


Project Tsurugiユーザー会
次世代データベース全体像の
ご紹介

NEC 並木

2021年10月11日

はじめに

- 本セッションでは前述の「コア/周辺系」の実装の全体像を説明する

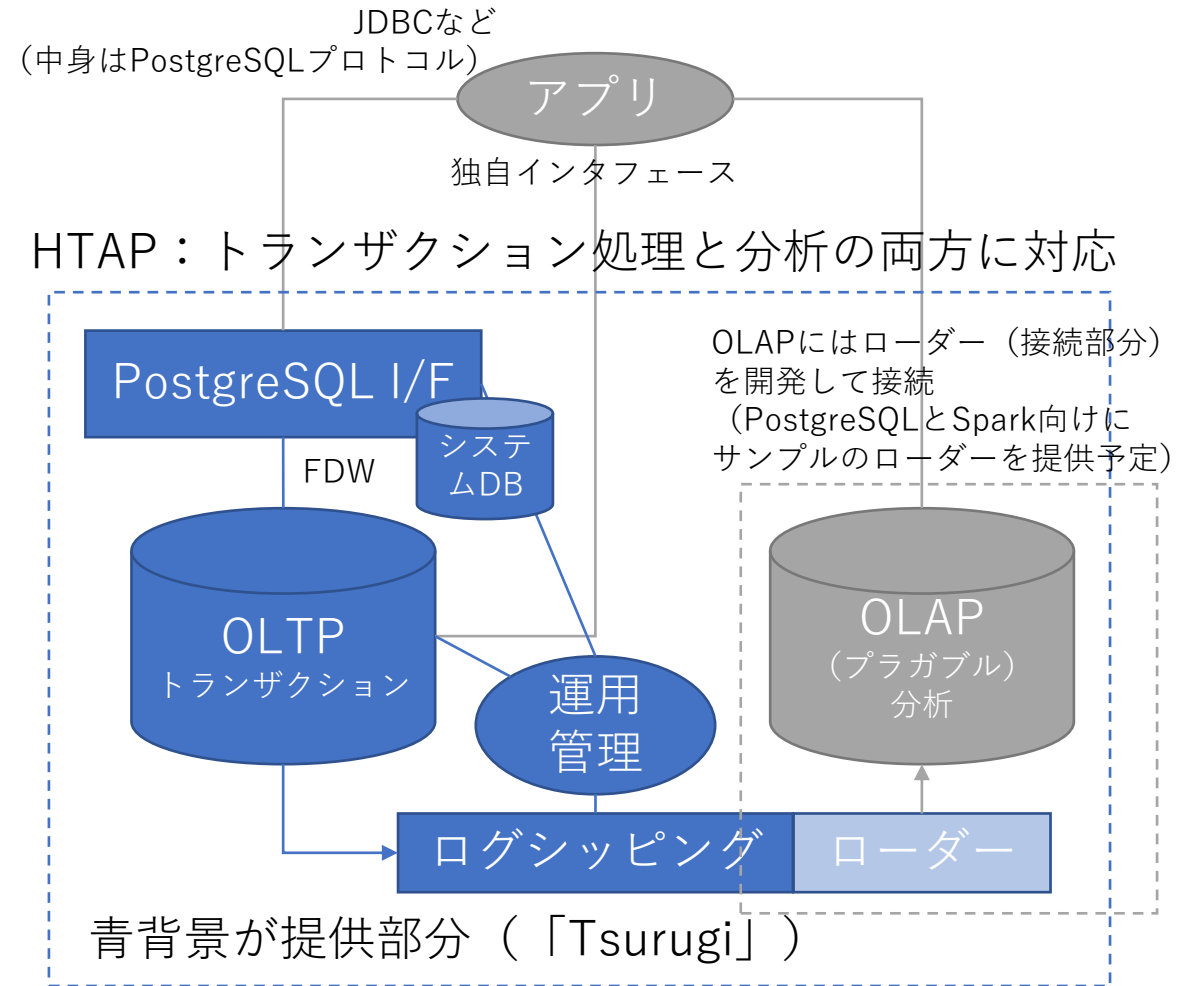


アウトプット予定(予定は予定で決定ではない)

