

# OSSデータベースPostgreSQLへの取組みとOSSコミュニティへの貢献

立床雅司\*  
Masashi Tatedoko  
藤井雄規\*  
Yuki Fujii  
柿村直拓\*  
Naohiro Kakimura

追立真吾\*  
Shingo Oidate

\*情報技術総合研究所

Case Studies of Extending Open-Source Database Software  
PostgreSQL and Contributing to Open-Source Communities

## 要 旨

近年、OSS(Open-Source Software)の重要性がますます高まってきている。三菱電機はOSSの利活用に加えて、その価値向上と持続性のためOSSコミュニティへの貢献を重視しており、活動を拡大させる計画である。当社は、OSSのRDBMS(Relational DataBase Management System)であるPostgreSQL<sup>(注1)</sup>の外部データ連携への取組みを通じて、PostgreSQL内部動作への理解を深めることで分散構成での集約処理速度を12倍高速化できる余地を見だし、開発成果の一部をOSSコミュニティに提案した。さらに、OSSの普及と発展に向けて、様々なOSSコミュニティに対してパッチ投稿、ドキュメント翻訳、カンファレンス発表等による貢献活動を行った。

(注1) PostgreSQLは、PostgreSQL Community Association of Canadaの登録商標である。

## 1. ま え が き

産業界では、DX(Digital Transformation)が急速に進展している。このような潮流の中、当社では循環型 デジタル・エンジニアリング企業に向けた企業変革を加速させている。当社の経営戦略では、持続的な成長を実現するため、技術力と創造力でオープンなイノベーションを推進する戦略を掲げている。これはグループ内外での知見の融合、共創によって、未来をデザインし新しい価値をタイムリーに創出することを目指したものである。この戦略での重要な取組みの一つにOSSコミュニティへの貢献がある。OSSとは、グローバルなインターネット上のコミュニティ(これをOSSコミュニティと呼ぶ)で、世界中のエンジニアによる知見の融合と共創によって、高品質かつスピーディーに開発されるソフトウェアである。OSSの利用は世界中で拡大しており、世の中のソフトウェアプロジェクトの96%がOSSを含んでおり、ソフトウェアプロジェクトのソースコード全体のうちの77%がOSSであると言われる<sup>(1)</sup>。当社を含めた世界中のソフトウェア開発で必要不可欠なものになっている。これらOSSの価値向上と持続性のために、当社がOSSコミュニティに参画、貢献することは社会的な責務と考えている。当社は2014年に“OSS活用ガイドライン”を策定して、2018年に“社内OSSマネージャー連絡会”を設立し、OSSの思想やライセンスに則(のっと)った活用を進めてきた。OSS RDBMSのPostgreSQLは、データ操作や定義に用いられる標準SQL(Structured Query Language)をサポートし、高い拡張性を備えていることから、当社では独自のデータベース製品のフロントエンド部分に採用している。PostgreSQLの外部データベース連携機能を用いて当社データベース製品を接続するに当たって、当社データベース製品のインターフェースによる制限を回避するためPostgreSQLの内部動作の理解が必要であった。その過程で、外部データベース連携機能でPostgreSQL同士を接続した分散構成で並列集計処理を行わせる場合、大きな性能向上の余地が残されていることを見だし、PostgreSQLコア機能に対するパッチを作成してOSSコミュニティに提案するに至った。

本稿では、2章でPostgreSQLの機能拡張への取組みとして、外部データベース連携機能の活用と分散構成での集約処理高速化の詳細について述べて、3章で他のOSSでの貢献への取組みについて述べる。

## 2. PostgreSQLの機能拡張への取組み

PostgreSQLは、データの操作や定義に標準的に用いられるSQLをサポートし、データ型、関数、演算子などを追加可能な高い拡張性を備えている。この拡張性を用いて、PostgreSQLと連携する様々な拡張機能がOSSとして公開されている。また、PostgreSQLは、BSD(Berkeley Software Distribution)やMIT(Massachusetts Institute of Technology)ライセンスに類似した使用、複製、改変、配布の自由が保証されているPostgreSQLライセンスで提供されている。そのため、ソースコードを改変することでカスタマイズを行った商用又はOSSの派生製品が多数提供されている。PostgreSQL

