PGECons
PostgreSQL Enterprise Consortium

2024年度CR部会活動報告

PostgreSQLの適用領域拡大に向けた技術的課題改善の動向

PostgreSQL エンタープライズ・コンソーシアム
CR部会
2025.5.30

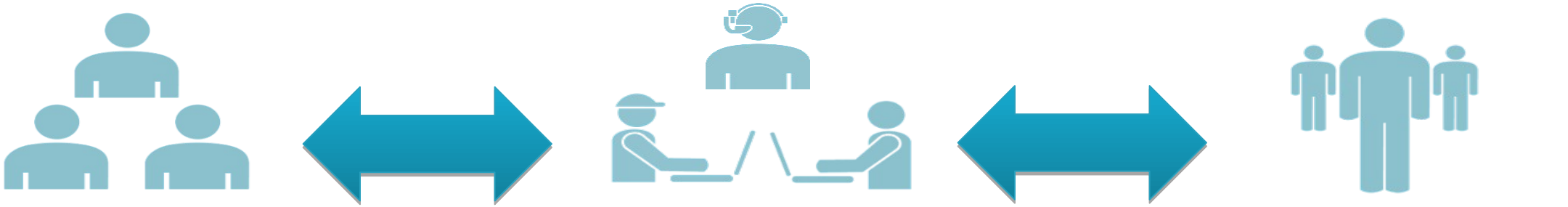
アジェンダ

- CR部会のご紹介
  - CR部会の目的、参加メンバ
- 2024年度の取り組みのご紹介
  - 課題の解決状況およびコミュニティでの議論を調査
- おわりに
  - 2025年度に向けて

CR部会のご紹介

CR（Community Relations）部会の目的

- PostgreSQL 開発コミュニティへのフィードバック
  - エンタープライズ領域への PostgreSQL の適用に向けて、開発コミュニティに技術的課題をフィードバック



