OSS-DB PostgreSQL に付加価値機能を提供する pgpool-II、その機能と導入のメリットについて

SRA OSS, Inc. 日本支社
pgpool-II 開発者
北川 俊広

SRA OSS,INC.

pgpool-IIとは

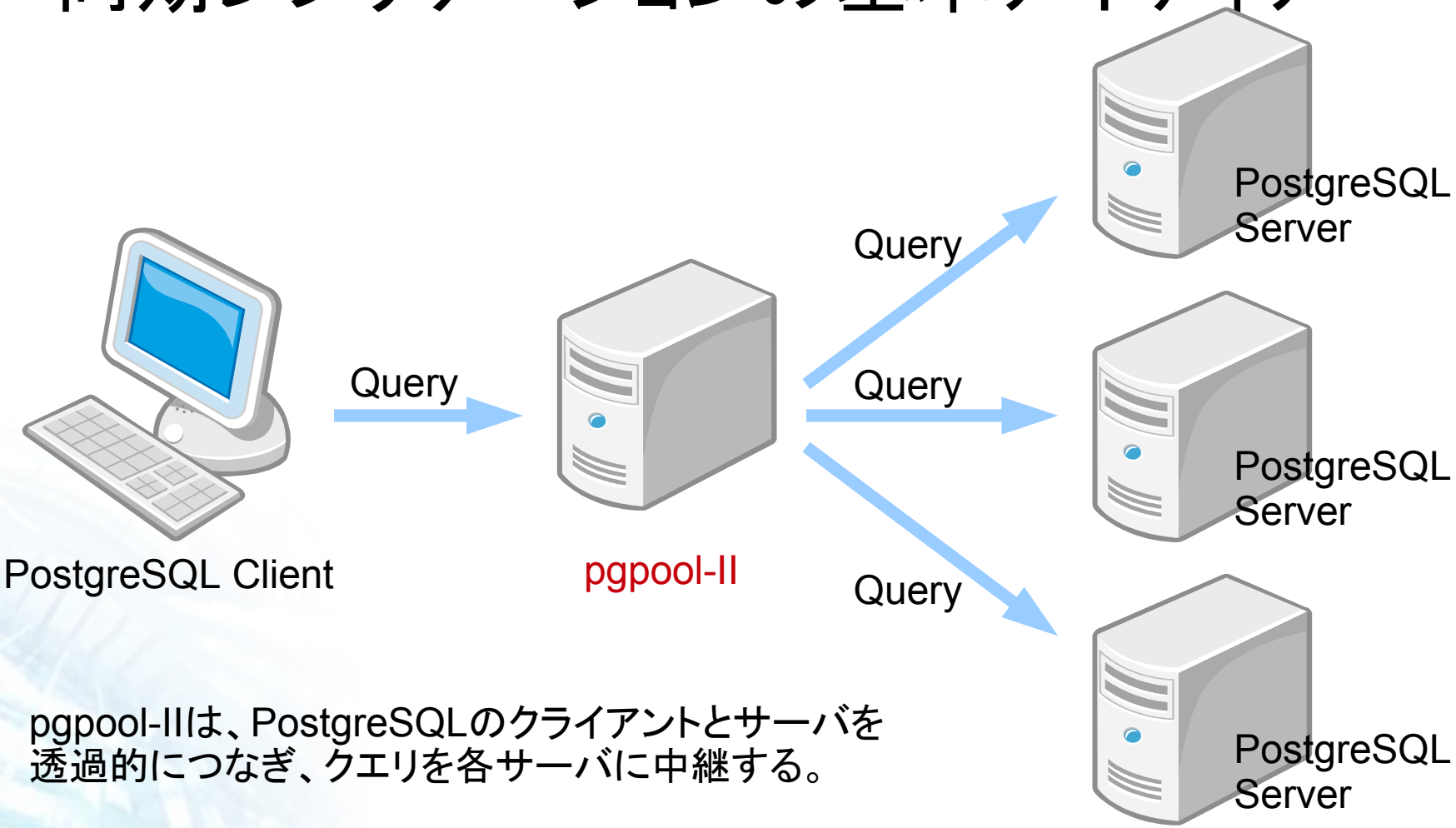
- PostgreSQL専用のクラスタ管理ツールの一つ
- オープンソースソフトウェア(BSDライセンス)
  - pgpool Global Development Groupが開発
- 多彩な機能
  - 同期レプリケーション、ロードバランス、
    自動フェイルオーバー、コネクションプーリングなど
  - 他のレプリケーションツールとの連携
    - Streaming Replication, Warm Standby, Slony-I
- Webベースのpgpool-II管理ツール
  - pgpoolAdmin

選択できるレプリケーション方式

- レプリケーション方式は選択可能
  - pgpool-IIの同期レプリケーション機能
  - 他のレプリケーションツールを利用
    - Streaming Replication, Warm Standby, Slony-I, etc.
- レプリケーション方式の比較

	pgpool-II	Streaming Replication	Warm Standby	Slony-I
クエリ制約	△	◎	◎	△
レプリケーション遅延	◎	○	△	△
レプリケーション負荷	△	◎	◎	○
ロードバランス	◎	○	×	○