は特定の企業が開発の主体になるような一部のOSSにみられる開発体制ではなくグローバルなコミュニティによって開発されており、誰でもインターネット上の議論に参加し開発に貢献できる。データ活用技術分野でも同様に活発な開発コミュニティが存在し、信頼性の高いソフトウェアが継続的に開発され続けている。

当社では、機能拡張を始めとしたPostgreSQLに関する様々な取組みを行っている。その一例を図1に示す。その中の機能拡張について、2.1節～2.3節で述べる。

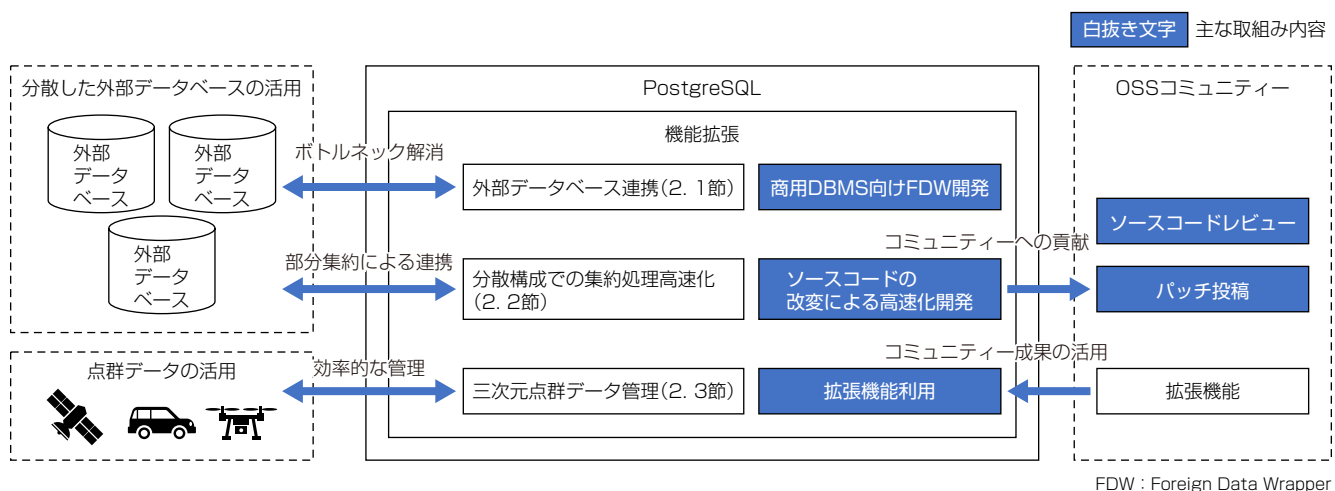


図1 - PostgreSQLへの機能拡張を始めとした様々な取組みの一例

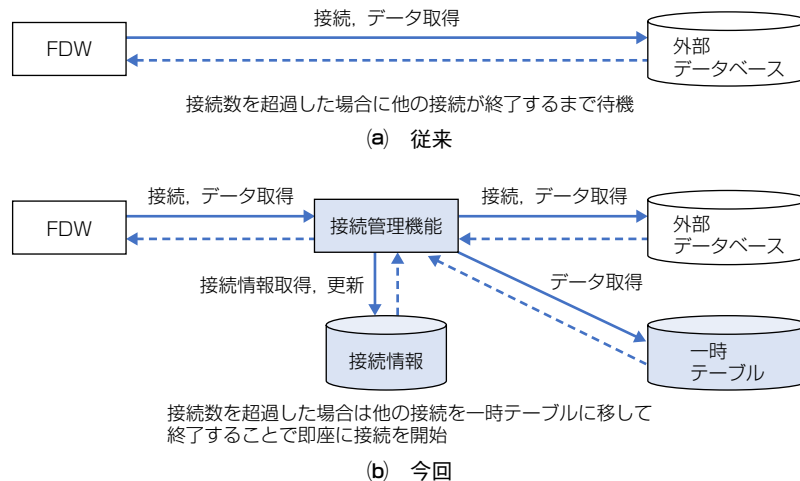
## 2.1 外部データベース連携

大量のデータを管理する場合、書き込み、読み出しの性能向上のために複数のデータベースにデータを分散し、データの特性に応じて様々なデータベースを使い分けることが多い。分散したPostgreSQLの外部データベースをPostgreSQLの外部データ連携機能であるFDWを用いて連携する場合に、外部データベースとPostgreSQLのインターフェースの違いに起因する性能のボトルネックや、処理を外部データベースに委譲できないことによる処理の非効率性が課題になる。

