

# Fujitsu Enterprise Postgres 15

## 解説書

Linux

J2UL-2836-01PJZ0(00)  
2023年4月

# まえがき

---

## 本書の目的

本書は、“Fujitsu Enterprise Postgres”を使ってデータベースを運用する方に、“Fujitsu Enterprise Postgres”とは何かを学習していただくためのマニュアルです。

本書では、Fujitsu Enterprise Postgresの特長について説明しています。

## 本書の読者

本書は、以下のような読者を対象に書かれています。

- Fujitsu Enterprise Postgresの導入を検討している方
- Fujitsu Enterprise Postgresを初めて使う方
- Fujitsu Enterprise Postgresとは何かを学習したい方
- Fujitsu Enterprise Postgresの機能概要を知りたい方

本書を読むためには、以下の知識が必要です。

- コンピュータについての一般的な知識
- 業務に関する知識
- Linuxに関する一般的な知識

## 本書の構成

本書の構成と内容は以下のとおりです。

### 第1章 Fujitsu Enterprise Postgresとは

Fujitsu Enterprise Postgresの特長について説明しています。

### 付録A 機能一覧

Fujitsu Enterprise Postgresが提供する主要機能の一覧について説明しています。

### 付録B Fujitsu Enterprise PostgresがサポートするOSS

Fujitsu Enterprise PostgresがサポートするOSSについて説明しています。

### 付録C データベースサーバとは異なるサーバで使用可能な機能

データベースサーバとは異なるサーバで使用可能な機能について説明しています。

## 輸出管理規制について

本ドキュメントを輸出または第三者へ提供する場合は、お客様が居住する国および米国輸出管理関連法規等の規制をご確認のうえ、必要な手続きをおとりください。

## 出版年月および版数

2023年 4月 初版
-------------

## 著作権

Copyright 2022-2023 Fujitsu Limited

# 目次

第1章 Fujitsu Enterprise Postgresとは.....	1
1.1 データベースの柔軟な復旧.....	2
1.2 GUIによる簡単な導入・運用管理.....	3
1.3 データベース多重化による高信頼性.....	4
1.4 Oracleデータベースからのシームレスな移行.....	5
1.5 透過的データ暗号化による格納データの保護.....	6
1.6 セキュリティ向上のためのデータ秘匿化.....	6
1.7 監査ログによるセキュリティ強化.....	7
1.8 機密管理支援機能によるアクセス制御の管理.....	7
1.9 問い合わせ計画の安定化.....	8
1.10 インメモリ機能による集計処理の高速化.....	8
1.11 高速データロード機能によるデータロード処理の高速化.....	10
1.12 Connection Managerによる可用性向上.....	11
1.13 メタキャッシュの削減および制限機能によるメモリ使用量削減.....	11
1.13.1 Global Meta Cache機能によるメモリ使用量削減.....	12
1.13.2 Local Meta Cache制限機能によるメモリ使用量削減.....	13
付録A 機能一覧.....	15
付録B Fujitsu Enterprise PostgresがサポートするOSS.....	16
付録C データベースサーバとは異なるサーバで使用可能な機能.....	17
C.1 WebAdmin機能.....	17
C.2 サーバアシスタント機能.....	17
C.3 Pgpool-IIのフェイルオーバー機能、コネクションプーリング機能、およびロードバランス機能.....	17
索引.....	20

# 第1章 Fujitsu Enterprise Postgresとは

Fujitsu Enterprise Postgresは、PostgreSQLと運用方法、アプリケーション開発用のインタフェース、SQLの互換を維持しながら、機能を拡張することで、信頼性、運用性および操作性が強化されています。

本書では、Fujitsu Enterprise Postgresが拡張した機能について説明しています。

エディション間の機能差については、“[付録A 機能一覧](#)”を参照してください。

また、Fujitsu Enterprise Postgresでは、さまざまなオープンソースソフトウェア(OSS)をサポートしています。Fujitsu Enterprise PostgresがサポートするOSSについては、“[付録B Fujitsu Enterprise PostgresがサポートするOSS](#)”を参照してください。

Fujitsu Enterprise Postgresには、以下の特長があります。

- データベースの柔軟な復旧  
障害発生時に、データベースの必須要件である最新のデータに復旧するだけでなく、任意の時点で復旧することもできます。また、任意のコピー技術を使用したバックアップ/リカバリが可能です。
- GUIによる簡単な導入・運用管理  
データベースの煩雑な操作をGUIで簡易化し、データベースを直感的に利用できます。
- データベース多重化による高信頼性  
データベース多重化により、重要なデータを保護し、高信頼なデータベースの運用を実現します。
- Oracleデータベースからのシームレスな移行  
Oracleデータベースとの互換機能を提供します。互換機能を利用することにより、既存アプリケーションの改修を局所化し、Fujitsu Enterprise Postgresへ容易に移行できます。
- 透過的データ暗号化による格納データの保護  
データベースに格納するデータの暗号化により、データ盗難の脅威から情報を守ることができます。
- セキュリティ向上のためのデータ秘匿化  
アプリケーションからの問合せに対して、一部のデータのみ改訂して参照可能とします。これにより、個人情報などの機密データを扱う際のセキュリティを向上できます。
- 監査ログによるセキュリティ強化  
監査ログにより、データベースに対する不正アクセスや権限濫用などのセキュリティの脅威に対抗することができます。
- 機密管理支援機能によるアクセス制御の管理  
機密管理支援機能により、アクセス権の設計/管理を支援し、アクセス制御を適正化することで、セキュリティの強化を実現することができます。
- 問い合わせ計画の安定化  
以下の機能を利用することで、SQL文の問い合わせ計画を制御できます。
  - オプティマイザヒント
  - 統計情報の固定化

これらの機能は、SQL文の処理性能の向上より性能の安定化を重視する基幹系の業務などで、SQL文の問い合わせ計画が変わることによる性能劣化を抑制する目的で利用します。

- インメモリ機能による集計処理の高速化  
以下の機能を利用することで、大量の行に対する集計処理でも高速に検索することができます。
  - ー カラム型のインデックスVertical Clustered Index (以降、VCI)
  - ー データのメモリレジデント機能
- 高速データロード機能によるデータロード処理の高速化  
高速データロード機能により、外部ファイルのデータをFujitsu Enterprise Postgresのテーブルに高速にロードすることができます。
- Connection Manager機能による可用性向上  
Connection Manager機能により、アプリケーションの接続先を意識せずにレプリケーション運用を継続することができます。
- Global Meta Cache機能によるメモリ使用量削減  
Global Meta Cache機能により、これらの情報の一部を共有メモリ上にキャッシュできます。これにより、システム全体のメモリ使用量の削減につながります。
- Local Meta Cache制限機能によるメモリ使用量削減  
ローカルメモリに保持するメタキャッシュを制限することで、メモリの消費量を削減します。

## 1.1 データベースの柔軟な復旧

データベースを利用したシステムでは、ディスク障害や、誤操作によるデータ破損などの危険を避けられません。このようなトラブルが発生しても利用者に多大な損害を与えず、破壊されたデータベースを確実に復旧できることが、データベースシステムに求められる要件です。

