2C-05

PostgreSQLの外部データ管理における集約演算のpush-down方式

立床　雅司†　　山岸　義徳†　　高山　茂伸†

三菱電機株式会社　情報技術総合研究所†

1　はじめに

PostgreSQL[1]の Foreign Data Wrapper(FDW)を用いて外部データベースに集約演算を push-down する方式について述べる．本方式では，PostgreSQL に用意されているプランナフックを用いて，外部データベースへの問合せが生成され，PostgreSQL により最適化されたプラン木を辿り，集約演算を push-down する．電力データを用いた評価により，本方式の有効性を検証した．

2　背景

IoT(Internet of Things)に関連する技術の進展により，製造現場や社会インフラにおいて日々発生するセンサデータを長期間蓄積し，異常検知や生産効率改善に利用する試みが広まっている．センサデータは，時系列で生成されるデータであり，同一の系列をまとめて扱うことで，データ圧縮処理が効率化でき，蓄積する場合のストレージコストが削減できる．センサデータを異常検知等の分析に用いる場合は，アプリケーションからアドホックな問合せを柔軟に発行可能とするため，SQL 問合せ文のサポートが有効であると考えられる．我々は，自社開発の列指向データベース[2]である高速集計検索エンジン[3]と PostgreSQL を組み合わせ，センサデータ向けのデータベースとする試み[4]を行ってきた．

3　外部データ管理の課題

3.1　ユーザ定義関数を用いた外部データ管理

PostgreSQL には，ユーザ定義関数による拡張機能が備わっている．ユーザ定義関数は，引数として外部データベースへの問合せ文を受け取る．従来[4]は，ユーザ定義関数を用いた実装となっていた．外部データベースの機能をすべて利用できる反面，独自の問合せ構文は隠ぺいされず，SQL 問合せ文に埋め込む必要があった．

3.2　FDW を用いた外部データ管理

FDW は，PostgreSQL のプラン木から外部データベース向けの問合せ文を生成することで，独自問合せ構文の隠ぺいが可能となる．そのため，PostgreSQL の問合せ構文で表現できない外部データベースの機能は利用できない．また，PostgreSQL 9.2 では，外部データベースに push-down 可能な処理が Where 句の選択条件に限られる．図 3-1に示す集約演算では，すべてのデータを PostgreSQL に転送し，PostgreSQL 側で集約演算が処理されるため，PostgreSQL-外部データベース間のデータ転送がボトルネックとなることが想定される．独自の問合せ構文を隠ぺいしつつ，高速集計検索エンジンの性能を引き出すことが可能な外部データ管理が求められる．

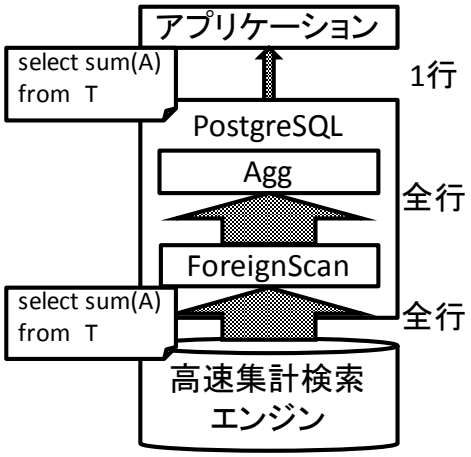

図 3-1 FDW の集約を含む問合せのプラン例

4　集約演算の push-down 方式

PostgreSQL に拡張手段の 1 つとして用意されているプランナフックを用いて，プランを書き換えることにより，集約演算の push-down を実現する．プランナフックは，PostgreSQL の起動時に組み込まれ，PostgreSQL が問合せを最適化する際に下記の順序で実行される．

(1) PostgreSQL のプランナを呼び出し，プラン木の最適化を行う．外部データへの問合せが含まれる場合は，外部データの問合せ文生成を行う．

(2) プランナフックにより集約演算を push-down する．集約演算の push-down は，集約演算の子のプランが外部データへの問合せの場合かつ集約演算に含まれる選択リストが外部データベースで実行可能な場合に限る．

5　評価

集約演算を push-down しない FDW と外部データベースを用いない場合との比較により本方式の有効性を評価した．

A Method of Push-down for Aggregation Operator on Foreign Data Management in PostgreSQL

†Masashi Tatedoko, Yoshinori Yamagishi, Shigenobu Takayama

Information Technology R&D Center, Mitsubishi Electric

5.1 評価データ

評価には，表 5-1に示す電力データを用いた．データは，高速集計検索エンジンと PostgreSQL にロードした．

表 5-1 評価データ

センサ数	123 センサ (32 ビット整数)
データ数	10,781,856 行
論理データサイズ	4.94GB
高速集計検索エンジン格納時のサイズ	0.24GB
PostgreSQL 格納時のサイズ	11.7GB