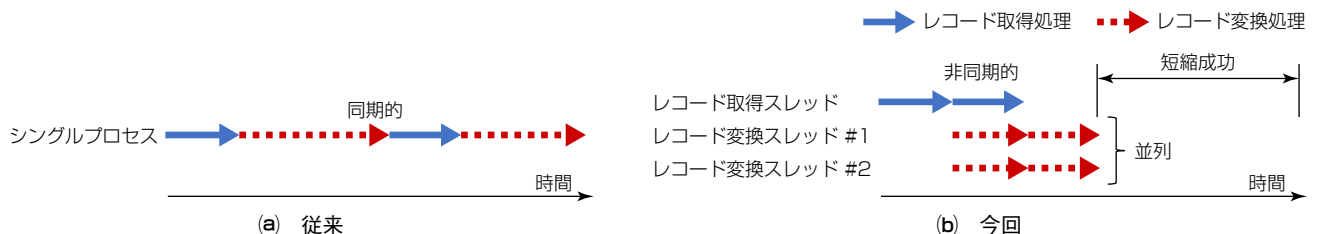
FDWは、あらかじめ連携する外部データベースのテーブルを外部テーブルとしてPostgreSQLに登録しておくことで、外部テーブルに対する問合せを透過的に行うことができる。PostgreSQLはクライアントから外部テーブルに対する問合せを受け付けると、はじめにFDWに外部データベース側で実行可能な処理の切り分けを依頼し、PostgreSQL側で実行する処理と外部データベース側で実行する処理の分担を決定する。このとき、FDWは外部データベース側で実行する処理について外部データベースが認識できる問合せに変換する。その後、PostgreSQLはFDWに対して、①外部データベースに接続し、問合せを開始する処理、②外部データベースからレコードを取得して、PostgreSQLが扱える形式にレコードを変換する処理、③問合せ完了後に外部データベースと接続を終了して後処理を行う処理、の三つを依頼する。PostgreSQLは1件ずつレコード要求して処理するため、②の処理は全てのレコードを取得するまで繰り返し行われる。

外部データベースの特性によっては、FDWを単純に実装しただけでは問題が発生する場合がある。これまでに直面した課題を二つ述べる。

一つ目は接続数の制限が厳しい外部データベースとの連携を行うときに発生する課題である。FDWは問合せ完了まで外部データベースとの接続を保持する必要があるため、外部データベースへの問合せを複数同時に実行したり、多数の外部テーブルが含まれる問合せを実行したりすると、接続数が制限を超過し、接続待ちのデッドロックが発生して問合せ全体の処理が停止してしまうことがある。この課題を解決するため、FDWに外部データベースの接続情報管理機能を追加し、接続数制限を超過した場合に既存の問合せ処理を続行しつつ新たな接続を開始するようにした<sup>(2)</sup>(図2)。接続情報管理機能は外部データベースとの接続時に新しい接続先等の情報を接続情報へ追加し、接続制限に達しているか確認する。達している場合は既に確立している接続のうち一つを選んで処理待ちレコードを全て一時テーブルに移して接続を終了し、新たな接続を開始する。接続を終了した問合せの処理は、一時テーブルからレコードを取得することで続行する。このように処理の大部分を変えることなく接続数制限の課題を解決できた。