Fujitsu Enterprise Postgresでは、これらの要件に柔軟に対応したリカバリ機能を提供します。

- 最新時刻まで復旧するメディアリカバリ
- 指定時刻まで復旧できるポイントインタイムリカバリ
- 多様なコピー技術と連携可能なバックアップリカバリ

### 最新時刻まで復旧するメディアリカバリ

メディアリカバリでは、ディスク障害が発生した場合に、障害発生直前の状態へデータを復旧することができます。

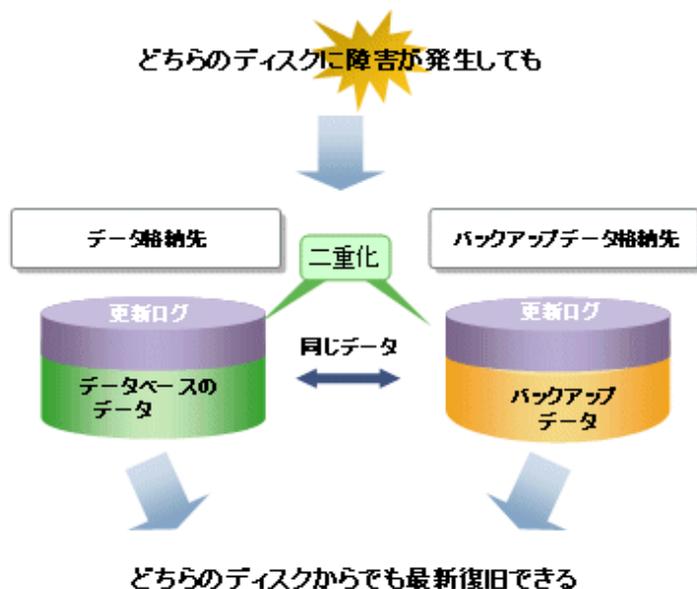
Fujitsu Enterprise Postgresでは、データベースを復旧するために、データの追加・削除などのデータベース更新操作の履歴を更新ログとして蓄積しています。また、バックアップ実行後の更新ログを、データ格納先のディスクとバックアップデータ格納先のディスクに二重化して保持しています。このため、一方のディスクに障害が発生した場合にも、もう一方のディスクのデータを使用して、データベースを最新状態に復旧することができます。

メディアリカバリは、Fujitsu Enterprise Postgresが提供するGUIツール(WebAdmin)、またはサーバコマンドから実行します。



#### 参考

WebAdminを利用したリカバリでは、WebAdminがリカバリの範囲を自動的に決定するため、復旧の手間と時間が短縮できます。



### 指定時刻まで復旧できるポイントインタイムリカバリ

ポイントインタイムリカバリでは、誤操作などにより更新されてしまったデータベースを、バックアップ実施完了の日時を指定して復旧することができます。

ポイントインタイムリカバリは、Fujitsu Enterprise Postgresのサーバコマンドから実行します。



### 多様なコピー技術と連携可能なバックアップリカバリ

ユーザーコマンドで実装した任意のコピー技術を使用して、バックアップデータ格納先または任意のバックアップ先にバックアップすることができます。



ユーザーコマンドを使用したバックアップリカバリについては、“運用ガイド”の“コピーコマンドを使用したバックアップリカバリ”を参照してください。

## 1.2 GUIによる簡単な導入・運用管理

Fujitsu Enterprise Postgresでは、データベースの導入から運用管理を行うためのGUIツールとして、WebAdminを提供しています。これにより、データベースを簡単、かつ直感的に利用することができます。

WebAdminでは、Fujitsu Enterprise Postgresのセットアップ、ストリーミングレプリケーションクラスタの作成と監視、データベースのバックアップおよびリカバリを行うことができます。また設定によって、WebAdminを使用してFujitsu Enterprise Postgresのインスタンスを単一サーバまたは複数サーバ構成で管理できます。

- セットアップ

WebAdminによるセットアップでは、インスタンスを作成します。インスタンスとは、データベースクラスタ(データ格納先のディスク上にあるデータベース格納領域)を管理する一連のサーバプロセスです。  
運用に最適な設定を自動的に判断するため、必要最低限の入力だけで、簡単にインスタンスを作成できます。

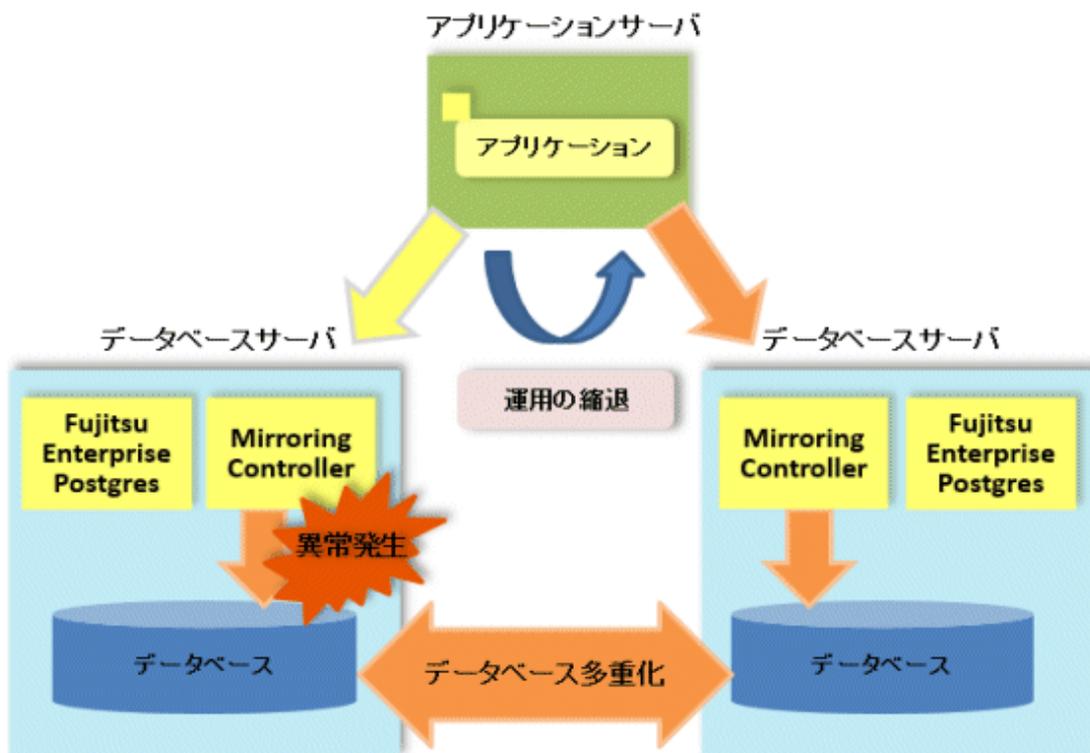
- データベースのバックアップ/リカバリ

データベースのバックアップおよびリカバリを簡単なGUI操作で行うことができます。

特に、リカバリでは、トラブル発生時に、Fujitsu Enterprise Postgresが異常箇所を自動的に特定することで、リカバリ箇所が局所化され、リカバリ時間やリカバリ手順が短縮できます。

## 1.3 データベース多重化による高信頼性

データベースを利用したシステムでは、ハードウェアやソフトウェアの故障など、様々な要因で引き起こされるデータの破壊や損失からデータを保護することが不可欠です。データベースを多重化することで、重要なデータを保護し、高信頼なデータベースの運用を実現します。



Fujitsu Enterprise Postgresは、PostgreSQLのストリーミングレプリケーション機能に基づいてデータベースを二重化するだけでなく、データベースプロセス、ディスク、ネットワークなどのデータベース運用の継続に不可欠な要素の障害検知機能と、簡易化されたスイッチオーバーやスタンバイの切り離し機能などを提供します。

スイッチオーバーが実施されても、クライアントがプライマリサーバやスタンバイサーバを自動的に識別することで、アプリケーションは物理的なサーバを意識せずに透過的に接続できます。

また、Mirroring Controllerにより、プライマリサーバ(主の業務で利用するデータベースサーバ)で異常が発生した場合に、プライマリサーバを自動的にスタンバイサーバに切り替えることができます。