- アウトプット予定
 - 2023/2/EがNEDOプロの終了なので、その時点で強制開示は確定
- コア/周辺系
 - 実装環境一式
 - ソースコード
 - DB本体
 - 運用ツール (既存PostgreSQL接続環境)
 - 必要ドキュメント (たぶん最低限)
 - コア/周辺に関する公表された論文
 - 論文とそれに関する実装
- 現時点で商用サポートは行う前提

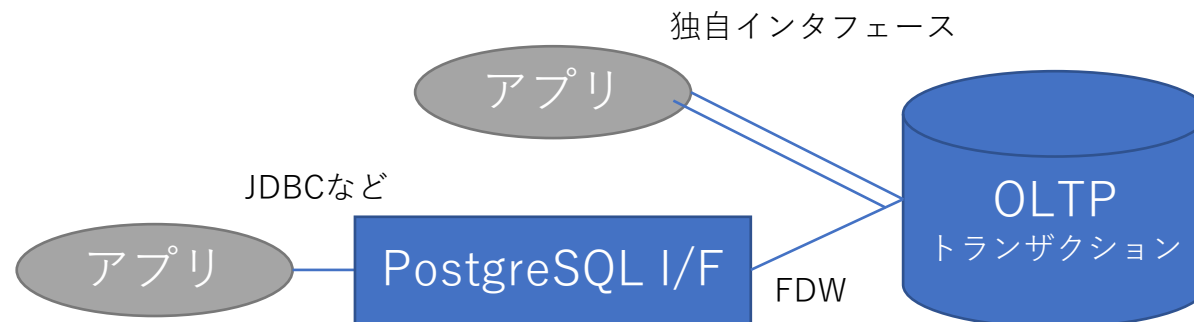
全体像

- 最新のハードウェアに最適な設計を行い、トランザクション処理を超高速かつserializableに実現
 - 詳細はノーチラス・テクノロジーズのセッションにて
- 整合性のある断面を高頻度にOLAPへ反映可能（リリース時期調整中）
 - 詳細は東京工業大学のセッションにて



アプリケーションとのインタフェース

- 3種類あり（使い分けは次のセッションで）
- PostgreSQL経由（ただし性能面は不利）
 - PostgreSQLが提供するインタフェースをそのまま利用可能：Psql、ecpg、libpq、ストアドプロシージャ、JDBC、ODBC、…
 - FDWでTsurugiと連携
- OLTP直結（2種類）
 - 高速な独自のインタフェース（別セッションで紹介）



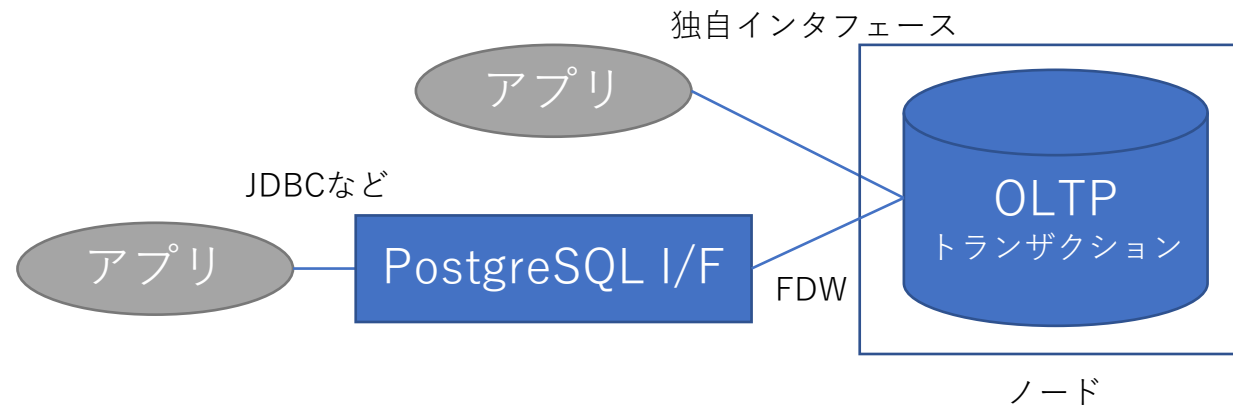
PostgreSQL I/FからのSQLの使用

- TCL
 - SAVAPPOINTは現状なし
 - アプリケーションはCOMMITが失敗したらリトライするような制御が必要
 - COMMITの失敗は特別なことではない
 - トランザクションの隔離レベルはSERIALIZABLE前提（充分早い）
- DCL、DDL
 - そのまま
- DML
 - ロック方式ではないのでSELECT FOR UPDATEは効果がない

