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

藤井 雄規[1,a)]　立床 雅司[1]　高田 佳典[1]

概要：PostgreSQL は Foreign Data Wrapper（以下，FDW）により目的特化のデータベース管理システム（以下，DBMS）と標準 SQL で連携可能である．一般的な FDW の実装では，連携先の DBMS（以下，外部 DBMS）にて同時に実行可能な問合せ（以下，リモート問合せ）の数に上限がある際に，PostgreSQL の問合せの実行可能性を保証できない問題がある．この問題は，待ち状態のリモート問合せの数が上限に達した状態で，PostgreSQL が FDW に新たなリモート問合せの開始を要求する場合に発生する．本稿では実行可能性を保証するため FDW によりリモート問合せの同時実行数を制御する方式を提案した．

A Method for Limiting a Number of Concurrent Remote Queries on Foreign Data Wrapper in PostgreSQL

1. はじめに

PostgreSQL[1] は SQL/MED[2] に基づき FDW と呼ばれるモジュールにより外部 DBMS と標準 SQL で連携する機能を提供している．FDW は外部表と呼ばれる表を参照する問合せの一部の処理をリモート問合せに書き換えて実行する．外部表は，外部 DBMS の表に対応する PostgreSQL の仮想的な表である．目的特化の DBMS に標準 SQL でアクセスする FDW が実現されている [3][4][5].

FDW の一般的な実装では，PostgreSQL からリモート問合せの結果 1 件の要求を受理する度に外部 DBMS から結果レコードを 1 件フェッチして PostgreSQL に受渡しする（以下，逐次転送方式）[3][4]．ただし，外部 DBMS に同時実行問合せ数の上限がある際には，PostgreSQL の問合せの実行可能性を保証できない問題がある．この問題は，待ち状態のリモート問合せの数が上限に達した状態で，PostgreSQL が FDW に新たなリモート問合せの開始を要求する場合に発生する．よって，従来から知られている同時実行問合せ数が一定値以下になるまで待ち合せる方式 [6] では実行可能性を保証できない．

一方で FDW と類似のレコード集合を出力するユーザ定義関数を用いた連携方式 [7] のように，リモート問合せの開始直後に結果レコードを全件 PostgreSQL に転送する方式（以下，一括転送方式）とすれば問合せの実行可能性を保証できる．しかし，常に一括転送方式でリモート問合せを実行すると，一時ファイルの I/O データサイズが増大し処理速度が低下するという問題がある．

従来から異種データベースの連携技術においては，同時実行制御の必要性は指摘されていた [8]．しかし問合せの実行可能性を保証しつつ，一括転送方式よりも処理速度を向上させる方式は明らかにされていなかった．

本稿では，逐次転送方式において問合せが実行不可能となる場合に実行中のリモート問合せと新たに開始を要求するリモート問合せの間に成立する関係を利用し，FDW にて逐次転送方式と一括転送方式を切り替えることで課題を解決するリモート問合せの同時実行数制御方式を提案する．

2. 想定するシステム

本稿では PostgreSQL 10 を対象とし，PostgreSQL と外部 DBMS が 1 件ずつあり，外部 DBMS が以下の条件を満たすことを想定する．

想定 (1) 外部 DBMS に同時実行問合せ数の上限値 M があり，M を超える問合せの開始要求は待たされる

想定 (2) ある $N \leq M$ が存在し，リモート問合せの同時実行数が N 以下であれば実行可能性が保証される

[1] 三菱電機株式会社情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Kanagawa 247-8501, Japan
[a)] Fujii.Yuki@df.mitsubishielectric.co.jp

想定 (2) は PostgreSQL を介さずに外部 DBMS にアクセスする検索アプリケーションがあることを想定した条件である．特に，N が数件と小さい状況を想定する．

