクション	2	Oracle Database の PL/SQL に対して、PostgreSQL は PL/pgSQL が利用できますが、仕様差があり書き換えが必要になります。
アクセス制御	排他範囲	0
	排他指定	0
	デッドロックの自動検出	0
	データアクセスの同時実行性と一貫性	0
コネクション	基本管理機能	1
	生死監視機能	2
		Oracle Database は Oracle Real Application Clusters または Oracle Active Data Guard で監視するのに対して、PostgreSQL は Pgpool-II などで対応が可能です。 【参考】

項目	移行難易度	補足
管理（冗長構成時）		Enterprise Postgres はデータベース多重化機能でも対応できます。ただし、監視の構成と動作は異なります。
	2	Oracle Database は Oracle Real Application Clusters または Oracle Active Data Guard で接続するのに対して、PostgreSQL は Pgpool-II などで対応が可能です。 【参考】 Enterprise Postgres は独自の Connection Manager 機能で対応できます。ただし、監視の構成と動作は異なります。

バックアップ、リカバリーに関する移行観点

項目	移行難易度	補足
バックアップの単位	2	PostgreSQL はデータベース、インスタンス、表、またはパーティションごとにデータベースをバックアップします。ただし、Oracle Database で利用可能な表領域単位でのバックアップはできません。
バックアップ設定	1	
リカバリー	1	

Oracle Database から PostgreSQL への移行を前提とした、双方のデータベースの違いについて述べました。また、移行作業を安全に進めるためにも、「データベース移行で押さえておくべきこと～移行プロセスについて～」の記事を併せてお読みください。実際に移行作業を行う際の観点としてご利用頂けると幸いです。

参考

Enterprise Postgres は、OSS の PostgreSQL を拡張し、エンタープライズ利用に向けて信頼性、性能、セキュリティを強化したデータベースであり、ワンストップの長期サポートにも対応しています。また、他社データベースからの移行支援サービスとして、アプリケーションの移行アセスメントから資産移行、導入までのトータルなサポートを提供しています。

2022 年 6 月 10 日