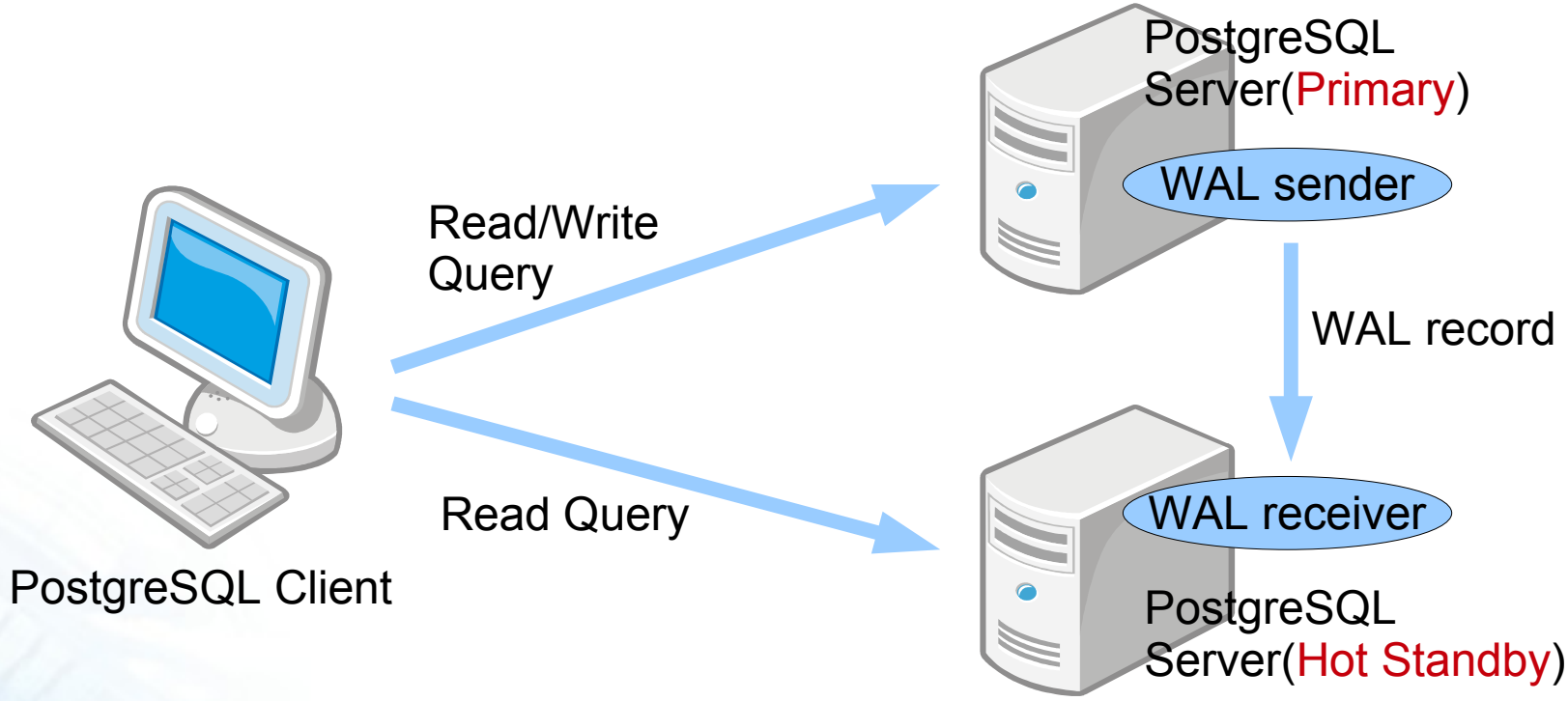
pgpool-IIによる 同期レプリケーションの基本アイディア



pgpool-IIは、PostgreSQLのクライアントとサーバを
透過的につなぎ、クエリを各サーバに中継する。

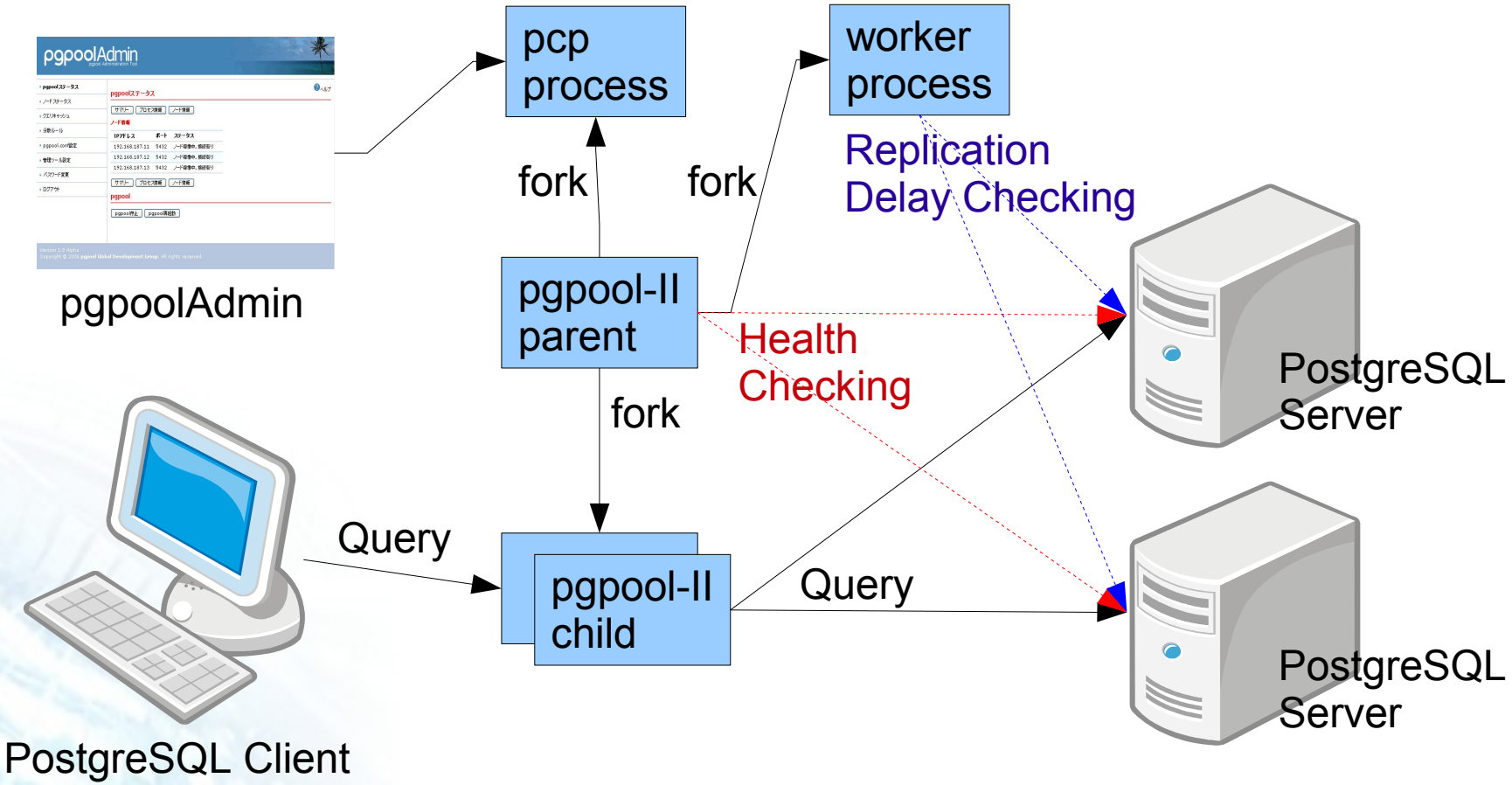
SRA OSS, INC.