3. 従来の外部表を参照する問合せ処理の実装

PostgreSQL は問合せ 1 件にサーバプロセス 1 件を対応させ複数の問合せを並列に処理する．PostgreSQL は 1 件の問合せの処理を**図 1** のように内部的に複数のノードからなる木構造の実行計画に分解する．FDW のリモート問合せを実行するノードは，外部表スキャンと呼ばれる．外部表スキャンの処理は FDW のモジュール毎にプログラミングされる．PostgreSQL は実行計画の各ノードを再帰的に実行する．従って PostgreSQL は結合等を含む 1 件の問合せにおいて，ある外部表スキャンの結果を 1 件取得した後，別の外部表スキャンの結果を 1 件取得しようとする場合がある．例えば，結合を含むある問合せの実行計画は図 1 のようになり，PostgreSQL は最初に外部表スキャン A の結果を 1 件取得した後，外部表スキャン B の結果を 1 件取得しようとする．PostgreSQL が 1 件の問合せにおいて同時に実行するノードは 1 件のみである．

PostgreSQL の外部表を参照する問合せの処理は大きく 4 つのフェーズに分かれる．各外部表スキャンは自由に利用可能なメモリ領域（以下，プライベート領域）を確保し，各フェーズの間でのデータの受渡しが可能である．以下各フェーズにおける従来の FDW の一般的な実装を説明する．

(1) 問合せ最適化
外部表スキャンに対応するリモート問合せを作成する．また，別途収集した外部 DBMS の統計情報を基にリモート問合せの結果レコードの数や平均レコードサイズの推定値を作成する．作成した情報を外部表スキャンの情報としてプライベート領域に登録する．

(2) 問合せ初期化
リモート問合せの結果レコードを PostgreSQL のレコードに変換するために必要な情報を PostgreSQL から受理し，プライベート領域に登録する．ここで必要な情報とは，リモート問合せと外部表スキャンの結果レコードを構成する各列のデータ型と，これら 2 つの対応情報である．

(3) 問合せ実行
外部表スキャンに対応するリモート問合せの結果レコードを 1 件フェッチし，PostgreSQL のレコードに変換し，PostgreSQL に渡す．問合せ実行フェーズで最初に外部表スキャンのフェッチ要求を PostgreSQL から受理した際に，対応するリモート問合せを開始する．結果レコードがなくなると外部表スキャンに対応するリモート問合せの終了を外部 DBMS に要求する．