抱えている課題が解決される！

ユーザの声が届く！

CR部会 2024年度参加メンバ

- 株式会社SRA OSS【主査】
- NECソリューションイノベータ株式会社
- 日本電信電話株式会社
- 富士通株式会社

（企業名50音順・敬略称）

2024年度の取り組みのご紹介

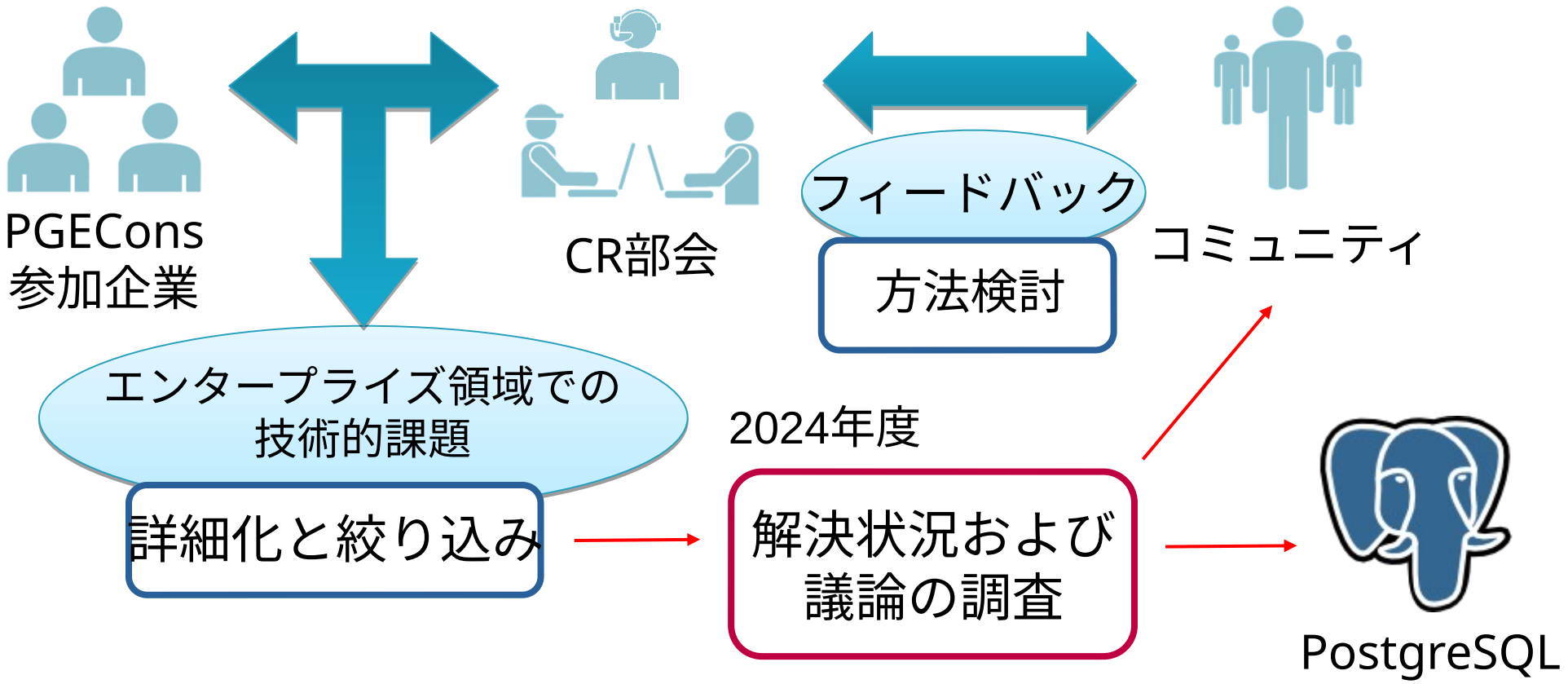
2024年度の取り組み内容

1. 課題の解決状況およびコミュニティにおける議論の調査

2. PostgreSQLコミュニティの行動規範（Code of Conduct, CoC）の原文が改訂されたことを受け、日本語訳を更新

2024年度の活動概要

- 課題の解決状況およびコミュニティにおける議論を調査



課題の解決状況およびコミュニティでの議論の調査

- 「基幹領域への適用におけるPostgreSQLの抱える課題」
  - エンタープライズ用途でPostgreSQL を使う際の課題として、PGECons 設立当初（2012年）に参加企業から集められた
  - CR部会で課題の詳細化と絞り込みを実施

- 以下の課題について、現在の解決状況およびコミュニティにおける議論を調査
  - 「実行計画が制御できない」
  - 「COPY機能の改善」
  - 「更新系の性能が悪い」
  - 「負荷分散方式が確立されていない」
  - 「監査手法が確立されていない」
  - 「障害解析情報が少ない」

実行計画の制御：課題と解決策

- PostgreSQLの実行計画の特徴と課題
  - SQL実行時のDBのデータ分布（統計情報）に従って最適な実行計画を作成
    - データによって実行計画は異なることがある
      ⇒　検証環境では業務システムのクエリ性能の検討が難しい
    - 不適切な実行計画が作成されると突如性能が低下することがある
    - 真に最適な実行計画が選択されないこともある　（※ただし、そういう事例は比較的稀だとみられる）

- 外部ツールによる解決策
  - pg_hint_plan
    クエリに「ヒント句」を付与することで特定の実行計画が選択されやすくなるように誘導する機能
  - pg_dbms_stats
    ある時点での統計情報を保存しておき、クエリの実行時にこれを参照させることで、クエリの実行計画を「固定」する機能

実行計画の制御：外部ツールの特徴と利用状況

- pg_hint_plan
  - SQLの中に実現したい実行計画を指示する「ヒント句」を埋め込む
    - 結合方式や結合順序、表アクセス（index, seq scan）の方法などを指示できる
  - 比較的利用されている模様
    - OSS コミュニティで定常的に維持管理されており、利用ノウハウ情報も豊富
    - AWS や Google Cloud 上の PostgreSQL でも使用可能

- pg_dbms_stats
  - 統計情報の固定化を通じて実行計画を一定に保つ
    ⇒　運用中に突如性能が低下するトラブルを防ぐ
  - 業務システムの統計情報を使うことで、検証システムで業務システムの実行計画を再現する
    （※逆に、検証システムで確認済みの統計情報を業務システムに適用することもできる）
  - pg_hint_planよりも利用されていない模様
    - PostgreSQL ver. 15以降に対応できていない