PostgreSQLのStreaming Replication/Hot Standby機能



PostgreSQLのStreaming Replication機能は、ログ先行書き込み(WAL)のレコードをスタンバイサーバに転送し、それを常に適用していくことでレプリケーションを実現する。Hot Standbyサーバでは参照クエリのみ実行可能。

pgpool-IIのプロセス構成



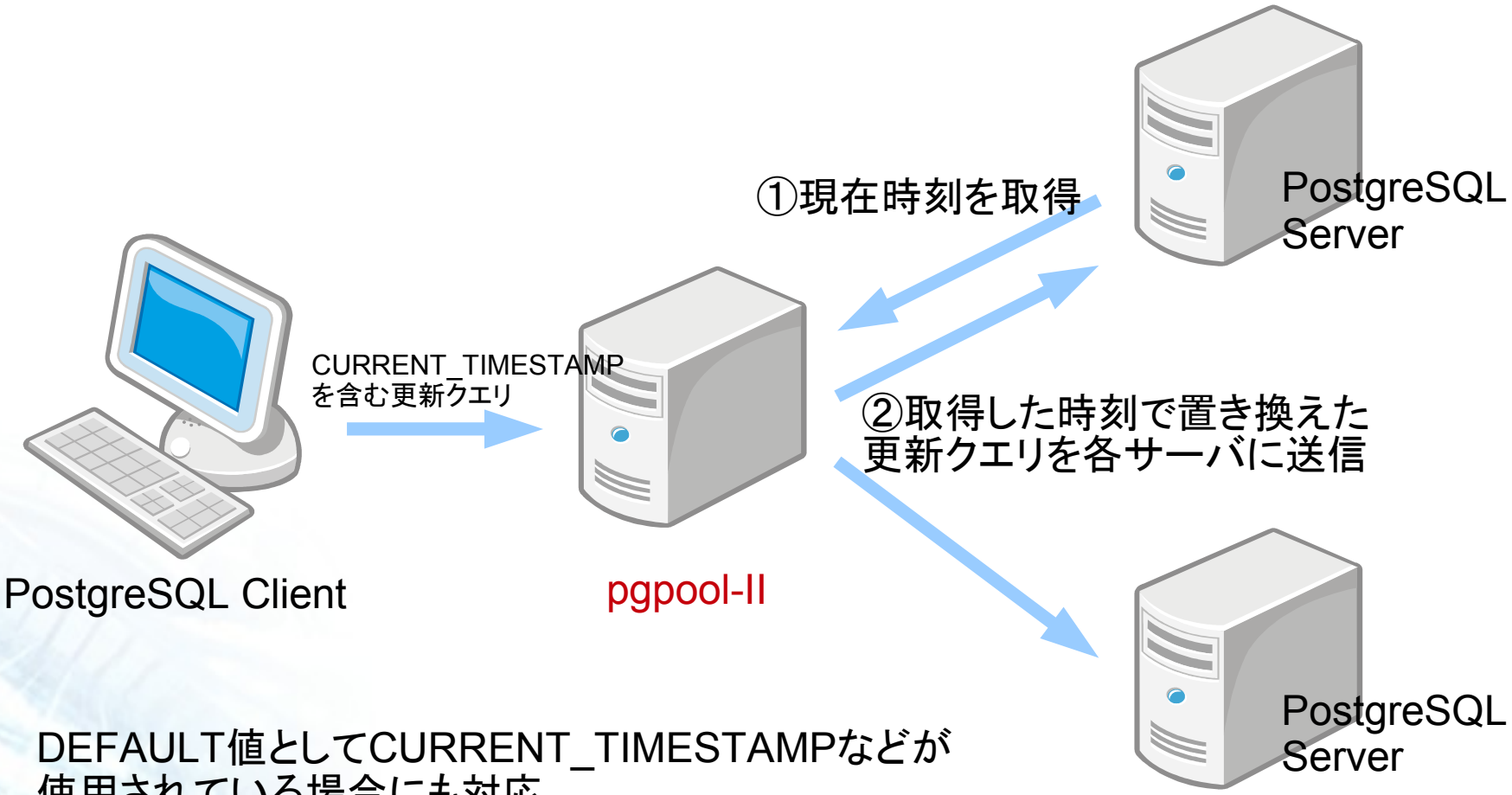
SRA OSS, INC.

クエリベースレプリケーションの問題と解決策

- サーバごとに異なる値が返る関数をクエリに含む場合、サーバ間のデータの整合性が崩れる。
  - 具体例
    - 現在時刻、シーケンス値、乱数を求める関数など

      CURRENT_TIMESTAMP, nextval(), random(), ・・・
  - 解決策
    - 1つのサーバでそのような関数を実行して値を取得し、その値を更新クエリに埋め込んで各サーバへ送る。

