

postgres_fdw における application_name のパラメーター化 PostgreSQL 15 でコミットされた機能の紹介

技術者 Blog

黒田 隼人

富士通株式会社

ソフトウェアオープンイノベーション事業本部 データマネジメント事業部

はじめに

本記事では、postgres_fdw に新たに追加されたパラメーターである postgres_fdw.application_name について紹介します。

背景 - 外部データラッパーと接続の確立

postgres_fdw とは、外部の PostgreSQL サーバーに格納されたデータへアクセスするために使用する外部データラッパー（Foreign Data Wrapper, FDW）です。外部のデータベースへ接続する際には、内部的に C クライアントライブラリーである「libpq」を使用しています。

PostgreSQL クライアントプログラムが libpq を使用する場合、「application_name」という接続パラメーターが指定できます。これは接続元のアプリケーションを識別するためのものであり、ログから各アプリケーションのアクティビティを監視するために活用できます。また似たような接続パラメーターとして「fallback_application_name」が存在し、こちらは上書き可能なデフォルトのアプリケーション名を指定するために使用します。

既存の動作の問題点 - 実際の例から

postgres_fdw が外部の PostgreSQL サーバーへ接続する際には、fallback_application_name には固定の文字列「postgres_fdw」が使用されています。よってユーザーが特別な設定をしていない場合、外部の PostgreSQL サーバーからは「postgres_fdw」というアプリケーションが接続してきたと判断されます。このとき、ユーザーが構築するシステムによっては、詳細なシステム状態の監視・解析が難しいことがありました。

1 つ例を挙げてみましょう。分析用 PostgreSQL サーバー（Analyzer）が 1 台存在し、postgres_fdw を通じてデータ格納用サーバーへアクセスしています。データ格納用 PostgreSQL サーバー（Source）上での動作を元に業務システムを改善するため、データ格納用サーバーのログ出力設定を少し変更し、接続パラメーター application_name と SQL 実行時間を表示しているとしましょう。さて、あるときデータ格納用サーバーのログを確認してみると、いくつかの SQL の実行に非常に時間がかかっていることが分かりました（図 1 の duration 部分）。この現象を解明するため、実行したクライアントを探ろうとアプリケーション名を確認すると、すべて「postgres_fdw」と記載されています。この SQL は、どのユーザーが送ったものでしょうか？ また各セッションではどのような順序で SQL を送ったのでしょうか？ 見分ける手段はありませんでした。（注 1）

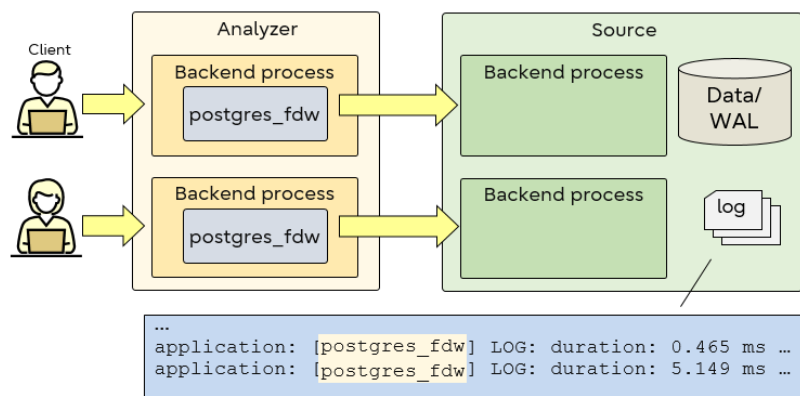


図 1：分析用 PostgreSQL サーバーからデータ格納用 PostgreSQL サーバーへのアクセス

注 1) 複数の分析用 PostgreSQL サーバーが同一のデータへアクセスし、参照および更新を行うシステムでは、CREATE SERVER 文の OPTIONS の接続パラメーターに application_name を記載することで、接続元サーバーを識別することは可能です。しかしながら、同一サーバーから同時にセッションを貼られる場合、データ格納用サーバーはこれらを識別することはできませんでした。

機能の概要

そこで私は、postgres_fdw に新たな文字列パラメーター「postgres_fdw.application_name」の追加を PostgreSQL コミュニティーに提案しました。

postgres_fdw が新たに接続を確立する際に、このパラメーターが設定されている場合は、指定された文字列を libpq 接続パラメーターとして使用します。このパラメーターは postgresql.conf からの指定に加えて、SET 文を使用したセッション単位での指定・変更ができます。

```
-- pg_settings ビューでの確認
postgres=# SELECT name, short_desc, context, vartype from pg_settings WHERE name = 'postgres_fdw.application_name';
-[ RECORD 1 ]-----
name          | postgres_fdw.application_name
short_desc    | Sets the application name to be used on the remote server.
context       | user
vartype       | string

-- SET 文によりセッション単位で変更が可能
postgres=# SET postgres_fdw.application_name TO 'test wrapper'
SET
postgres=# SHOW postgres_fdw.application_name;
postgres_fdw.application_name
-----
test wrapper
(1 row)
```

なお、すでに確立済みの接続に対して、application_name を上書きすることはできません。SET 文で新たに指定したパラメーターを使用したい場合は、postgres_fdw_disconnect()関数や postgres_fdw_disconnect_all()関数を使用し、外部の PostgreSQL サーバーとの接続を切断した後、再度接続を確立する必要があります。

このパラメーターの重要な特徴は、log_line_prefix パラメーターのように、文字列に接続元サーバーの情報を埋め込むことです。パラメーターに指定された特定の文字列は、application_name として使用される際に、接続元サーバーの情報へ置き換えられます。具体的なエスケープ文字列と、変換先の文字列は以下の通りです。