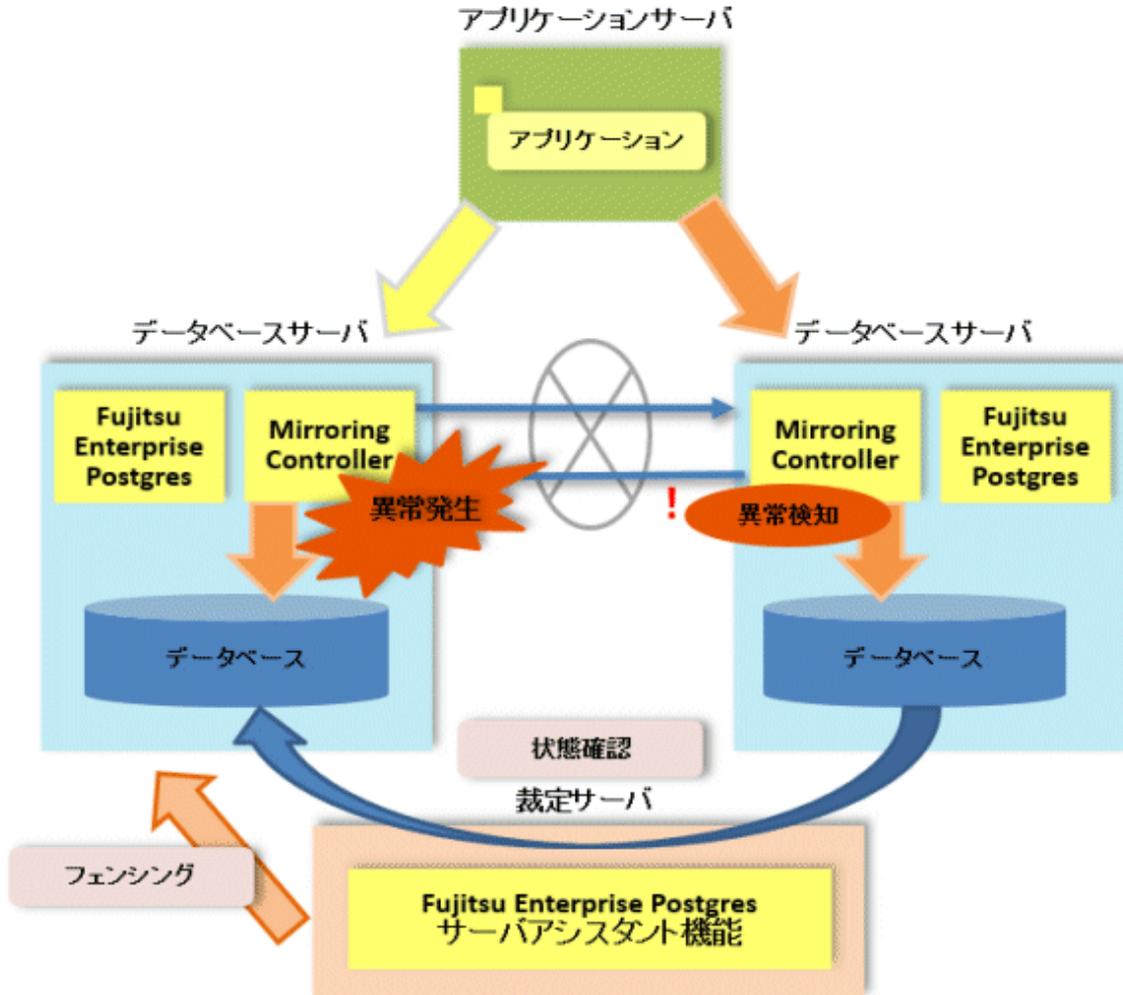
さらに、スタンバイサーバのデータを活用することで、プライマリサーバの業務と並行し、データ分析や帳票出力などの参照を行う業務が可能となります。

### 裁定サーバによる運用

Mirroring Controllerは、データベースサーバ間のネットワーク異常やサーバが不安定な状態になった場合、相手サーバの状態を正しく判断することができない場合があります。その結果、両サーバが一時的にプライマリサーバとして動作し、どちらのサーバからも更新業務が可能な現象が発生することがあります。

サーバアシスタント機能は、このような場合に、第三者としてデータベースサーバの状態を客観的に確認し、不安定なサーバを隔離(フェンシング)する機能です。

データベース多重化運用では、サーバアシスタント機能をインストールした新たなサーバ(裁定サーバ)を追加で設置することができます。裁定サーバを設置することで、前述のように両サーバが一時的にプライマリサーバとして動作する現象を防止でき、信頼性の高い運用が可能です。



#### 参照

データベース多重化については、「クラスタ運用ガイド(データベース多重化編)」の「データベース多重化運用」を参照してください。

## 1.4 Oracleデータベースからのシームレスな移行

Oracleデータベースとの互換機能として、オープンソースの「Orafce」をサポートします。

これにより、既存のアプリケーションを改修するコストを削減し、データベースの移行が容易になります。

#### 参照

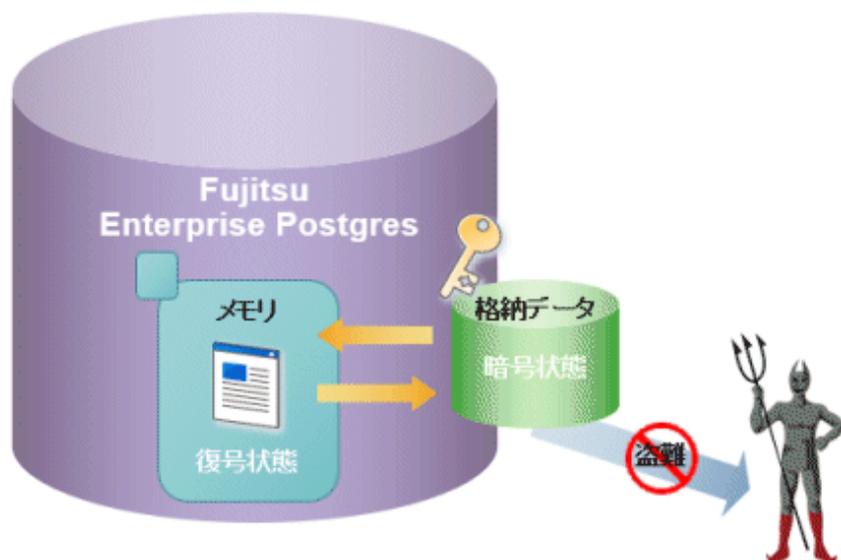
利用可能な互換機能については、「アプリケーション開発ガイド」の「Oracleデータベースとの互換性」を参照してください。

## 1.5 透過的データ暗号化による格納データの保護

データベースに格納するデータの暗号化は、クレジット業界データセキュリティ基準であるPCI DSS(Payment Card Industry Data Security Standard)の暗号化に対する以下の要求事項に対して必須です。

- ・ 機密情報(クレジットカード番号など)を判別不可能な状態にできること
- ・ 暗号化キーとデータを分離して管理すること
- ・ 定期的に暗号化キーの交換を行うこと

これに対して、Fujitsu Enterprise Postgresでは透過的データ暗号化機能を提供します。なお、PostgreSQLでは、pgcryptoという暗号化機能が適用されており、Fujitsu Enterprise Postgresでも使用することができますが、アプリケーションに暗号化を意識した修正が必要となるため透過的データ暗号化機能の使用を推奨します。



透過的データ暗号化機能では、暗号化キーの保管場所として、外部の鍵管理システムも選択することができます。

### 参照

透過的データ暗号化については、“運用ガイド”の“透過的データ暗号化による格納データの保護”または“鍵管理システムをキーストアとして使用する場合の透過的データ暗号化の運用”を参照してください。