時刻データへの対応

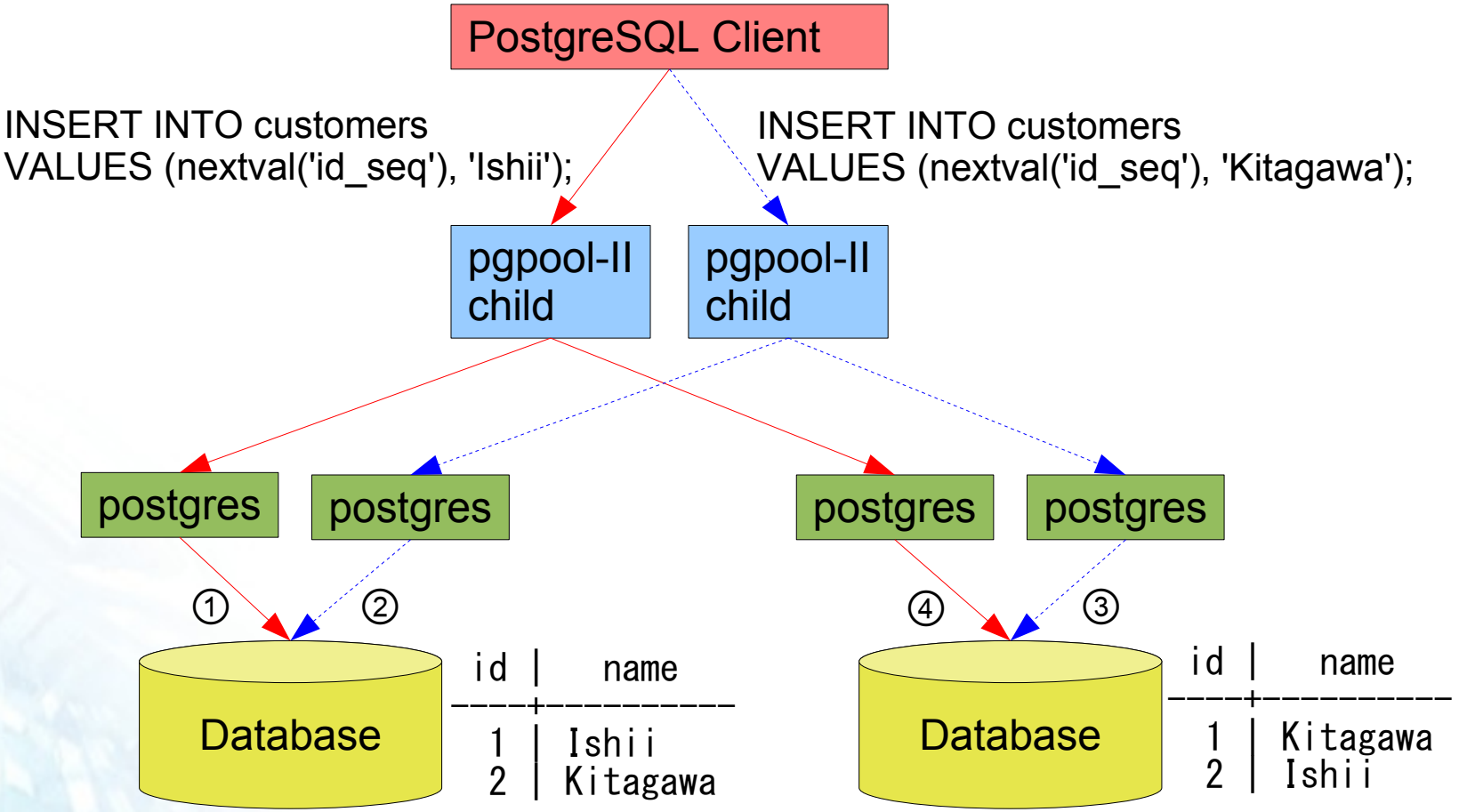


DEFAULT値としてCURRENT_TIMESTAMPなどが
使用されている場合にも対応。

SRA OSS, INC.

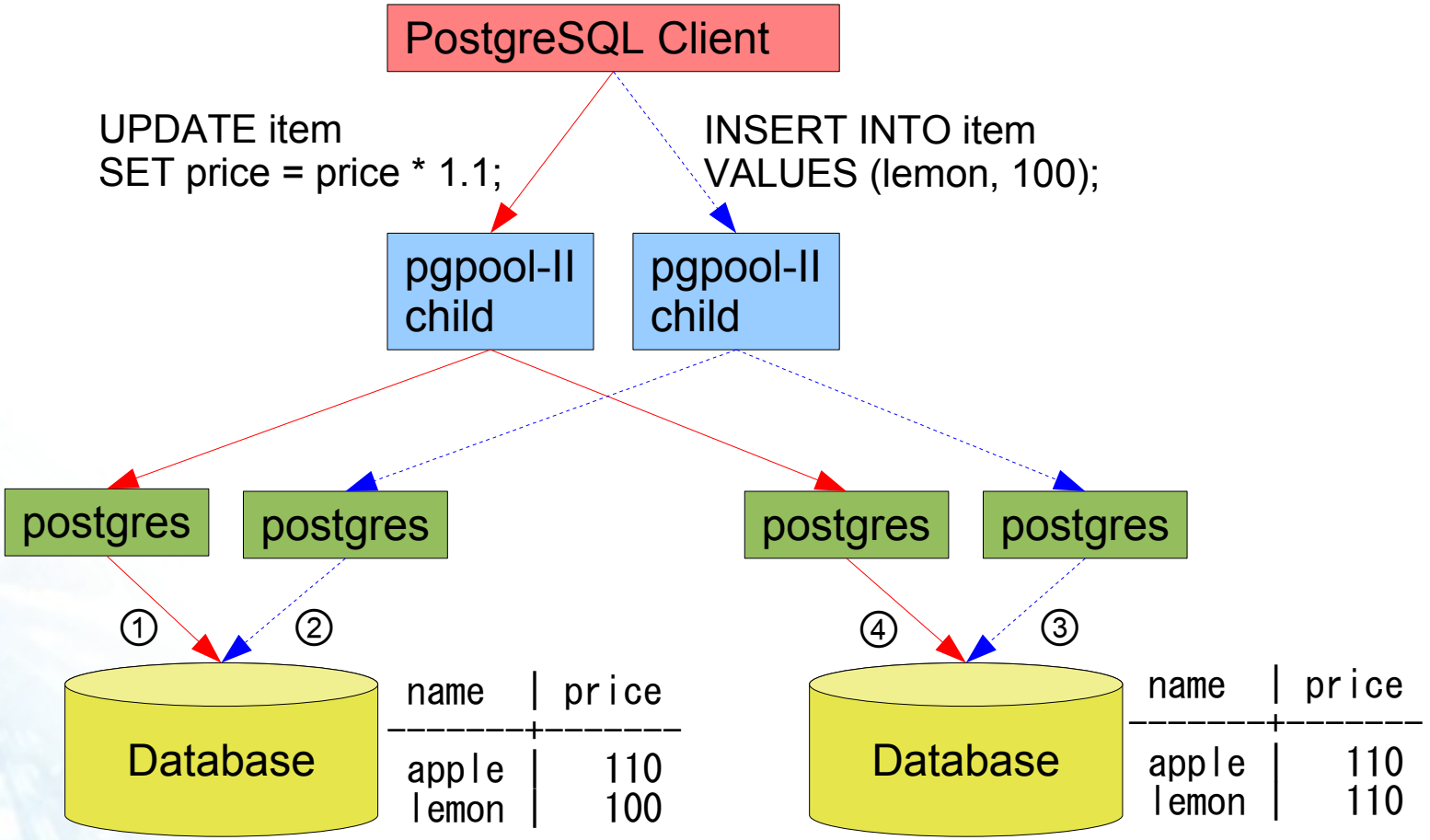
シーケンスの問題

シーケンス値がずれる原因



更新クエリの干渉問題

更新内容がずれる原因



SRA OSS, INC.