実行計画の制御：PostgreSQL コミュニティにおける議論（１）

- 拡張モジュールによる実行計画制御を、本体側から支援する機能の提案

  (allowing extensions to control planner behavior, pgsql-hackers, 2024-08-26 ～ 2024-10-23)

  - テーブルのスキャン方法、結合方法の戦略を拡張モジュールから直接変更できる仕組みを追加
    - pg_hint_plan では enable_seqscan などの GUC パラメータ経由で制御していた

- ただし、「スキャン方法・結合方法を拡張モジュールに指示する方法」については議論がまとまっていない
  - 例）クエリコメントに「ヒント句」を埋め込む場合、拡張機能はどのようにしてコメントにアクセスするか
    - 構文解析や構文の拡張が必要？
  - 提案されたパッチのデモでは「テーブルの別名」でスキャン方法、結合方法を指示
  - ヒント句に対する反対意見もある（※）

※参考：https://wiki.postgresql.org/wiki/OptimizerHintsDiscussion

実行計画の制御：PostgreSQL コミュニティにおける議論（2）

- 「ヒント」による実行計画制御に対する反対意見に関する議論

  (allowing extensions to control planner behavior, pgsql-hackers, 2024-08-26 ～ 2024-10-23)

  - クエリプランナーそのものを改善すべき？
    - 以前はそう思っていたが、もうそうは思わない。15 年以上 PostgreSQL に取り組んできたが完璧なクエリプランナーにはあまり近づいていない。(Robert Haas)
  - ヒント句を埋め込むためクエリテキストの変更が必要？
    - これは確かに面倒ではあるが、ヒント句が役に立たないということではない
    - テーブルにヒント情報を格納するなど、コメント以外の選択肢もある。
  - 私たちはヒントに対して非常に否定的になり、それについて合理的な議論をほとんど行っていない？

- 少なくとも、実行計画を制御するツールの有用性・必要性はコミュニティでも認識されている

実行計画の制御：（参考）統計情報のインポート/エキスポート （１）

- PostgreSQL18の新機能：統計情報のインポート・エクスポート機能
  - pg_upgrade
    - PostgreSQL17以前：移行時に統計情報が移行されない

      ⇒ 移行後にANALYZEが必要になるが、データ量が多い場合に時間がかかるのが課題
    - PostgreSQL18：統計情報も移行されるようになった
  - pg_dump
    - PostgreSQL18：統計情報もバックアップ可能になった

実行計画の制御：（参考）統計情報のインポート/エキスポート （2）

- CR部会での取り組み
  - コミュニティへの質問
    - 本番環境から pg_dumpでエクスポートした統計情報を開発環境にインポートして性能検証ができるのでは？
  - コミュニティからの回答
    - 統計情報以外の情報（テーブルの実サイズやインデックスのoidの順番）に依存してプランが決定されるため、この方法ではプランが本番環境と同じになることは保証されない

→PostgreSQLドキュメントの注意事項に記載することを提案中

COPY機能の改善

- もともとは「初期ロードが遅い」という課題だったが、CR部会ではCOPY 機能の改善の議論に注力
- 最近のCOPY 機能の改善
  - psql \copy の性能改善（PostgreSQL 15）
    - メッセージの出力を１行毎から8KB 単位に
  - COPY FROM （テーブルへのデータロード）を同時実行したときの性能改善（PostgeSQL 16）
    - テーブルにページを追加する処理の改善
  - COPY FROM で不正行を無視して処理を継続するオプション「ON_ERROR ignore」が追加（PostgreSQL 17）
    - 不正行の数を制限する REJECT_LIMIT オプション（PostgreSQL 18）
  - COPY TO （テーブルからのデータ抽出）高速化（PostgreSQL 17)
    - サーバ・クライアント間の通信で不要な内部データコピーを削減
  - 値が不正で COPY できなかった行の情報をテーブルに書き出す / 値をNULLに置き換えて挿入（議論中）
  - COPY の形式（text, csv, binary など）をカスタマイズ可能に（議論中）

更新系の性能 (1)

