PostgreSQL インサイド

接続パラメーター target_session_attrs のエンハンス

PostgreSQL 14 でコミットされた機能の先行紹介

技術者 Blog

Greg Nancarrow

Fujitsu Australia Software Technology
Principal Software Development Engineer

はじめに

この記事では、PostgreSQL 14 で利用可能になるデータベース接続関連の強化について説明します。この機能の歴史、使用方法、およびこの機能が提供するさまざまなメリットについて解説します。

背景

多くの PostgreSQL クライアントプログラムは、C クライアントライブラリー「libpq」を使用してデータベースにアクセスします。libpq は、クライアントプログラムが PostgreSQL バックエンドサーバーに接続し、クエリを渡し、その結果を受け取るための関数群を提供します。
PostgreSQL は libpq を通じて、接続文字列（または接続 URI）をサポートします。この文字列は、ホスト名とポート番号の組み合わせを複数指定することができます。接続が正常に確立されるまで、各候補が順番に試されます。

複数ホストの指定は、PostgreSQL 10 でサポートされました。その際、新しいパラメーターとして「target_session_attrs」が提供され、接続するバックエンドサーバーのタイプを指定できるようになりました。以降 PostgreSQL 13 までは、指定可能なパラメーター値は以下のみでした。

target_session_attrs パラメーターの値	意味
any（デフォルト）	すべての接続が受付可能です。
read-write	読み書きトランザクションがデフォルトで許容される接続だけが受付可能になります。

PostgreSQL 11 で、富士通は他の接続タイプに対応するためのパラメーター値の追加と、パラメーターの検証および接続時の内部処理の改善を提案しました。そして、この機能改善に向けて、PostgreSQL コミュニティーとのレビュー、協議、そして改善についての長いプロセスを開始しました。具体的には以下のやり取りがありました。

- 複数の富士通開発者との協議。
- 多数のパッチ・バージョンとパッチ・セットの変更：あるバージョンで、そして複数の PostgreSQL のバージョンにまたがって、パッチを結合、分割、およびリベースします。
- 様々な意見やアイデアを持つ、多くのコミュニティーメンバーからのフィードバック。
- 一部のメンバーからの機能追加の要求：例えば、類似する PostgreSQL JDBC ドライバーパラメーターとの整合性と互換性の向上（これがさらなる開発とパッチ・バージョンを引き起こしましたが、最終的にこれらの追加は却下されました）。
- コミッター自身による最終的な改善、微調整、単純化。

とても長い道のりでしたが、遂に PostgreSQL 14 で、この libpq のエンハンスがコミットされました！

機能概要

PostgreSQL 14 は、target_session_attrs パラメーターに次の値をサポートします。新規の値はハイライトで表示しています。

target_session_attrs パラメーターの値	意味
any（デフォルト）	成功する接続はすべて受付可能です。
read-write	セッションはデフォルトで読み書きトランザクションを受け入れる必要があります。（つまり、サーバーはホットスタンバイモードではなく、default_transaction_read_only パラメーターは off であるべきです）
read-only	セッションは、デフォルトでは読み書きトランザクションを受け入れません。（「read-write」の逆）（注）
primary	サーバーがホットスタンバイモードであってはなりません。
standby	サーバーはホットスタンバイモードでなければなりません。
prefer-standby	最初にスタンバイサーバーを探しますが、リストされたホストのいずれもスタンバイサーバーでない場合は、「any」モードで再試行します。

注　default_transaction_read_only 設定パラメーターが on に設定されているため、PostgreSQL サーバーは読み取り専用であってもホットスタンバイモードではないことに注意してください。

サポートされるクライアント

libpq ライブラリーをリンクする PostgreSQL クライアントは、target_session_attrs パラメーターと、ここで説明された機能の改善を利用することができます。これには PostgreSQL 言語サポートライブラリー（psycopg2［Python 用ドライバー］や psqlODBC［C 言語インターフェイスを備えた PostgreSQL ODBC ドライバー］など）のほとんどが含まれますが、PostgreSQL JDBC ドライバーや Npgsql（PostgreSQL 用 ADO.NET データ・プロバイダー）は含まれません。JDBC ドライバーは、primary / secondary / preferSecondary の値を受け付ける「targetServerType」接続パラメーターを使用して、同様の機能をサポートします。Npgsql には現在同様の機能はありません。