問題への対応

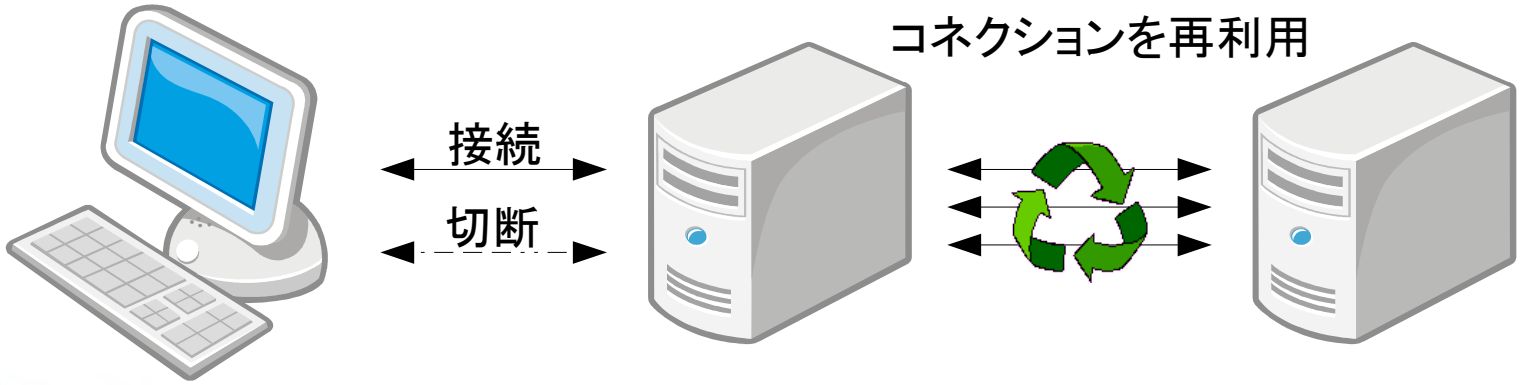
- 現状の実装
  - SERIALデータ型などシーケンスを使用しているテーブルへの挿入に限り、ロックを用いて排他制御を行い、挿入が並行して行われないようにする。
  - 更新行数が異なるときはトランザクションをアボート、もしくはフェイルオーバーする。

pgpool-IIの機能

- 利用したい機能を組み合わせて使用できる
  - コネクションプーリング
  - ロードバランス
    - PostgreSQLのパーサを用いてクエリの種類を判別
    - レプリケーション方式、レプリケーション遅延、トランザクション隔離レベル、一時テーブル・システムカタログへの検索、更新処理を含む関数の呼び出しなどを考慮してクエリを振り分ける
  - 自動フェイルオーバー
    - 死活監視
    - フェイルオーバー、フェイルバック時に任意のコマンドを実行可能
  - オンラインリカバリ
    - オンラインの状態でダウンしたサーバを復旧する