- 更新系の性能が悪い
  - 課題設定時（2012）に、どのようなケースの性能不足を指してしたのか不明
  - 具体的なケースではなく、一般的に論じられているPostgreSQL の性能問題への懸念？
    - PostgreSQLの MVCC は更新ごとにテーブルに新しい行を追加する方式
    - その際に全てのインデックス更新が必要となり無駄な書き込みが多くなる

      （= index write amplification）

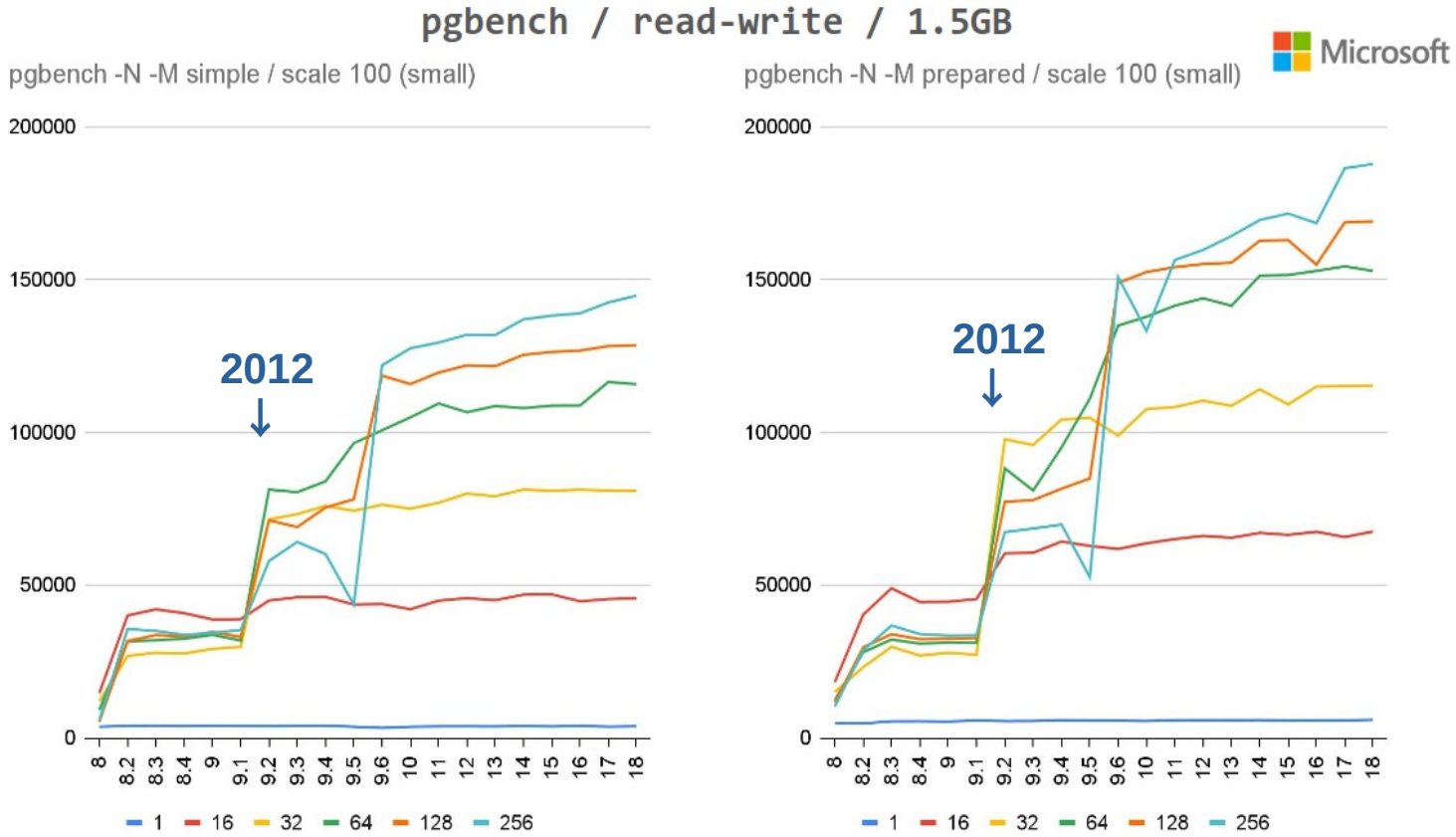
      - 例）Why Uber Engineering Switched from Postgres to MySQL (Uber Blob, 2016)
        https://www.uber.com/en-JP/blog/postgres-to-mysql-migration/