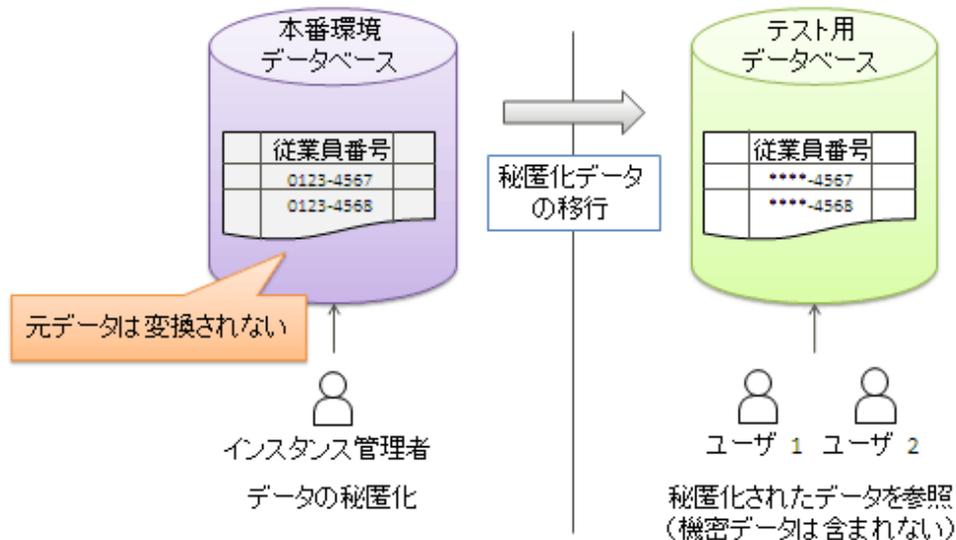
## 1.6 セキュリティ向上のためのデータ秘匿化

Fujitsu Enterprise Postgresでは、システムで扱うデータのセキュリティを保つために、データの保護を目的としたデータ秘匿化機能を提供します。

データ秘匿化機能では、アプリケーションによって発行された問合せに対して、一部のデータを改訂して参照させることができます。

たとえば、従業員データの問合せに対して、8桁の従業員番号の最後の4桁以外を“\*”で改訂して参照させる場合などに利用できます。

具体的には、データ秘匿化機能で改訂したデータをテスト用データベースに移行し、テストや開発を行うユーザーに参照させる利用方法があります。テスト時には本番環境データベースで利用するデータを用いてテストを行うことが求められます。しかし、機密データが流出するリスクが高まるため、本番環境データベースのデータをそのまま使用するべきではありません。本機能を利用することで、本番環境データベースにより近いデータをテストや開発環境でも安全に利用することができます。



### 参照

データ秘匿化については、“運用ガイド”の“データ秘匿化”を参照してください。

## 1.7 監査ログによるセキュリティ強化

データベースアクセスに関する詳細な情報を監査ログとして取得することができます。監査ログ機能を使用することで、データベースに対する不正アクセスや権限濫用などのセキュリティの脅威に対抗することができます。

PostgreSQLでは、ログ出力機能を利用することで、サーバログとして出力されたログを監査ログとして利用することができますが、SQL実行時のログとして、スキーマ名が出力されないなど、適切な分析ができないログが存在します。また、出力条件について、詳細な指定ができないため、ログ量が多く、性能劣化につながる可能性があります。

Fujitsu Enterprise Postgresの監査ログ機能は、pgauditに機能を拡張することで、データベースアクセスに関する詳細な情報を監査ログとして取得することができます。また、監査ログは、専用ログファイルまたはサーバログに出力できます。これにより効率的かつ正確なログ監視が可能になります。



### 参照

監査ログ機能については、“セキュリティ運用ガイド”の“監査ログ機能”を参照してください。

## 1.8 機密管理支援機能によるアクセス制御の管理

機密管理支援機能は、データの機密レベルに合わせたアクセス制御の実現をサポートします。また、アクセス制御にしたがって運用されていることを確認する作業もサポートします。

### アクセス制御

データ保護規則を定めた法律やルールに従うためには、機密データへのアクセスは制限されなければなりません。しかし、多様なデータと多様なタスクを実行するユーザーを持つデータベースでは、この制限を設計することは容易ではありません。なぜならば、すべてのデータベースオブジェクトとデータにアクセス可能なすべてのロールとの組み合わせについて、アクセスを許可するかどうかを決定し、データベースに定義しなければならないからです。

私たちの現実の世界では、そのようなことはしません。例えば、ある業務において、同じ機密レベルのデータをグルーピングし、そのデータにアクセスできるいくつかのロールをグルーピングします。そして、その後に、データのグループとロールのグループの組み合わせについて、アクセスを許可すべきかを検討する方が自然です。なぜならば、データが追加されたならば、そのデータがどのグループに属するかを決定すれば、自然に誰がアクセスできるのかが決まるからです。ロールを追加したときも、どのロールのグループ（機密グループと呼びます）に追加すれば良いかを考えるだけで十分です。機密管理支援機能は、このような自然な設計を支援します。

また、高い機密レベルに属するデータは、データベースにログインできるユーザーからのアクセスだけでなく、物理的な媒体やファイルへの不正なアクセスに備える必要があるかもしれません。機密管理支援機能では、高い機密レベルに属するテーブルに対して、暗号化されていることを強制することができます。同じように、機密グループに属するロールに対して、機密グループに設定された属性以下の属性を持つことを強制することができます。このような、テーブルの暗号化やロールの管理も、この機能によって自然に設計することができます。

## 運用の確認

データ保護規則を定めた法律やルールに従うためには、設計したとおりに安全にデータベースが運用されていることを確認しなければなりません。機密管理支援機能を使用しているならば、そのような心配はありません。しかし、この機能を使わずにテーブルやロールの定義が変更されたならば、それをタイムリーに検知しなければなりません。機密管理支援機能は、そのような行為を禁止したり、検知する機能を持ちません。その代わりに、監査ログを使ってそのような権限の変更などを検出するようにしてください。

しかし、検知したとしても対処を忘れるかもしれません。これを防止するために、定期的に、機密レベルや機密グループと、実際のテーブルやロールの定義が一致していることの差分を確認しなければなりません。そのときには、その差分を得るために用意された機密管理支援機能の機能を使うことができます。



### 参照

機密管理支援機能については、“セキュリティ運用ガイド”の“機密管理支援機能”を参照してください。

## 1.9 問い合わせ計画の安定化

Fujitsu Enterprise Postgresでは、SQL文とデータベースの統計情報を基に、問い合わせ計画のコストを見積もり、最もコストの低い問い合わせ計画を選択します。しかし、他のDBMSと同様に、必ずしも最適な問い合わせ計画を選択するとは限りません。例えば、データの状況変化に伴い、突然、不適切な問い合わせ計画を選択してしまう可能性もあります。

基幹系システムにおいては、性能の向上よりも性能の安定化の方が重要であり、問い合わせ計画の変化を避けたい場合があります。このような場合に、以下の機能を利用することで、問い合わせ計画を安定化することができます。

- ・ オプティマイザヒント

SQL文ごとに問い合わせ計画が指定できます。

`pg_hint_plan`を利用します。

- ・ 統計情報の固定化

データベース、スキーマ、表などのオブジェクト単位に、統計情報を固定できます。

`pg_dbms_stats`を利用します。



### 参照

オプティマイザヒントについては、“アプリケーション開発ガイド”の“オプティマイザヒント”を参照してください。

統計情報の固定化については、“アプリケーション開発ガイド”の“統計情報の固定化”を参照してください。



### 注意

オプティマイザヒント、および統計情報の固定化については、Fujitsu Enterprise Postgresのインストール時に導入される機能を利用してください。オープンソースを独自に利用している場合、これらの機能については、Fujitsu Enterprise Postgresのサポート対象外となります。

## 1.10 インメモリ機能による集計処理の高速化

Fujitsu Enterprise Postgresでは、カラム型のインデックスとデータのメモリレジデント機能によるインメモリ機能を提供しています。これにより、ディスクI/Oを削減して、集計処理を高速化することができます。

## カラム型のインデックス VCI

集計処理においては特定の列中の大部分のデータを必要とする処理が多くあります。しかし、従来の行型のデータ構造では不要な列も読み出してしまいうため、メモリやCPUキャッシュの利用効率が悪く、十分な速度が出ません。そこで、Fujitsu Enterprise Postgresはカラム型のインデックスであるVCI(Vertical Clustered Index)を提供します。これにより上記の問題を解決し、集計処理を高速化できます。