(4) 問合せ終了
独自に確保したメモリ領域の解放処理等を行う．

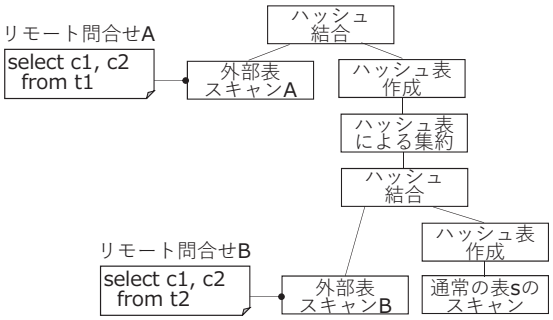

図 1: 複数の外部表スキャンを含む実行計画の例

Fig. 1 An example of query plans executing multiple foreign scans

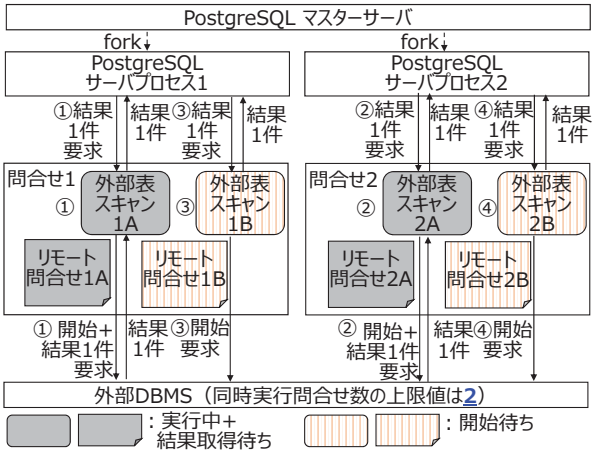

図 2: 問合せが実行不可能となる例

Fig. 2 A case in which any query becomes not executable

4. 問合せが実行不可能となるケース

本章では逐次転送方式にて問合せが実行不可能となる例を説明する．以下，想定（1）の M と想定（2）の N に関して $M = N = 2$ と仮定する．PostgreSQL が図 1 の問合せを 2 件（以下，問合せ 1，2）同時に受理したとする．問合せ 1，2 の実行計画に含まれる外部表スキャンと対応するリモート問合せを外部表スキャン 1A，1B，2A，2B，リモート問合せ 1A，1B，2A，2B とする．3 章で示したように，PostgreSQL は複数の問合せを並列に実行する．よって，FDW がリモート問合せ 1A，1B を並列に開始し，その後リモート問合せ 2A，2B を並列に開始しようとする場合がある（**図 2**）．FDW は，リモート問合せ 1A，2A を開始して結果レコードを 1 件ずつ取得（図 2 中①②）した後，リモート問合せ 1B，2B の開始を要求する（図 2 中③④）．ここで，想定（1）により外部表スキャン 1B，2B は実行中のリモート問合せ 1A，2A が終了するのを待つ．当該時点で問合せ 1，2 にて実行中のノードは各々外部表スキャン 1B，2B である．PostgreSQL は 1 件の問合せで同時に実行するノードは 1 件であるため，外部表スキャン 1A，2A は PostgreSQL の結果取得要求の待ち状態となる．以上から，問合せ 1 及び 2 が実行不可能となる．

5. リモート問合せの同時実行数制御方式

5.1 方式の概要

問合せが実行不可能となる場合に以下の 2 条件が両方成立することに注目し，FDW にて問合せが実行不可能となる状況を検知・解消する方式を提案する．

条件 (1) 実行中のリモート問合せの数が N 件（以下，$RQ_e(1)$〜$RQ_e(N)$）

条件 (2) $RQ_e(i)$ に対応する PostgreSQL の問合せ $PQ(i)$ はリモート問合せ $RQ_s(i)(\neq RQ_e(i))$ の開始を要求中 $(i = 1, 2, ..., N)$

以下，検知と解消の方式を個別に説明する．

5.1.1 検知の方式

条件（2）にて，$RQ_e(i)$ と $RQ_s(i)$ は $PQ(i)$ により関連づいている．提案方式では，実行中のリモート問合せと開始要求中のリモート問合せを PostgreSQL の問合せにより関連づけた情報を全ての外部表スキャンで共有・監視することにより，問合せが実行不可能となる状況を検知する．

5.1.2 解消の方式

PostgreSQL が 1 件の問合せにおいて同時に実行する外部表スキャンは 1 件のみである．よって**条件（2）**により，問合せが実行不可能な状況では，$RQ_s(i)(i = 1, 2, ..., N)$ のいずれかに対応する外部表スキャンが $RQ_e(j)(j = 1, 2, ..., N)$ の結果取得を行うことができる．提案方式では，$RQ_e(j)$ の結果データサイズが最小となることが期待される $j' \in \{1, 2, ..., N\}$ を選択し，$RQ_e(j')$ の結果を全件取得し，問合せが実行不可能な状況を解消する．

5.2 データ構造

提案方式では各外部表スキャンのプライベート領域に加え，全外部表スキャンで実行中のリモート問合せの情報全件を共有する領域（以下，実行中リモート問合せ記憶領域）と開始要求中のリモート問合せの情報全件を共有する領域（以下，リモート問合せ開始キュー）を利用する．以下各領域について個別に説明する．

5.2.1 実行中リモート問合せ記憶領域

実行中のリモート問合せの情報 1 件に含まれる項目の一覧を**表 1** に示す．各外部表スキャンのプライベート領域には，表 1 の情報を格納する．表 1 の PG 問合せ ID は PostgreSQL の問合せを，ID と PG 問合せ ID の組は外部表スキャンを一意に識別する．

5.2.2 リモート問合せ開始キュー

リモート問合せの開始要求の情報 1 件に含まれる項目の一覧を**表 2** に示す．

5.3 アルゴリズム

FDW は実行中リモート問合せ記憶領域及びリモート問合せ開始キューに参照/更新アクセスをする場合に，必ず 1 つのサーバプロセスのみで参照/更新アクセス可能となるような排他ロックを取得する．以下，3 章で示した 4 つの