更新系の性能 (2)

- 更新系の性能が悪い
  - Index write amplification に対するコミュニティからの反応
    - HOT（Heap-Only Tuples）最適化により、非インデックス列の更新ではインデックス更新は不要。
      - Thoughts on Uber’s List of Postgres Limitations (Simon Riggs, EDB, 2016)
        https://www.enterprisedb.com/blog/thoughts-ubers-list-postgres-limitations
      - A PostgreSQL Response to Uber (Christophe Pettus, PostgreSQL Experts, Inc., 2017)
        https://thebuild.com/presentations/uber-perconalive-2017.pdf
    - しかし、他のRDBMSの設計に比べ、性能上の問題になり得るのは確か
      - １つのテーブルに多数のインデックスがある場合、など
      - Why we lost Uber as a user (pgsql-hackers, 2016）
      - The Part of PostgreSQL We Hate the Most (Andy Pavlo, 2023)
        https://www.cs.cmu.edu/~pavlo/blog/2023/04/the-part-of-postgresql-we-hate-the-most.htm
    - 過去に改善案の提案もあった（以下はいずれも活動停止）
      - WARM (Write Amplification Reduction Method)
      - zheap

更新系の性能 (3)

- 更新系の性能自体は改善されている
  - “Performance Archaeology”(PGConf.EU 2024, Tomas Vondra, Microsoft)



  - PGECons WG1 による定点観測でも改善を確認

https://www.postgresql.eu/events/pgconfeu2024/sessions/session/5585-performance-archaeology/

負荷分散方式

- 利用されている負荷分散技術
  - 参照負荷分散（レプリケーション + ロードバランサ）
    - Pgpool-II などで参照・更新クエリの振り分けが必要
  - 更新負荷分散（シャーディング）
    - PostgreSQL 機能のFDW（外部データラッパ）とパーティショニングの利用
- 負荷分散に関する最近の議論
  - libpq のロードバランス機能（更新と参照の振り分けはできない）(PostgreSQL 16)
  - FDW ベースのシャーディング
    - postgres_fdw の接続状態チェック機能の改善（PostgreSQL 18）
    - 集約の一部分を外部テーブルにさせる機能（議論中）
    - 分散トランザクションにはまだ非対応（２相コミットに対応した postgres_fdw_plus は存在する）
  - 論理レプリケーションの改善に関する議論
    - 衝突の検出・回避、シーケンスレプリケーション、DDLレプリケーション

監査手法

- pgAudit の使用がメジャーな手段
  - PGECons WG3でも紹介済（「PostgreSQLセキュリティガイド」2015年度）
  - PostgreSQL の log_statement では不足している機能を提供
    - 参照されるオブジェクト情報の取得
    - 対象の細かい指定でログ出力量を抑える
- pgAudit で対応できないもの
  - 運用ログと監査ログの分割
    - rsyslog や pgAudit Log to File などで対応
    - （関連）PostgreSQL 本体でも log_duration のログを別ファイルに分離する機能の提案はあり
  - スーパユーザの適切な監査
    - スーパユーザのログインや、一般ユーザのスーパユーザ昇格を制限する必要がある
    - pgaudit では、スーパユーザへの昇格をログに記録できる set_user というツールも提供
- PostgeSQL 本体 では監査を目的とした機能開発の議論はあまり行われていない

障害解析情報

- 障害解析時にどのような情報があるとよいか？
  - サポート側の視点からPGEcons 参加企業にヒアリングを実施（2018年）
- 例）「実行中のクエリのプランを知る機能」
  - 関連する機能が現在開発コミュニティで議論中
    - 実行中クエリの実行計画をログに出力する機能
    - EXPLAIN ANALYZE で実行中のクエリのプランをリアルタイムに確認できるビュー
- pg_stat_activity などのシステムビューで取得できる情報は増えてきている
  - pg_stat_activity ビューに待機イベント情報（ wait_event_type, wait_event ）追加 (PostgreSQL 9.6)
  - pg_stat_progress_... ビューによるVACUUM, CLUSTER, ANALYZE, CREATE INDEX, COPY などの進捗状況の確認 (PostgreSQL 9.6〜)
  - pg_backend_memory_context : バックエンドプロセスのメモリ使用状況の確認 (PostgreSQL 14) 、... etc
- 運用やサポートの現場においてどのような情報が求められているかについては、改めて調査が求められる