VCIは以下のような特徴を持ちます。

- ・ 既存業務への影響を抑えつつ、業務データをリアルタイムに利用した集計処理を行えます
- ・ インデックスとして提供するため、アプリの改修は必要ありません
- ・ ディスク上にもデータを持つため、障害発生時(インスタンス再起動時)に速やかにVCIを利用した集計業務を再開できます
- ・ VCIの使用メモリ量が設定値を超えた場合でも、ディスク上のデータを使って動作するため、VCIを利用した参照を継続できます

また、下記のように圧縮機能と並列機能も備えています。

- ・ 圧縮機能  
ディスク上のVCIデータを圧縮することで、ディスク容量の増加を抑えます。また、仮にディスクアクセスが必要になった場合でも、少ないオーバーヘッドで読み出すことができます。
- ・ 並列機能  
複数のCPUコアに分散させて並列に処理することで、集計処理を高速化します。

## データのメモリレジデント機能

以下の機能により、VCIのデータをメモリに常駐させ、集計処理のたびに発生するディスクI/Oを抑止します。

- ・ 事前ロード機能  
インスタンス再起動後など、アプリケーションが検索する前にVCIのデータをメモリにロードすることで、安定したレスポンスを保証します。
- ・ stable buffer機能  
他の業務データによるVCIデータのメモリ追い出しを防ぐことで、ディスクI/O回数を削減します。

## 本機能の狙い

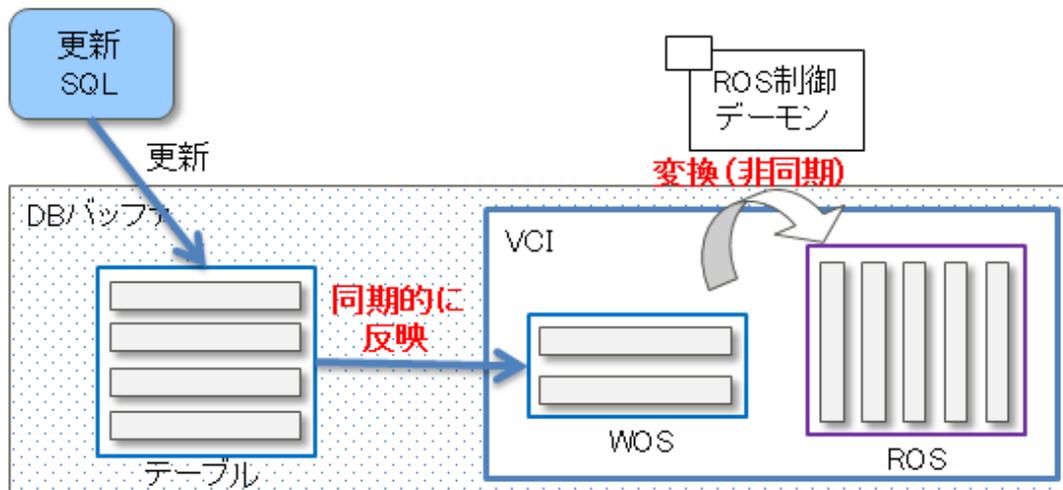
本機能は新たに追加したリソースを効率よく活用できるデータ構造をとり、通常業務における既存の集計処理を並列検索以上に高速化することが狙いです。別途提供している並列検索機能も集計処理の高速化を目的としていますが、空きリソースを活用して夜間バッチを高速化することが狙いである点が異なります。

## VCIのアーキテクチャ

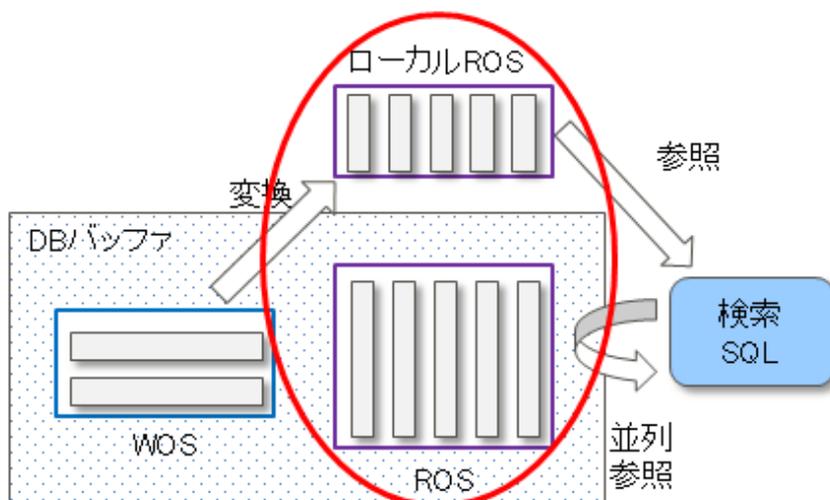
パラメータ設定等に必要となる基本的な用語の説明のため、VCIのアーキテクチャについて簡単に解説します。

業務データのリアルタイムな利用を実現するための、更新と集計それぞれについての動作を述べます。

VCIはカラム型のデータ構造であるROS(Read Optimized Store)に加えて、書き込みバッファである行型のWOS(Write Optimized Store)を持ちます。更新ごとにカラム型へ変換すると、更新処理のレスポンスに大きな影響があるためです。このため、更新時には行型のWOSへ同期的にデータを反映します。WOSに一定以上のデータが溜まったときに、ROS制御デーモンが更新とは非同期にROSへ変換します。以上のように、VCI全体でテーブルの対象列との同期を実現しながら、更新のオーバーヘッドを抑えます。



検索時にはWOSとROSを合わせて利用することで、VCIを利用しない場合と同じ検索結果を得られます。具体的には、集計処理ごとにWOSをローカルなメモリ上のLocal ROSに変換し、ROSと合わせて集計します。



### 📖 参照

VCIの導入と運用についての詳細は、“運用ガイド”の“インメモリ機能の導入と運用”を参照してください。

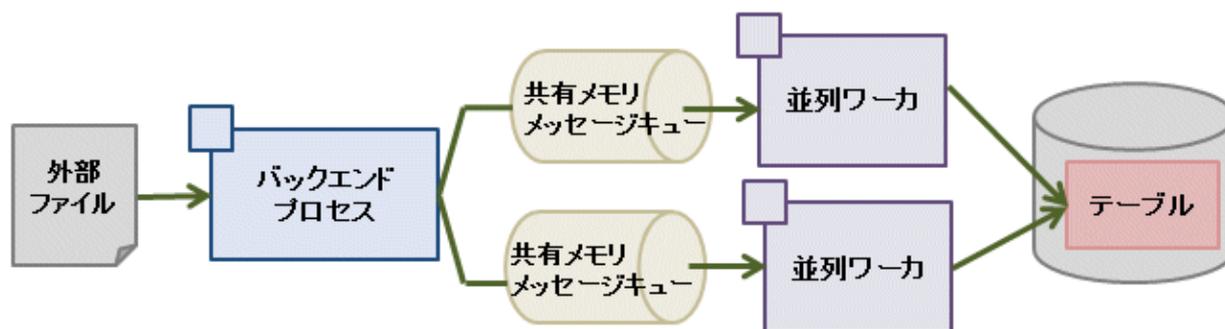
VCIを利用した検索についての詳細は、“アプリケーション開発ガイド”の“Vertical Clustered Index(VCI)を利用した検索”を参照してください。

## 1.11 高速データロード機能によるデータロード処理の高速化

高速データロード機能は、与えられたCOPY FROMコマンドを複数の並列ワークで並列実行します。これにより、外部データに対する適切な内部形式への変換処理、テーブルの作成処理およびインデックスの作成処理が並列で行われるため、大量データの高速投入が可能となります。