  など

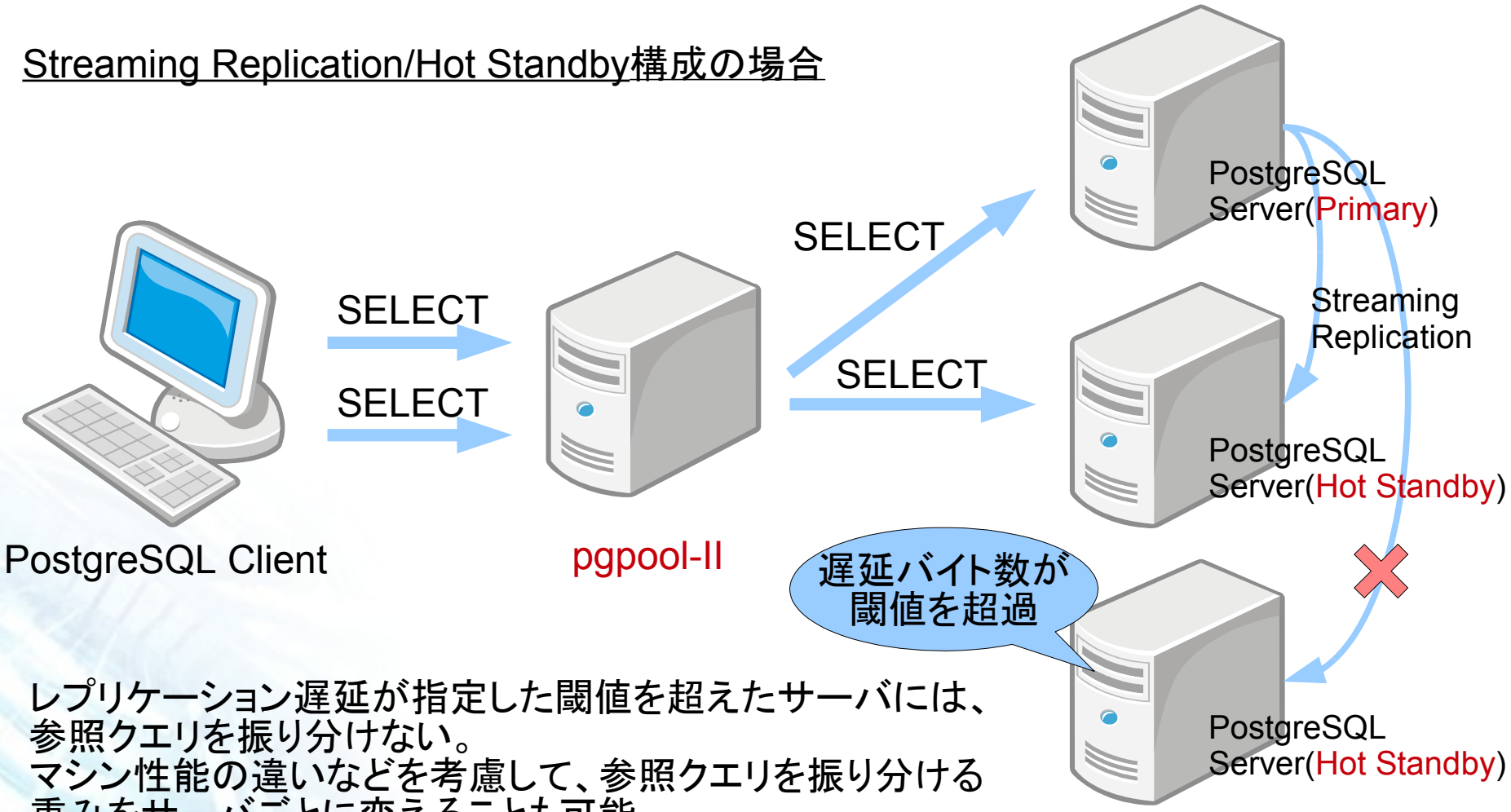
コネクションプーリング



すでに確立しているコネクションを再利用することにより、PostgreSQLが接続時に行っている、認証、子プロセスの生成、データアクセスのための前処理などを省いて効率化できる。