2024年度の取り組み内容

1. 課題の解決状況およびコミュニティにおける議論の調査

2. PostgreSQLコミュニティの行動規範（Code of Conduct, CoC）の原文が改訂されたことを受け、日本語訳を更新

Code of Conduct (CoC)

- 団体やコミュニティのメンバーが従うべき「行動規範」を定めたもの
- PostgreSQL の CoC
  - 2018年9月に制定
    - 正文は英語。各言語に翻訳された版が公開されている。
    - CR部会は日本語への翻訳で貢献
  - 主な内容
    - 参加者と期待される行動、行動規範委員会による苦情の対応、報復の禁止
- 英語版が改訂されたことを受け、日本語版の改訂を実施
  - ライセンス情報の追加、および委員の解任に関する一文の修正があったので、日本語版の改定を提案

おわりに

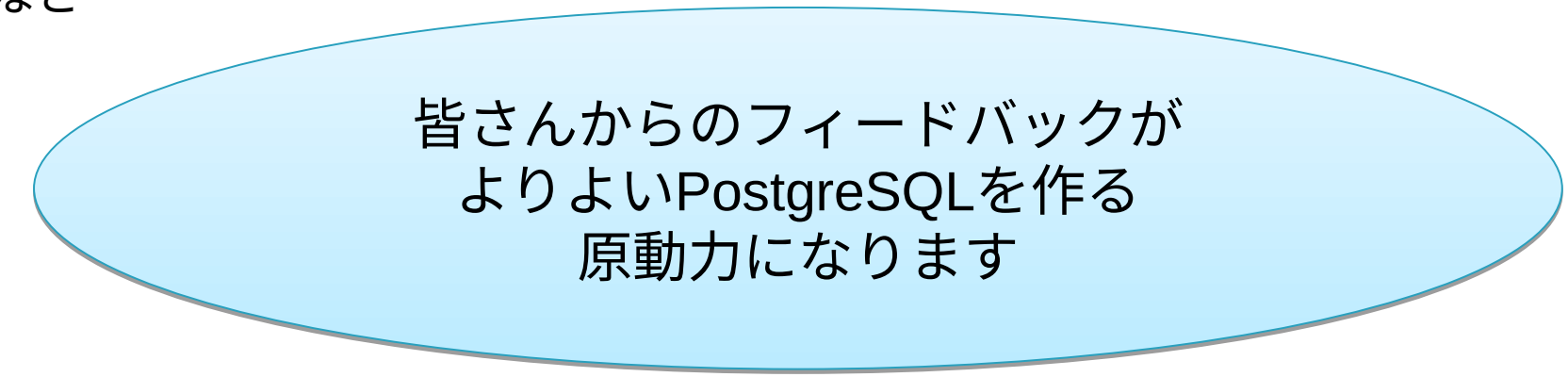
まとめ

- CR 部会の目的
  - エンタープライズ領域へのPostgreSQLの適用に向けて、開発コミュニティに技術的課題をフィードバック
- 課題の解決状況およびコミュニティにおける議論を調査
  - 実行計画の制御、COPY機能の改善、更新系の性能、負荷分散方式、監査手法、障害解析情報
  - 一部課題は残しつつも、改善や議論は続いている
  - 実行計画制御へのアプローチの議論など、新しい流れもある
- 今後
  - 現状の課題解決状況および具体的な課題の調査継続
  - フィードバックの方法の検討

皆様へのお願い

- 今年度（2025年度）も課題解決状況の調査を継続しながら、今後のフィードバックについての検討をしていく予定です。
  - 是非、アンケートに PostgreSQL に皆様が感じている課題についてご意見をお寄せくだ

    例えば、
    「性能性能が問題になっている具体的な状況」
    「障害やトラブルの解析時に、このような情報や機能があるとうれしい」
    など



- 一緒にフィードバック活動しませんか？

PGECons

PostgreSQL Enterprise Consortium