### 高速データロード機能のアーキテクチャ

パラメータ設定やリソース見積りに必要となるため、高速データロード機能のアーキテクチャを簡単に解説します。



高速データロード機能は、1つのバックエンドプロセスと複数の並列ワーカが協調してデータロード処理を並列実行します。バックエンドプロセスと並列ワーカ間のデータの授受は共有メモリメッセージキューを介して行われます。バックエンドプロセスは、読み込んだ外部ファイルのデータを複数の並列ワーカへ分散します。それぞれの並列ワーカは、共有メモリメッセージキューから読み込んだデータを適切な内部形式へ変換し、テーブルへ挿入します。テーブルにインデックスが作成されている場合は、インデックスのキーを抽出し、インデックスページへキーを挿入します。

### 参照

高速データロード機能についての詳細は、“運用ガイド”の“高速データロード機能”を参照してください。

## 1.12 Connection Managerによる可用性向上

Connection Managerには、以下の機能があります。これらの機能を利用することで、システムの可用性を向上させることができます。

### 生死監視機能

クライアントが動作するサーバとインスタンスが動作するサーバのそれぞれで発生したカーネルパニックや物理サーバのダウン、およびサーバ間ネットワークのリンクダウンが生じたときにこれを検知して、クライアントまたはインスタンスにそれを通知します。クライアントにはSQL接続を通じてエラー事象として通知され、インスタンスには不通となったクライアントとのSQL接続を強制回収する形で通知されます。

### 透過的接続の支援機能

アプリケーションが、レプリケーションで構成されたある属性（プライマリスタンバイ）のインスタンスに接続したいと思ったときに、そのインスタンスがどのサーバで稼働しているかを意識することなく、適切なインスタンスに接続できます。

### 参考

Connection Managerが利用可能なクライアントドライバは、libpq (C言語用ライブラリ)、ECPG (C言語による埋め込みSQL)、ODBCドライバおよびJDBCドライバです。

### 参照

Connection Managerについては、“Connection Manager 利用ガイド”を参照してください。

## 1.13 メタキャッシュの削減および制限機能によるメモリ使用量削減

SQLを実行するときには、アクセスするテーブルやインデックスの定義を参照しなければなりません。これらの定義は、SQLを実行するたびに参照するので、shared\_buffers上のデータベースの共有メモリバッファとは別に、バックエンドプロセスのローカルメモリにキャッシュしています。直接的な定義は、システムカタログの1つのタプルです。このタプルのキャッシュを“カタログキャッシュ”と呼びます。定義を利用しやすいように構造化したものを“リレーションキャッシュ”と呼びます。そして、Fujitsu Enterprise Postgresでは、この2つをまとめて“メタキャッシュ”と呼びます。

メタキャッシュは、性能のために無制限に保持し続けます。そのため、大規模なデータベースになると、1つのバックエンドプロセスがアクセスするテーブルなどが非常に多くなり、それによってバックエンドプロセスが保持するメタキャッシュも大きくなり、多数のバックエンドプロセスのローカルメモリの合算値が現実的なメモリサイズを超えてしまうかもしれません。

これに対して、メタキャッシュをバックエンドプロセス間で共有することで、インスタンス全体でのメタキャッシュを削減する機能が、Global Meta Cache機能です。現在のGlobal Meta Cache機能では、カタログキャッシュだけを共有化しています。そのため、Global Meta Cacheにおけるメタキャッシュとは、現在はカタログキャッシュのことを指しています。

現在のGlobal Meta Cache機能を使用しても、依然として共有できずにローカルメモリに保持する必要があるものは、共有メモリ上のGlobal Meta Cacheにアクセスするための情報(メタキャッシュ管理情報)とリレーションキャッシュです。Global Meta Cache機能を使用しない場合には、カタログキャッシュとリレーションキャッシュをローカルメモリに保持します。このように、ローカルメモリに保持するメタキャッシュを“Local Meta Cache”と呼びます。長時間アクセスしていないLocal Meta Cacheを削除することでサイズを制限する機能が、Local Meta Cache制限機能です。

Global Meta Cache機能とLocal Meta Cache制限機能を併用すると、最も厳格にメモリ消費をコントロールすることができます。もちろん、どちらか一方だけを使用することもできます。

ただし、Global Meta Cache機能は、共有メモリにアクセスするために、数%のオーバーヘッドがあります。また、Local Meta Cache制限機能を用いると、一度保持したメタキャッシュを破棄することがあるので、再度メタキャッシュを保持するためのオーバーヘッドが生じます。そのため、見積もりによってメモリ消費量を許容できないときに、これらの機能を使用することを検討してください。

### 1.13.1 Global Meta Cache機能によるメモリ使用量削減

---

Global Meta Cache機能は、メタキャッシュを共有メモリ上にキャッシュする機能です。共有メモリ上のメタキャッシュをGlobal Meta Cache (GMC)と呼びます。

Fujitsu Enterprise Postgres 11以前ではメタキャッシュをプロセスごとのメモリにキャッシュしていたため、大量のデータベースや多数のコネクションを使用する環境では、メモリ使用量が著しく増加するといった課題がありました。Global Meta Cache機能を使用することで、共有メモリ上のメタキャッシュが共有でき、システム全体のメモリ使用量の削減につながります。

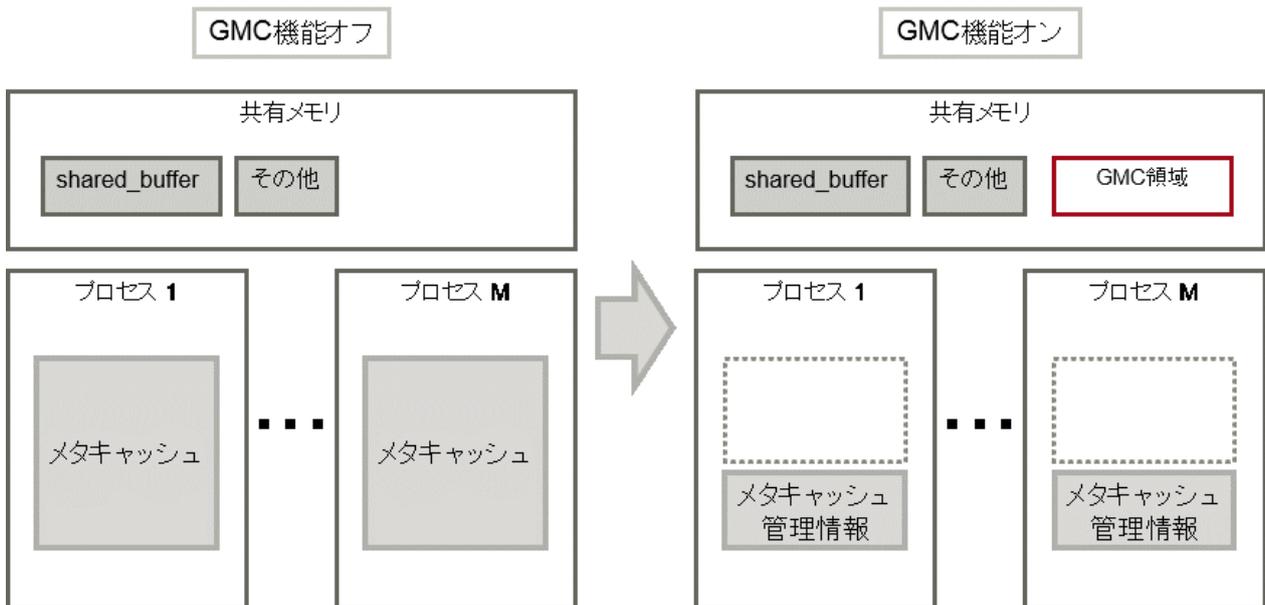
#### メタキャッシュ

クエリを処理するには、クエリのパース、プラン作成、プラン実行などの処理が必要となります。これらの一連の処理を実行するためにシステムカタログにアクセスします。一度アクセスしたシステムカタログのタプルは、プロセスごとのメモリにキャッシュしています。直接的な定義は、システムカタログの1つのタプルです。各プロセスは、毎回システムカタログの必要なタプルを検索する代わりに、メタキャッシュを検索することで高速なクエリ処理を実現しています。