表 1: 実行中のリモート問合せの情報 1 件に含まれる項目一覧

Table 1 A list of data items for a running remote query

項目	内容
PG 問合せ ID	リモート問合せを作成する PostgreSQL のサーバプロセスのプロセス ID
ID	問合せの最適化においてリモート問合せが作成された順番
リモート問合せの実行情報	リモート問合せの結果取得及び終了及び結果レコードを PostgreSQL のレコードに変換するために必要な情報
結果データサイズ	結果データサイズの推定値
一時領域フラグ	結果データを一時領域に格納しているか否か．true/false（初期値）
一時領域の情報	一時領域へのレコード書込み，読込み，一時領域の削除に必要な情報

表 2: リモート問合せの開始要求の情報 1 件に含まれる項目一覧

Table 2 A list of data items for a starting remote query

項目	内容
PG 問合せ ID	リモート問合せを作成する PostgreSQL のサーバプロセスのプロセス ID
開始要求番号	リモート問合せ開始キューにおけるリモート問合せの開始要求の番号

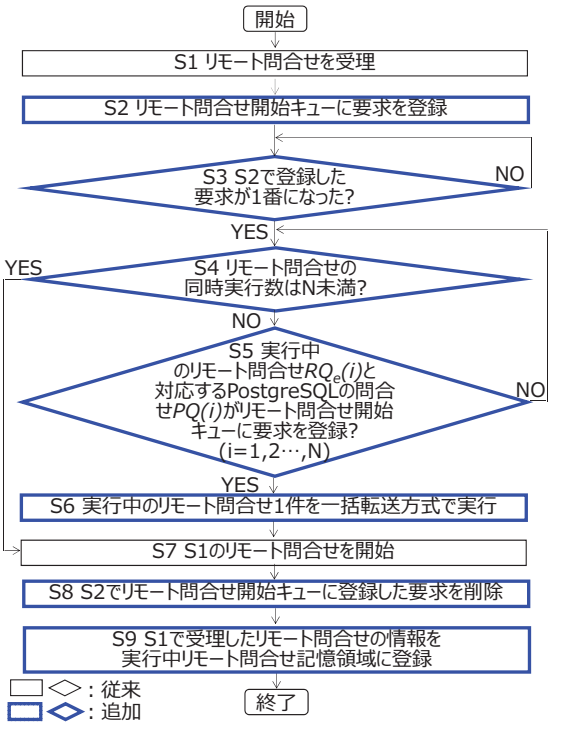

図 3: 提案方式におけるリモート問合せの開始

Fig. 3 The algorithm for starting a remote query in our proposal method

フェーズにおける FDW の動作を個別に説明する．

5.3.1 問合せ最適化

外部表スキャンの ID，PG 問合せ ID，結果データサイズを作成し，プライベート領域に格納する．結果データサイズは，結果レコードの数と平均レコードサイズを掛け合わせたものとする．

5.3.2 問合せ初期化

従来の動作からの変更はない．

5.3.3 問合せ実行

リモート問合せの開始動作と結果レコードのフェッチ動作に分けて説明する．以下の動作では，実行中リモート問合せ記憶領域またはリモート問合せ開始キューのいずれかを更新する際には，同時に一つのサーバプロセスのみ更新できるように排他制御する．