二つ目はFDWの処理がボトルネックになって、外部データベースの高速性が発揮できないという課題である。PostgreSQLは問合せに対して、シングルプロセスで処理を行うため、PostgreSQLとFDWが同期的に実行されるとPostgreSQLの処理実行中にFDWの処理が実行できず非効率になる。また、PostgreSQLが内部でデータを扱うときの形式と外部データベースの出力形式に乖離(かいり)がある場合はレコード変換の処理時間が長くなることがある。この二つの課題に対してはPostgreSQLの処理、レコード取得処理、レコード変換処理を非同期化し、レコード変換の処理を並列化することで高速化を実現した<sup>(3)</sup>(図3)。これによってPostgreSQLが問合せ結果の作成等の処理をすると同時にFDWが次に要求されるレコードを用意できる。さらに並列化によってレコード変換時間を短縮することで、問合せ全体の実行時間を短縮することに成功した。これらの開発成果は三菱電機インフォメーションネットワーク(株)の商用データベース製品である“AnalyticMart”に適用されている<sup>(4)</sup>。このような技術開発を通じて、当社ではPostgreSQLの内部処理の知見を蓄積しており、必要に応じて適切な拡張機能を開発している。



## 2.2 分散構成での集約処理高速化

大量データ処理を高速化する手法として、複数サーバーを連携して並列処理するスケールアウトが標準的になっている。PostgreSQL開発コミュニティは、FDW等の標準機能だけでスケールアウトを実現するビルトインシャーディングの開発を進めている。既存のビルトインシャーディングでは、並列処理可能な演算は行の選択や結合等であり、集約の並列処理はサポートしていない。当社は、ビルトインシャーディングを協調領域と位置付けて、2.1節に述べた開発で蓄積した外部データベース連携のノウハウを基に改善策を提案した。

ビルトインシャーディングでは、ワーカーを制御するコーディネーターで、FDWが各ワーカーに要求する処理をSQL文で表現してワーカーへ発行し、各ワーカーのSQL文の結果を使って最終結果を生成する前提である。こうした前提と整合させるため、ワーカーがコーディネーターに渡す集約関数の状態値の生成処理(部分集約処理)の内容をSQLの集約関数で表す必要がある。ただし、一部の集約関数は部分集約を既存の集約関数で表現できない問題がある。例えば数値の平均処理の場合、ワーカーがコーディネーターに渡す状態値は合計とデータ件数の配列であるが、このような状態値を返す既存の集約関数は存在しない。

この課題に対して、部分集約処理を実行する集約関数(部分集約関数)をワーカーにインストールして、各ワーカーで実行するパッチをPostgreSQL開発コミュニティへ投稿した。並列処理可能な全ての集約関数に対して、部分集約処理を既存の仕組みで関数化し、コーディングを行うことなく簡単な定義文によってインストール可能にした。大規模データベースの分析向けクエリの国際標準ベンチマークであるTPC-Hを用いた性能評価の結果、集約処理速度がワーカー数に比例して増加することを確認した。ワーカー数が5件の場合、集約並列化と部分集約によるデータ転送オーバーヘッドの回避によって、従来のPostgreSQLと比較して集約処理速度が12倍高速になることを確認した。

パッチに関して採択権者であるコミッターからレビューを受けて議論した結果、大きく二点の課題に対応するよう要請された。一点目は、状態値の形式がサーバー間の設定やPostgreSQLのバージョン差異によって変わっていたり、通信経路で変更される場合への対策が不十分であったりする点であり、安全性や互換性を保証するよう要請されている。二点目は、既存の集約関数ごとに対応する部分集約関数を追加・管理することによる、開発者の負担や改修ソースコード量の増加の問題であり、部分集約処理の表現を簡素化するよう要請されている。今後は、これらの課題に対して対応策を検討し、同コミュニティへ再提案する予定である。