また、アクセスするテーブル数およびカラム数に比例してメタキャッシュの使用量が増加します。プロセスごとにキャッシュされるため、コネクション数に比例してシステム全体のメタキャッシュの使用量が増加します。

#### Global Meta Cache機能のアーキテクチャ

Global Meta Cache機能のアーキテクチャを解説します。



Global Meta Cache機能がオンの場合は、プロセスごとのメタキャッシュを共有メモリ上のGMC領域にキャッシュします。各プロセスのメモリ上には、GMC領域を参照するための情報およびプロセス固有の作業情報がキャッシュされます。各プロセスのメタキャッシュを検索し、取得した参照情報に基づいてGMCにアクセスします。プロセスのメモリ上に参照情報がない場合は、GMC領域を検索します。GMC領域にも該当のメタキャッシュがない場合は、システムカタログにアクセスしてメタキャッシュを作成します。

また、メタキャッシュの共有化によりデータの整合性が失われることはありません。トランザクション実行中にシステムカタログやテーブル定義が変更された場合、キャッシュの削除や作成はそのトランザクションを実行しているプロセスに対して処理されます。そのトランザクションがコミットした後に、GMC領域のキャッシュが削除されます。GMC領域のキャッシュ削除時に他のトランザクションがそのキャッシュを参照中の場合は、参照がなくなるまで削除が遅延されます。コミット後のトランザクションは、古いキャッシュを参照せずに新しいキャッシュを参照します。



## 参照

Global Meta Cache機能はデフォルトで無効になっています。この機能を導入するかどうかの判断方法および使用方法は、“運用ガイド”の“Global Meta Cache機能”を参照してください。

## 1.13.2 Local Meta Cache制限機能によるメモリ使用量削減

Local Meta Cache制限機能は、長時間アクセスしていないLocal Meta Cacheを削除することでサイズを制限する機能です。

SQLがアクセスする定義のうち、Local Meta Cacheを肥大化させる主な要因は、テーブルとインデックスです。さらに、テーブルのカラムの定義もまたカタログキャッシュとして保持されます。

例えば、ある1つの寿命の長いコネクションを様々な業務で共用するようなシステムでは、1つのコネクション(すなわちバックエンドプロセス)が、多数のテーブルにアクセスすることになります。このようなコネクションが3,000本あり、各コネクションが、50,000個のテーブルにアクセスした場合には、3,000個のバックエンドが消費するメモリサイズの合計が数TBになるかもしれません。

そのようなときに、本機能を使用すると数十GB程度に削減できる可能性があります。

### Local Meta Cache制限機能の仕組み

本機能有効時のキャッシュ戦略は、指定された上限値の範囲内で可能な限りキャッシュを保持することです。もし、新しいキャッシュを保持することで、上限値を超えるのであれば、参照の局所性を考慮して、参照されていない時間が最も長いキャッシュから順に削除します。

しかし、アクティブなトランザクションが使用しているキャッシュを削除することはできないので、1つのトランザクションで多数のキャッシュを使用する場合には、上限値を超えてキャッシュを保持することもあります。この場合には、トランザクションの終了時にキャッシュを削除します。



## 参照

Local Meta Cache制限機能はデフォルトで無効になっています。この機能を導入するかどうかの判断方法および使用方法は、“運用ガイド”の“Local Meta Cache制限機能”を参照してください。

## 付録A 機能一覧

Fujitsu Enterprise Postgresが提供する主要機能の一覧を示します。

機能分類	機能
スマートソフトウェアテクノロジー	WebAdmin (スマートセットアップ、スマートリカバリー)
信頼性/可用性の向上	データベース多重化
	ユーザーコマンドを使用したバックアップ/リカバリ
	Connection Manager
アプリケーション開発	埋込みSQL連携 Java連携 ODBC連携
	Oracleデータベース互換機能
セキュリティ	透過的データ暗号化
	データ秘匿化
	監査ログ機能
	機密管理支援機能
パフォーマンス	インメモリ機能
	高速データロード機能
	Global Meta Cache機能
	Local Meta Cache制限機能
性能チューニング	オブティマイザヒント
	統計情報の固定化

## 付録B Fujitsu Enterprise PostgresがサポートするOSS

Fujitsu Enterprise PostgresがサポートするOSSを以下に示します。

OSS名	バージョンレベル	機能概要	参照先
PostgreSQL	15.0	データベース管理システム	“PostgreSQL Documentation”
orafce	3.25.1	Oracle互換SQL関数拡張	“アプリケーション開発ガイド”の“Oracleデータベースとの互換性”
Pgpool-II	4.4.0	フェイルオーバ、コネクションプーリング、ロードバランスなど	“導入ガイド(サーバ編)”の“Pgpool-II”
oracle_fdw	2.5.0	Oracleデータベースサーバへの接続	“導入ガイド(サーバ編)”の“oracle_fdw”
pg_statsinfo	-	統計情報の収集および蓄積	“導入ガイド(サーバ編)”の“pg_statsinfo”
pg_hint_plan	14.1.4.0	チューニング(統計情報管理、クエリチューニング)	<ul style="list-style-type: none"> <li>“導入ガイド(サーバ編)”の“pg_hint_plan”</li> <li>“アプリケーション開発ガイド”の“オプティマイザヒント”</li> </ul>
pg_dbms_stats	-		<ul style="list-style-type: none"> <li>“導入ガイド(サーバ編)”の“pg_dbms_stats”</li> <li>“アプリケーション開発ガイド”の“統計情報の固定化”</li> </ul>
pg_repack	1.4.8	テーブルの再編成	“導入ガイド(サーバ編)”の“pg_repack”
pg_rman	1.3.14	バックアップ/リストア管理	“導入ガイド(サーバ編)”の“pg_rman”
pgBadger	12.0	ログ解析	“導入ガイド(サーバ編)”の“pgBadger”
pg_bigm	1.2	全文検索(マルチバイト用)	“導入ガイド(サーバ編)”の“pg_bigm”
PostgreSQL JDBC driver	42.5.0	JDBCドライバ	“アプリケーション開発ガイド”の“JDBCドライバ”
psqlODBC	13.02.0000	ODBCドライバ	“アプリケーション開発ガイド”の“ODBCドライバ”

## 付録C データベースサーバとは異なるサーバで使用可能な機能

本章では、Fujitsu Enterprise Postgresのデータベースサーバと合わせて、データベースサーバとは異なるサーバにインストールして使用する機能の構成と動作環境について説明します。

なお、本章ではFujitsu Enterprise Postgresプログラムのことをサーバプログラムと表記します。

データベースサーバと異なるサーバにインストールして使用する機能は以下のとおりです。

- WebAdmin機能
- サーバアシスタント機能
- Pgpool-II(フェイルオーバー機能、コネクションプーリング機能、およびロードバランス機能)

### C.1 WebAdmin機能

WebAdminは、データベースサーバが1台の場合、通常、データベースと同じサーバ上で使用します(WebAdminプログラムは、サーバプログラムと同時にインストールすることができます)。

データベースサーバが複数台ある場合は、WebAdmin専用のサーバ(WebAdminサーバ)を用意すれば、データベースサーバのインスタンスを一括で管理することができます。この場合、WebAdminサーバにWebAdminプログラムをインストールして、データベースサーバ側にはサーバプログラムとWebAdminプログラムをインストールします。



#### 参照

- WebAdminの機能については、“1.2 GUIによる簡単な導入・運用管理”を参照してください。
- WebAdmin使用時のサーバ構成については、“導入ガイド(サーバ編)”の“最適なWebAdmin構成の決定”を参照してください。
- WebAdminの動作環境については、“導入ガイド(サーバ編)”の“前提基本ソフトウェア”を参照してください。