システム構成 (1)

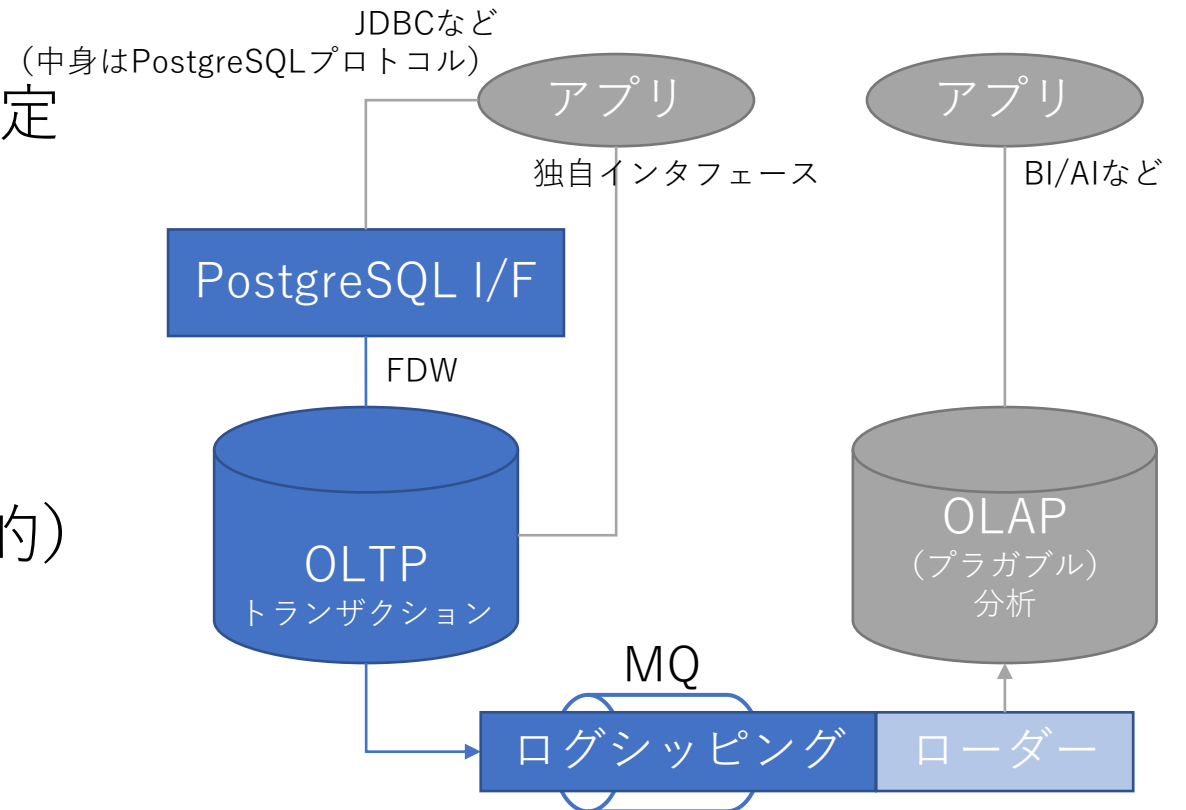
- OLTP

- 現時点で単一ノード構成が基本（レプリケーションに関しては2回目のセッションで）
- 無改造のPostgreSQLとFDW（Foreign Data Wrapper）で連携



システム構成 (2)

- OLTP (トランザクション処理) の更新をOLAP (分析) に反映
→ ログシッピングで実現
 - PostgreSQL、Sparkとの連携向けにサンプル実装を提供予定
 - ロード一部分を開発することで各種データストアをOLAPとして連携が可能
 - MQを利用
 - 不揮発性メモリを使ったマシン内OLTP-OLAP連携も検討中 (実験的)



まとめ & おわりに

- 次世代のハードウェアを活用した高性能データベースを、オープンソースで公開します
- PostgreSQLに近い運用性をもたせています
 - ただしトランザクション処理の挙動は異なるなど、完全な互換性はありません
- 高速化の恩恵を得るため、アプリケーション側も新しいスタイルへの進化をお願いします