(1) リモート問合せの開始

以下**図 3** を基にリモート問合せの開始動作を説明する．S1 では従来と同様に PostgreSQL からリモート問合せを受理する．S2 ではリモート問合せ開始キューに要求を登録し，S3 では S2 で登録した要求が一番になるまで待つ．S4 では実行中リモート問合せ記憶領域において一時領域フラグが false になっている PG 問合せ ID と ID の組の件数が N 件（以下，$RQ_e(1)$〜$RQ_e(N)$）になっているか否かを判定する．S4 の結果が YES であれば，S7 に進み，NO であれば S5 に進む．S5 では，任意の $i \in \{1, 2, ..., N\}$ について $RQ_e(i)$ に対応する PG 問合せ ID がリモート問合せ開始キューに存在するか否かを判定する．S5 の結果が YES であれば，問合せが実行不可能な状況となっているため，その状況を解消するために S6 に進む．S5 の結果が NO であれば S4 に戻る．S6 では，実行中リモート問合せ記憶領域を参照し，$RQ_e(i)$ の結果データサイズが最小となる $i' \in \{1, 2, ..., N\}$ を選択する．続いて $RQ_e(i')$ の結果レコードを格納する一時領域を作成し，リモート問合せの実行情報を用いて一括転送方式で実行する．$RQ_e(i')$ の実行が完了したら，外部 DBMS に $RQ_e(i')$ の終了を要求する．続いて，実行中リモート問合せ記憶領域における $RQ_e(i')$ の一時領域フラグを true に変更し，一時領域の情報を作成した一時領域の情報とする．S6 により**条件（1）**が不成立となるため，問合せが実行不可能な状況が解消される．S7 では従来と同様に S1 で受理したリモート問合せを開始する．S8 では S2 でリモート問合せ開始キューに登録した要求を削除する．S9 では S1 で受理したリモート問合せの情報を実行中リモート問合せ記憶領域に登録する．

(2) リモート問合せの結果レコードフェッチ

初回のフェッチでは PostgreSQL から受理したリモート問合せの実行情報を実行中リモート問合せ記憶領域から取得し，結果レコードをフェッチする．フェッチするレコードがなくなった場合には，リモート問合せの終了を外部 DBMS に要求する．リモート問合せを終了したら，受理したリモート問合せの実行情報を実行中リモート問合せ記憶領域から削除する．2 回目以降のフェッチでは実行中リモート問合せ記憶領域において受理したリモート問合せの一時領域フラグが true か否かを判定する．一時領域フラグが true の場合は一時領域の情報をプライベート領域にコピーし，受理したリモート問合せの情報を実行中リモート問合せ記憶領域から削除する．一時領域フラグが false の場合は初回のフェッチと同様に動作する．

5.3.4 問合せ終了

従来の動作に加え一時領域の削除処理を実行する．

5.4 効果

逐次転送方式の問題であった問合せが実行不可能な状況を検知・解消できる．また，実際に問合せが実行不可能な状況が発生した場合のみ一括転送方式を選択するため，常に一括転送方式とするよりも一時ファイルの I/O データサイズを削減し処理速度を向上できる．

6. おわりに

本稿では，PostgreSQL の FDW にて同時実行可能な問合せ数が小さい外部 DBMS と連携する場合を想定し，問合せの実行可能性を保証しつつ，従来よりも処理速度の向上が可能なリモート問合せの同時実行数制御方式を提案した．今後は提案方式を実装し，性能評価により有効性を検証する予定である．

参考文献

[1] PostgreSQL，入手先 ⟨https://www.postgresql.org⟩（参照 2019-05-27）.
[2] ISO/IEC 9075-9, Information technology — Database languages — SQL — Part 9: Managementof External Data(SQL/MED), International Organization for Standardization(2008).
[3] 片山大河，廣瀬繁雄，望月翔平，金松基孝, PostgreSQL の NoSQL データベース GridDB との連携，第 80 回全国大会講演論文集 (2018).
[4] mongodb_fdw，入手先 ⟨https://github.com/EnterpriseDB/mongo_fdw⟩（参照 2019-08-07）.
[5] 立床雅司，山岸義徳，高山茂伸, PostgreSQL の外部データ管理における集約演算の push-down 方式, 第 77 回全国大会講演論文集 (2015).
[6] Mehta, M., DeWitt, D.J.: Dynamic Memory Allocation for Multiple-Query Workloads. Proc 19th VLDB Conf., 354-367(1993).
[7] PostgreSQL ExecMakeTableFunctionResult 入手先 ⟨https://github.com/postgres/postgres/blob/master/src/backend/executor/execSRF.c⟩（参照 2019-05-27）.
[8] Sheth, A.P. and Larson, J.A.: Federated Database Sysems for Managing Distributed, Heterogeneous, and Antonomous Databases, *ACM Computing Surveys*, Vol.22, No.3, pp.183-236(1990).