5.2 評価環境

評価は，図 5-1に示す評価環境にて実施した．評価環境は，表 5-2に示す評価用 PC サーバ内で閉じたローカル環境とした．

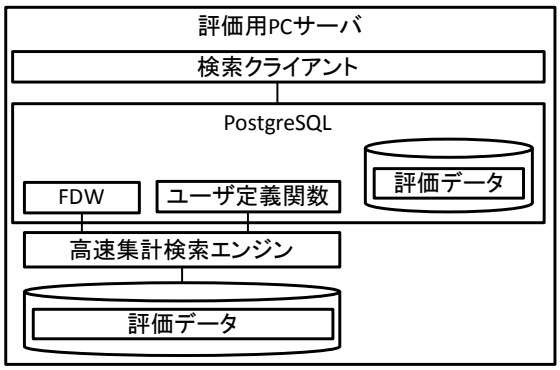

図 5-1 評価環境

表 5-2 評価用 PC サーバ

CPU	Intel® Core i7 2600 3.4GHz core x 4 (L2 cache 256KB x 4)
Memory	8 GB
Storage	SATA2，1TB，7,200rpm
OS	Windows 7 (x64) Professional
RDBMS	PostgreSQL 9.2.7

5.3 評価条件

表 5-3に示す評価対象手法に対して，下記の評価条件にて評価を実施した．ユーザ定義関数による集約は，従来[4]用いていた手法であり，同等以上の性能となることを確認するため対象に加えた．

表 5-3 評価対象手法

評価対象手法	データ格納先
FDW(集約演算 push-down あり)	高速集計検索エンジン
FDW(集約演算 push-down なし)	
ユーザ定義関数による集約	
PostgreSQL による集約	PostgreSQL

評価は，特定の期間を指定し，123 のセンサすべての平均値を求める問合せを用いて行った．期間は，100～1000 万件の間を 100 万件間隔で変化させた．問い合わせは連続で 11 回実行するものとし，初めの 1 回目を除いた 10 回分の平均値を評価結果とした．

5.4 評価結果と考察

FDW(集約演算 push-down あり)がユーザ定義関数による集約とほぼ同様の結果となった．入力行数 1000 万行の場合，FDW(集約演算 push-down あり)は 3.3 秒，FDW(集約演算 push-down なし)は 563.1 秒であった．評価の結果を図 5-2に示す．

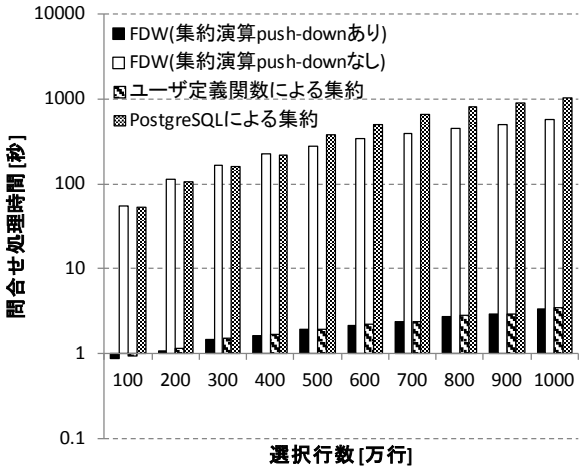

図 5-2 評価結果

FDW(集約演算 push-down なし)は，PostgreSQL による集約と比較して大きな差はなかった．タプルの PostgreSQL の内部データ形式への変換処理のオーバーヘッド，FDW のフェッチ処理と外部データへの問合せが直列に動作していること等が原因として考えられる．

本方式は，集約演算に限らず適用可能である．たとえば，PostgreSQL のテーブルと外部データベースのテーブルを結合する場合，Merge Join となる．ソートを push-down することで，FDW で取得したタプルを中間テーブルに出力してソートする処理を省略することが考えられる．

6 まとめ

PostgreSQL の FDW を用いた外部データベースへの集約演算の push-down する方式を実装し，その有効性を検証した．今後は，その他の演算についての push-down を検討する．

参考文献

[1] PostgreSQL，http://www.postgresql.org/.

[2] Daniel Abadi et al. , The Design and Implementation of Modern Column-Oriented Database Systems，Foundations and Trends in Databases, 5(3): 197-280.

[3] 山岸 義徳ほか：高速集計検索エンジンとセンサデータベースへの応用，三菱電機技報，Vol.83，No.12, pp.11-14(2009）.

[4] 竹内 丈志ほか：オープンソース DBMS 拡張によるセンサデータベースの実現，情報処理学会第 74 回全国大会講演論文集，pp.545-547(2012).