接続パフォーマンスの向上

新しい target_session_attrs パラメーターの値に加え、PostgreSQL 14 以降のサーバーに接続する際の接続パフォーマンスが改善されました。これは、セッションステータスに関連する報告可能な（GUC_REPORT）設定変数を使用することで実現されました。

GUC 変数	説明
default_transaction_read_only	PostgreSQL 14 で報告可能（GUC_REPORT）にしました。
in_hot_standby	PostgreSQL 14 での新しい GUC_REPORT 変数です。

これらは、接続が成功すると、セッション状態を決定するための余分なネットワークラウンドトリップを抑えながら、サーバーによってクライアントに直接報告されます。そのため、古いバージョンのサーバーに接続すると、SHOW クエリまたは SELECT クエリが発行され、セッションの読み取り専用状態またはサーバーのホットスタンバイ状態が検出されます。
セッション中にサーバーがプライマリーに昇格した場合、in_hot_standby もクライアントに報告されることに注意してください。

例

「psql（PostgreSQL の端末ベースのフロントエンド）」は libpq を使用しているため、コーディングすることなく target_session_attrs パラメーターのテストに使用することができ、便利です。2 つの利用可能なローカルサーバーのいずれかに接続しようとするときに、異なる target_session_attrs パラメーターの値を使用する簡単な例を以下に示します。

2 つのローカル・サーバー・インスタンスを作成して起動します（ポート 5432 および 5433 で実行）。デフォルトでは、どちらも読み書きトランザクションを受け入れます。

```
$ pg_ctl -D ./testdb1 initdb
$ pg_ctl -D ./testdb2 initdb
$ pg_ctl -D ./testdb1 -o '-p 5432' -l testdb_1.log start
$ pg_ctl -D ./testdb2 -o '-p 5433' -l testdb_2.log start
```

読み取り専用トランザクションのみを受け入れるサーバーへの接続を試みます。

```
$ psql "host=localhost,localhost port=5432,5433 dbname=postgres target_session_attrs=read-only"

psql: error: connection to server at "localhost" (::1), port 5432 failed: session is not read-only
connection to server at "localhost" (::1), port 5433 failed: session is not read-only
```

次に、testdb2/postgresql.conf を編集して、「default_transaction_read_only = on」を追加します。再起動して接続を再試行します。

```
$ pg_ctl -D ./testdb2 -o '-p 5433' -l testdb_2.log restart

waiting for server to shut down.... done
server stopped
waiting for server to start.... done
server started
```

読み取り専用サーバーへの接続を試みます。

```
$ psql "host=localhost,localhost port=5432,5433 dbname=postgres target_session_attrs=read-only"

psql (14devel)
Type "help" for help.

postgres=# show port;
 port
------
 5433
(1 row)
```

ホットスタンバイモードのサーバーへの接続を試みます。

```
$ psql "host=localhost,localhost port=5432,5433 dbname=postgres target_session_attrs=standby"

psql: error: connection to server at "localhost" (::1), port 5432 failed: server is not in hot standby mode
```



```
connection to server at "localhost" (::1), port 5433 failed: server is not in hot standby mode
```

プライマリサーバー（スタンバイモードではないサーバー）への接続を試みます。

```
$ psql "host=localhost,localhost port=5432,5433 dbname=postgres target_session_attrs=primary"

psql (14devel)
Type "help" for help.

postgres=# show port;
 port
------
 5432
(1 row)
```

ホットスタンバイモードのサーバーへの接続を優先します。

```
$ psql "host=localhost,localhost port=5432,5433 dbname=postgres target_session_attrs=prefer-standby"

psql (14devel)
Type "help" for help.

postgres=# show port;
 port
------
 5432
(1 row)
```

まとめ

新たにサポートされた target_session_attrs パラメーターの値により、クライアントが希望するターゲットサーバー接続をより細かく選択できるようになり、マルチホスト接続に役立ちます。また、プライマリサーバーの負荷を軽減するために、読み取り専用の要求をスタンバイサーバーにリダイレクトできるなど、スケーリングの初歩的な形式も提供します。これらの新しい target_session_attrs パラメーターの値によって、より柔軟にマルチホスト接続文字列 / URI を指定できるため、一部の PostgreSQL フェイルオーバーソリューションでは、利用可能なサーバーに再接続する際の利便性が向上するでしょう。

2021 年 6 月 11 日