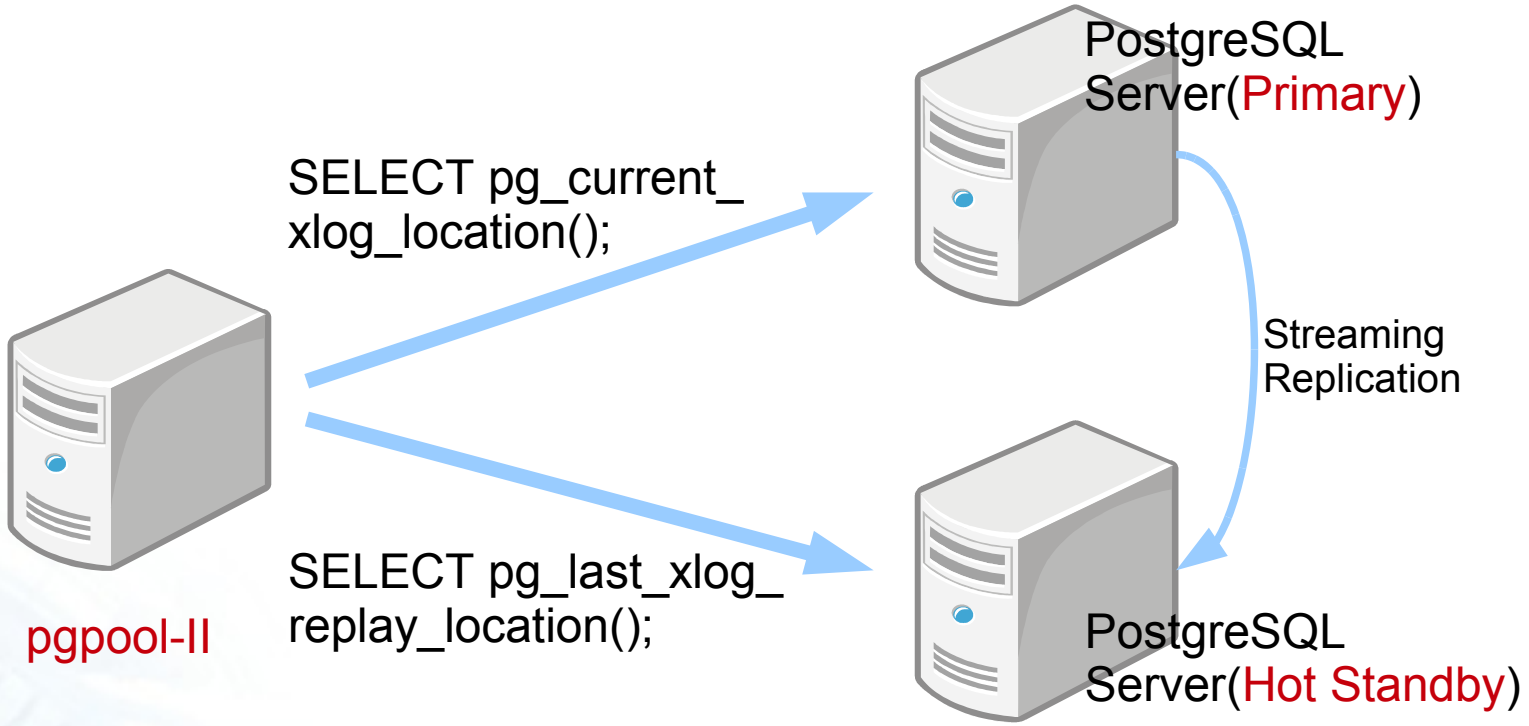
ロードバランス

Streaming Replication/Hot Standby構成の場合



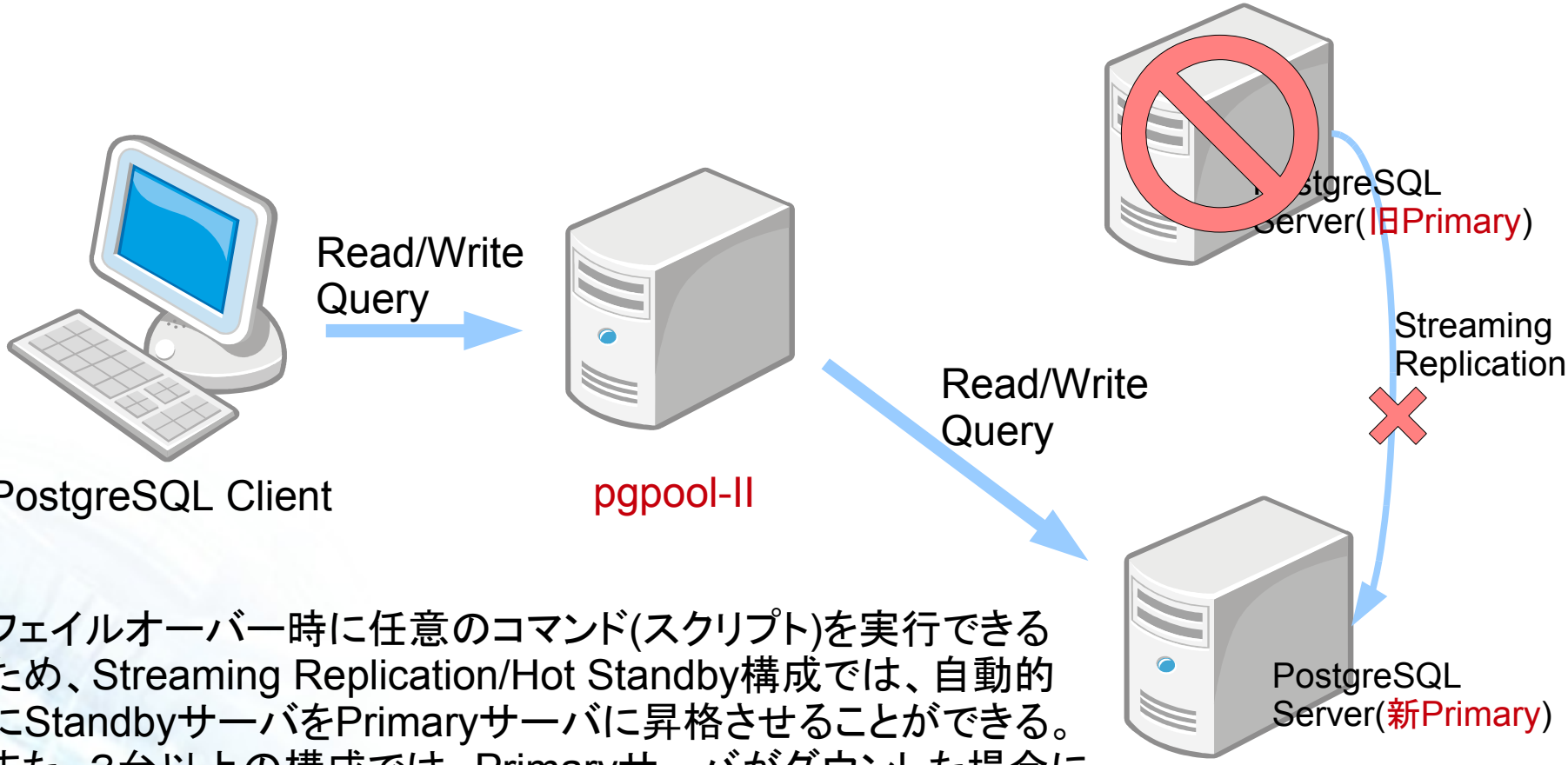
レプリケーション遅延が指定した閾値を超えたサーバには、参照クエリを振り分けない。
マシン性能の違いなどを考慮して、参照クエリを振り分ける重みをサーバごとに変えることも可能。

レプリケーション遅延の求め方



pgpool-IIが定期的に以下の関数を実行し、トランザクションログの位置の差から遅延を求める。
pg_current_xlog_location(): 現在のトランザクションログの書き込み位置を返す
pg_last_xlog_replay_location(): リカバリ中に再生された最後のトランザクションログの位置を返す