### C.2 サーバアシスタント機能

サーバアシスタント機能は、専用のサーバ(裁定サーバ)にサーバアシスタントプログラムをインストールして使用します。



#### 参照

- サーバアシスタント機能およびサーバ構成については、“クラスタ運用ガイド(データベース多重化編)”の“データベース多重化運用の概要”を参照してください。
- サーバアシスタント機能の動作環境については、“導入ガイド(サーバアシスタント編)”の“前提基本ソフトウェア”を参照してください。

### C.3 Pgpool-IIのフェイルオーバー機能、コネクションプーリング機能、およびロードバランス機能

Pgpool-IIは、データベースサーバとデータベースクライアントの間に位置し、接続を中継するソフトウェアです。

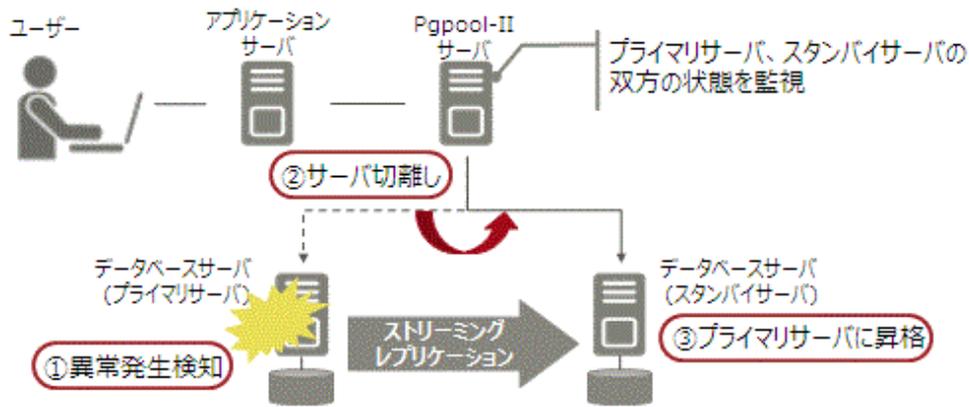
ストリーミングレプリケーション使用時のフェイルオーバー機能、コネクションプーリング機能、およびロードバランス機能があります。

#### フェイルオーバー機能

PostgreSQLでは同期ストリーミングレプリケーションを利用してデータベースを冗長化(高可用システムを構築)することができます。

同期ストリーミングレプリケーション利用時に、プライマリサーバまたはスタンバイサーバのいずれかのデータベースサーバがダウンしたり、アクセスできなくなったりすると、業務が停止します。

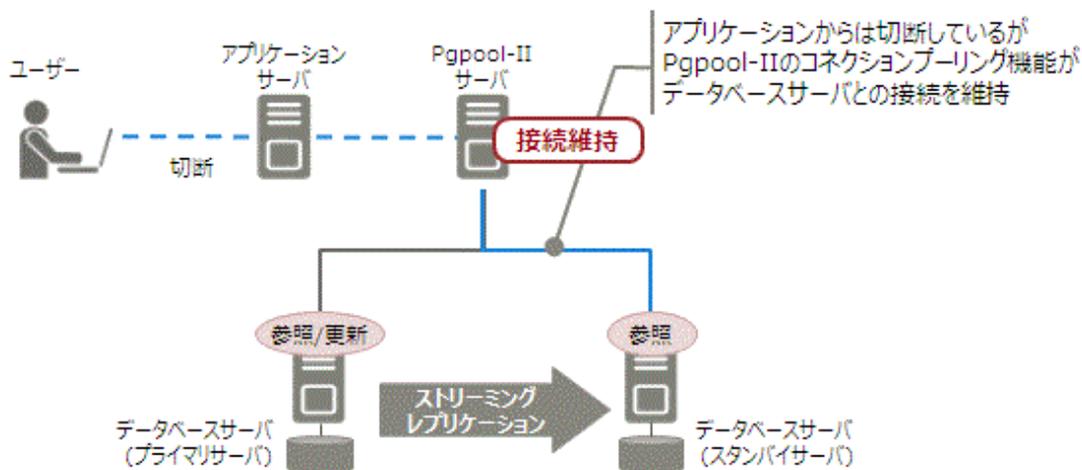
フェイルオーバー機能は各データベースの状態を監視し、異常時には自動的にサーバを切り離します。このため残ったサーバで業務を継続することができます。



## コネクションプーリング機能

データベースサーバへ確立された接続(コネクション)を維持(プーリング)し、ユーザー名、データベース、プロトコルバージョンのプロパティが同一の新規のコネクションが到着するたびに再利用する機能です。

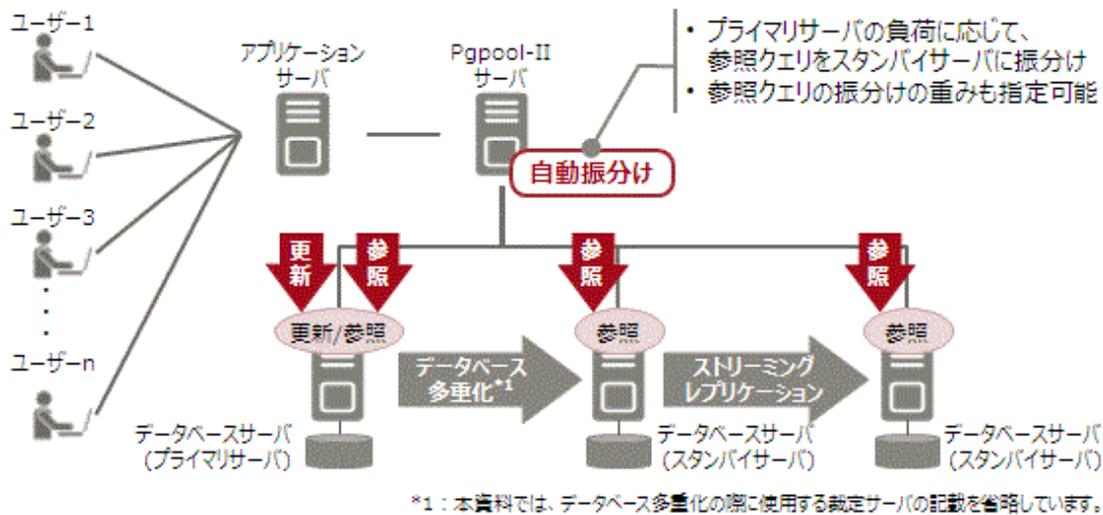
データベースサーバへの接続のオーバーヘッドを軽減でき、システム全体のスループットが改善します。



## ロードバランス機能

ロードバランス(負荷分散)機能は、参照クエリを複数のデータベースサーバに分散させ、システム全体のスループットを改善する機能です。

Fujitsu Enterprise Postgresのデータベース多重化機能、またはPostgreSQLのストリーミングレプリケーション機能とを組み合わせることで、データベースサーバの負荷を軽減します。



## 参照

- ・ Pgpool-II利用時のシステム構成については、“導入ガイド(サーバ編)”の“Pgpool-II利用時のシステム構成”を参照してください。
- ・ Pgpool-IIの動作環境については、“導入ガイド(サーバ編)”の“前提基本ソフトウェア”を参照してください。

# 索引

---

[C]	
Connection Managerによる可用性向上.....	11
[G]	
Global Meta Cacheきによるメモリ使用量削減.....	12
[O]	
Oracleデータベースとの互換機能.....	5
[か]	
カラム型のインデックスVertical Clustered Index (VCI).....	9
監査ログによるセキュリティ強化.....	7
高速データロード機能によるデータロード処理の高速化.....	10
[た]	
データのメモリレジデント機能.....	9
データ秘匿化.....	6
データベース多重化.....	4
透過的データ暗号化.....	6
[は]	
ポイントインタイムリカバリ.....	3
[ま]	
メディアリカバリ.....	2