## 2.3 三次元点群データ管理

センサー技術の発展に伴い、LiDAR(Light Detection And Ranging)などによる高精度の測定が安価に実施可能になり、三次元点群データの事業活用が進んでいる。点群データは点数が膨大なため、一つの点をPostgreSQLのレコードとして扱うと非効率になる。この課題を解決するため、PostgreSQLの拡張機能pgpointcloudがOSSコミュニティから提供されている。pgpointcloudは、PcPatchと呼ばれる複数の点をまとめたデータ型を追加することで扱うレコード数を減らしつつ、点群に高速にアクセスできる機能を提供している。また、pgpointcloudは、地理空間情報が扱えるようにPostgreSQLを拡張するPostGISと組み合わせることで、点群データを他の地理空間情報と連携可能である。さらに、点群処理に特化したツールであるPDAL(Point Data Abstraction Library)にpgpointcloudを介して容易に連携でき、ノイズ除去や形式変換といった点群データの操作が可能である。当社は、これらを用いた大規模な三次元点群データ管理の実現の可能性について検証を進めている。外部データベース連携の事例と同様に、検証によって有用な知見や課題解決手段が得られた場合はOSSコミュニティにフィードバックしていく。

## 3. 様々なOSSコミュニティへの貢献活動

当社は、2章で述べたPostgreSQL以外にも様々なOSSコミュニティへ参加し、パッチ投稿による機能強化・品質改善、ドキュメントの翻訳、カンファレンスでの発表などの活動を通じてOSSの普及と発展に貢献している。これら当社の貢献活動は立ち上がったばかりであるが、当社内で“OSS貢献ポリシー”“OSS貢献ガイドライン”の策定を進めており、これらの指針の下で活動を拡大させていく計画である。PostgreSQLのほか、AIフレームワークで最も利用者の多いAIフレームワークの一つであるPyTorch<sup>(注2)</sup>、AI処理を最適化し高速化するコンパイラーのApache TVM<sup>(注3)</sup>、HummingbirdなどのOSSに対して、機能拡張やバグフィックスのためのパッチ投稿の実施や、パッチ投稿に対するレビューを実施することによってOSSコミュニティへ貢献している。Hummingbirdでは当社エンジニアの平森将裕がコミット数No.3(2024年7月現在)であり、コントリビューターから提供されるソースコードをメインラインへ統合する権限を持つコミッターとしてOSSコミュニティの発展・持続に貢献している。また深層学習コンパイラーに関するカンファレンス“TVMCon”では数少ない日本企業として発表した。パッチ投稿以外にも、ブロックチェーン・プラットフォームであるHyperledger<sup>(注2)</sup>では、公式ドキュメントの英語原文から日本語への翻訳作業といった形でも貢献している。また、この翻訳作業に関しては、Hyperledgerカンファレンスで発表した。

これらのPyTorchやHyperledgerを始めとした主要OSSコミュニティを数多く取りまとめてOSSの発展を推進する非営利組織The Linux Foundationに参画し、OSSのエコシステム全体にも貢献している。当社は、今後もOSSを利用して高品質かつスピーディーに開発するとともに、OSSコミュニティ貢献によって社会的な責務を果たしていく。

(注2) PyTorch, Hyperledgerは、The Linux Foundationの登録商標である。

(注3) Apache TVMは、Apache Software Foundationの登録商標である。



## 4. む す び

PostgreSQLの拡張開発事例とOSSコミュニティとの関わり、及び当社がこれまでに実施したAIフレームワーク分野、データ管理分野での貢献について述べた。当社は、OSSコミュニティに参画、貢献することを社会的な責務ととらえて、PostgreSQL、Apache TVM、Hummingbirdなどへパッチ投稿している。今後は社内のOSS貢献ポリシー、OSS貢献ガイドラインの整備とともに、OSSコミュニティへの貢献を拡大させていく。

## 参 考 文 献

- (1) 日本シノプシス合同会社：2024 オープンソース・セキュリティ&リスク分析レポート  
<https://www.synopsys.com/ja-jp/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html>
- (2) 藤井雄規, ほか：PostgreSQLのForeign Data Wrapperにおけるリモート問合せの同時実行数制御方式, WebDB Forum 2019論文集, 45～48 (2019)
- (3) 柿村直拓, ほか：PostgreSQLの外部連携機能におけるレコード変換処理の高速化, 第16回データ工学と情報マネジメントに関するフォーラム(DEIM2024), T2-A-3-05 (2024)
- (4) データ分析フレームワーク“AnalyticMart”とPostgreSQLとの連携による高速データ分析システム, 三菱電機技報, 96, No.1, 84 (2022)

~~~~~