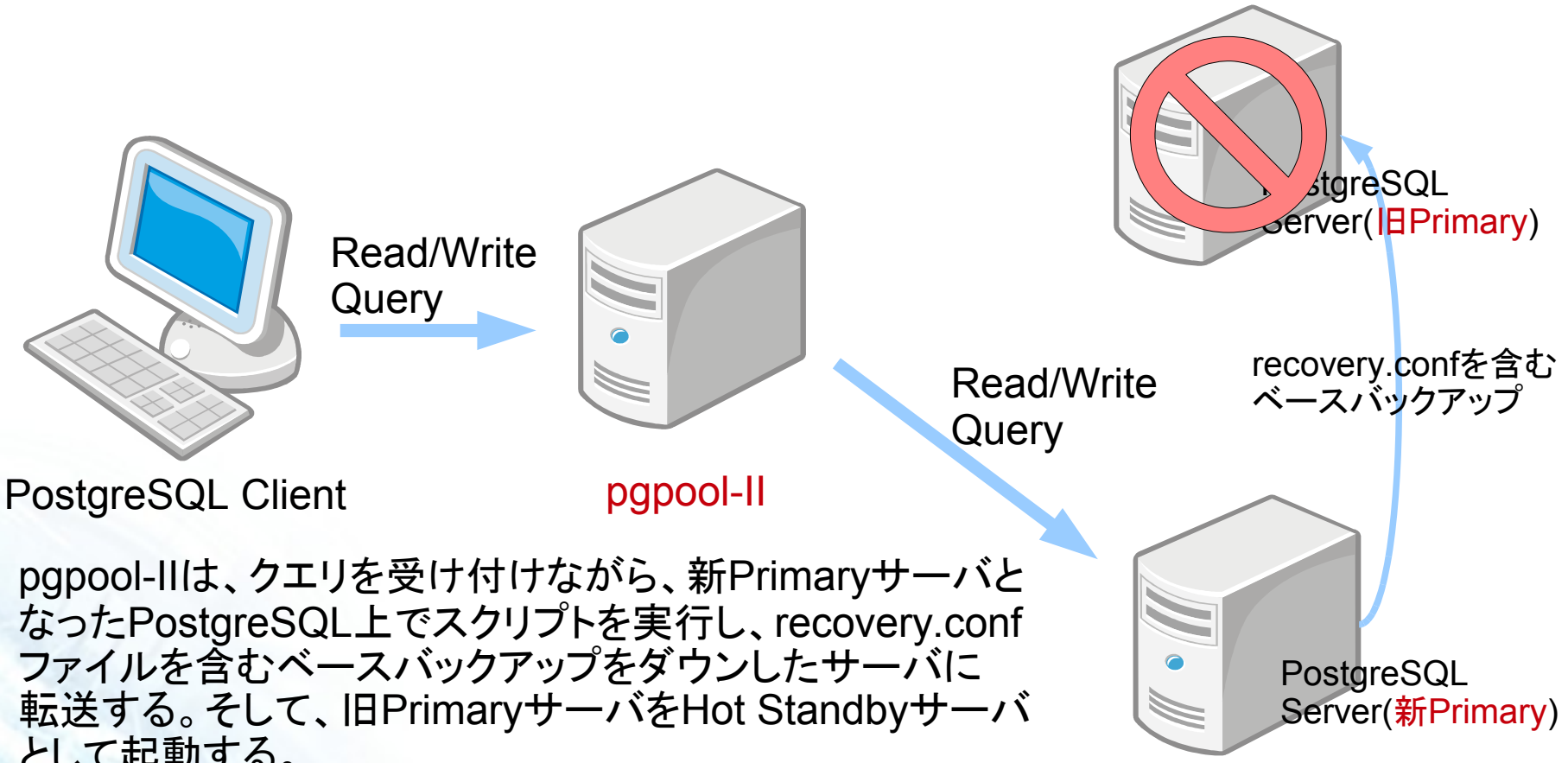
自動フェイルオーバー



フェイルオーバー時に任意のコマンド(スクリプト)を実行できるため、Streaming Replication/Hot Standby構成では、自動的にStandbyサーバをPrimaryサーバに昇格させることができる。また、3台以上の構成では、Primaryサーバがダウンした場合に同期が取れなくなったStandbyサーバを、自動的に新Primaryサーバにつなぎかえることも可能。

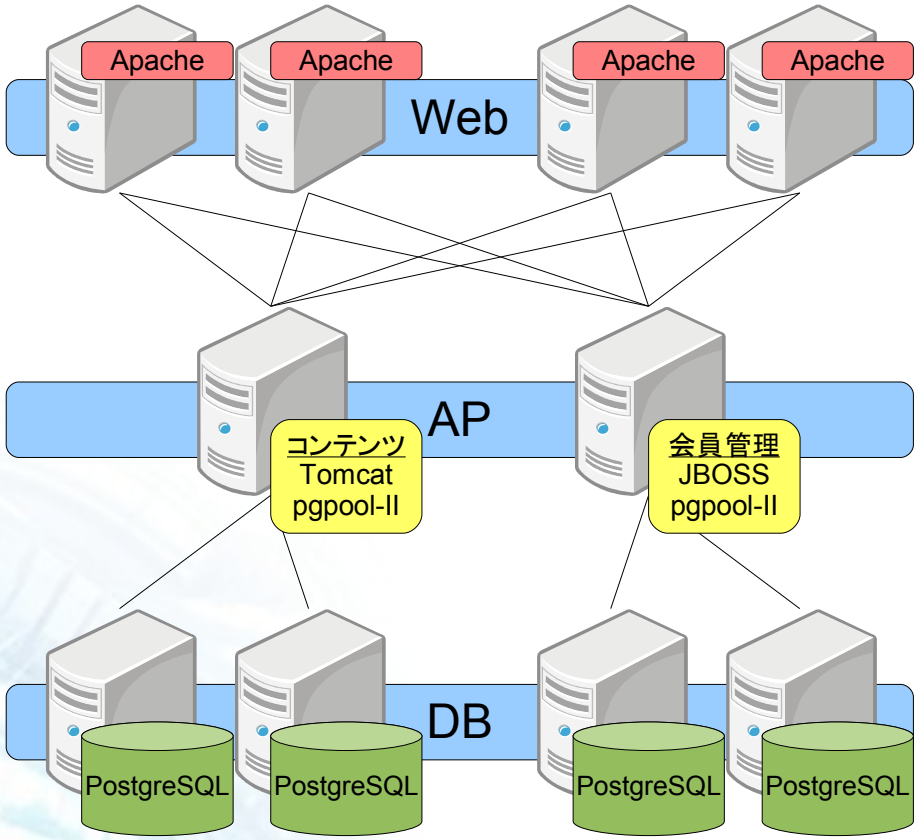
SRA OSS, INC.

オンラインリカバリ



pgpool-IIは、クエリを受け付けながら、新PrimaryサーバとなったpostgreSQL上でスクリプトを実行し、recovery.confファイルを含むベースバックアップをダウンしたサーバに転送する。そして、旧Primaryサーバを Hot Standbyサーバとして起動する。
同期レプリケーション機能を使用している場合は、ベースバックアップのほかにアーカイブログとトランザクションログもダウンしたサーバに転送する必要がある。

導入事例 「JTB旅カード」Webサイト



・概要
  ・カード会員が利用するポータルサイト
  ・ポイント管理を行う

・システム構成
  ・Webサーバ、APサーバ、DBサーバの3層構造
  ・APサーバとDBサーバはコンテンツ用と会員管理用で2つに分かれている

・pgpool-IIの同期レプリケーションやロードバランス、自動フェイルオーバー機能などを使用し、可用性と性能を向上

SRA OSS, INC.

今後の計画

- メモリベースのクエリキャッシュ機能
  - キャッシュストレージとして共有メモリとmemcachedを選択可能
  - キャッシュ更新は自動的に行う
    - 更新クエリが来たらキャッシュをクリア
    - 一定時間が過ぎたらキャッシュをクリア
  - Google Summer of Codeにてプロトタイプを実装
- pgpool-II自体の組み込みHA機能
  - pgpool-IIが単一障害点になることを避けたい
  - HAクラスタソフトウェアを使用しなくても、pgpool-IIを容易に冗長化できるようにしたい

まとめ

- pgpool-IIの特徴
  - レプリケーション方式を選択できる
    - クエリベースの同期レプリケーション
    - 他のレプリケーションツールとの連携
      - Streaming Replication/Hot Standbyとの連携は
        クエリの制約や性能面を考慮するとお勧め
  - 多彩な機能を組み合わせて使用できる
    - 可用性面： 同期レプリケーション、自動フェイルオーバー、
      オンラインリカバリなど
    - 性能面： ロードバランス、コネクションプーリングなど
- 導入事例
  - 「JTB旅カード」Webサイト
  - その他の導入、コンサルティング、サポートの実績多数

参考URL

- pgpool-II のWebサイト
  - http://pgpool.projects.postgresql.org/
- pgpool-II のダウンロード
  - http://pgfoundry.org/projects/pgpool/
- pgpool Wiki (pgpool 日本語MLも)
  - http://pgpool.sraoss.jp/
- Twitter
  - @pgpool2

ご清聴ありがとうございました。

SRA OSS, INC.