エスケープ文字	変換先
%a	接続元サーバー上での application_name
%c	接続元サーバー上でのセッション ID
%C	接続元サーバー上でのクラスタ名
%u	接続元サーバー上でのユーザー名
%d	接続元サーバー上でのデータベース名

エスケープ文字	変換先
%p	接続元サーバー上でのバックエンドプロセスの pid
%%	リテラルの%

効果 - 実際の例

本機能を利用した場合、ユーザーにはどのようなメリットがあるでしょうか？図 1 を元に説明します。

上で述べた例では、1 台の分析用 PostgreSQL サーバーから同時に複数のユーザーがアクセスしたためにログを用いたシステムの追跡が難しくなっていました。そこで、postgres_fdw.application_name を用いてこの問題を解決してみましょう。パラメーターを変更する方法としては設定ファイルを直接更新する方法がありますが、下記のように ALTER SYSTEM 文を使用することもできます。

```
-- ALTER SYSTEM 文によるパラメーターの変更
postgres=# ALTER SYSTEM SET postgres_fdw.application_name TO 'from %C user=%u';
ALTER SYSTEM
-- SIGHUP シグナルを送り、postgresql.auto.conf を再読み込みさせる
postgres=# SELECT pg_reload_conf();
pg_reload_conf
-----
t
(1 row)
-- 設定内容の確認。正しく変更されている。
-- エスケープ文字は外部の PostgreSQL サーバーへの接続時に変換されるため、ここでは実際の application_name を確認することはできない
postgres=# SHOW postgres_fdw.application_name;
postgres_fdw.application_name
-----
from %C user=%u
(1 row)
```

この設定を行ったうえで、業務システムを再開してみます。ログを再度確認してみると、下記のように出力されていました。

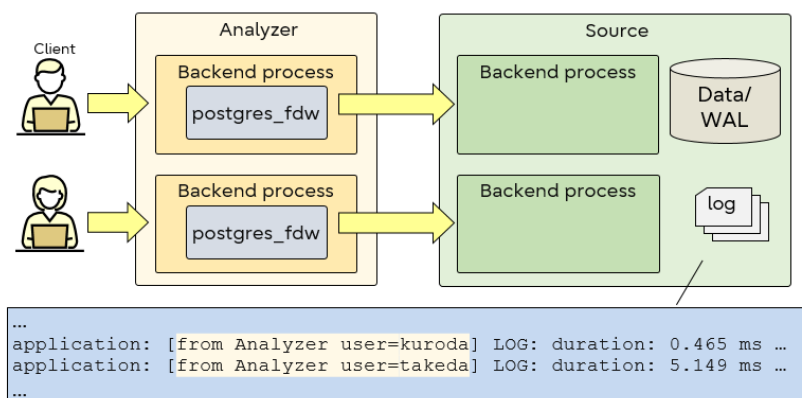


図 2：分析用 PostgreSQL サーバーからデータ格納用 PostgreSQL サーバーへのアクセス
(postgres_fdw.application_name を使用した場合)

postgres_fdw.application_name に設定した値と見比べてみると、%C がクラスタ名である“Analyzer”に、%u が Analyzer 上でのユーザー名である“kuroda”および“takeda”へ置換されていることが分かります。これらのアプリケーション名を用いて、分析用 PostgreSQL サーバーと、データ格納用 PostgreSQL サーバーの間で、ログを突き合わせた正しいワークロードの分析が行えるようになりました。

上記のように ALTER SYSTEM 文を用いてパラメーターを設定すると、業務システムを変更することなくこのパラメーターを活用することができます。また、業務システムを改修し、SET 文を使用することで処理種別ごとに application_name を変更することも可能です。

まとめ

今回は postgres_fdw の内部動作やその問題点、そして解決策である新たなパラメーターについて説明しました。データシャーディングやスケールアウト構成（注 2）といった複数のデータベースクラスタを活用したソリューションは、今後ますます必要性を増してくると考えられます。fdw を用いてそれらの機能を実現する際に、システムの監視がしやすくなる postgres_fdw.application_name は有用となるでしょう。

このほかにも、postgres_fdw には日々新しい機能が提案、追加されています。現在私が取り組んでいるのは、外部の PostgreSQL サーバーとの接続状態を確認する新たな関数を追加することです。提案が受け入れられた際には、再度このブログで取り上げたいと思っています。

注 2) スケールアウト構成の関連記事についての詳細は、以下をご覧ください。

- Enterprise Postgres のスケールアウト機能を紹介 ～ Active-Active 構成を実現 ～

注意事項

本パラメーターには任意の長さの文字列を指定できますが、application_name として使用される際には文字列長が NAMEDATALEN（通常は 64）以下となるまで切り詰められます。また本パラメーターに ASCII 文字以外を指定することもできますが、そのような文字は application_name として使用される際にクエスチョンマーク (?) へ置換されます。

詳細情報

本ブログで解説した本機能についての詳細は、PostgreSQL 文書または GitHub に投稿したコミット情報をご覧ください。

- postgres_fdw の設定パラメーターについて
 - PostgreSQL: Documentation: 15: F.38.8 Configuration Parameters (PostgreSQL オフィシャルのページへ)
<https://www.postgresql.org/docs/15/postgres-fdw.html#id-1.11.7.47.18>
- postgres_fdw.application_name について
 - postgres_fdw: Allow application_name of remote connection to be set via GUC. (GitHub.com のページへ)
<https://github.com/postgres/postgres/commit/449ab6350526e99d33363706b759951ebad7928e>
 - postgres_fdw: Allow postgres_fdw.application_name to include escape sequences. (GitHub.com のページへ)
<https://github.com/postgres/postgres/commit/6e0cb3dec10e460288d68a128e3d79d16a230cdb>

2023 年 4 月 14 日