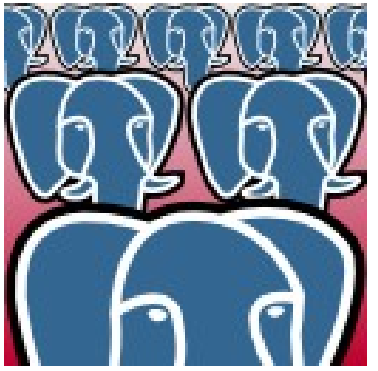
pgpool-II 最新バージョン 3.5 のご紹介

SRA OSS, Inc. 日本支社
pgpool-II 開発者
長田　悠吾

SRA OSS,INC.

はじめに

- pgpool-II とは
  - PostgreSQL のクラスタを管理、活用するためのミドルウェア
  - SRA OSS を中心に開発
  - オープンソースソフトウェア
    - BSDライセンス
    - メジャーバージョンアップは年1回
    - この秋に 3.5 をリリース予定



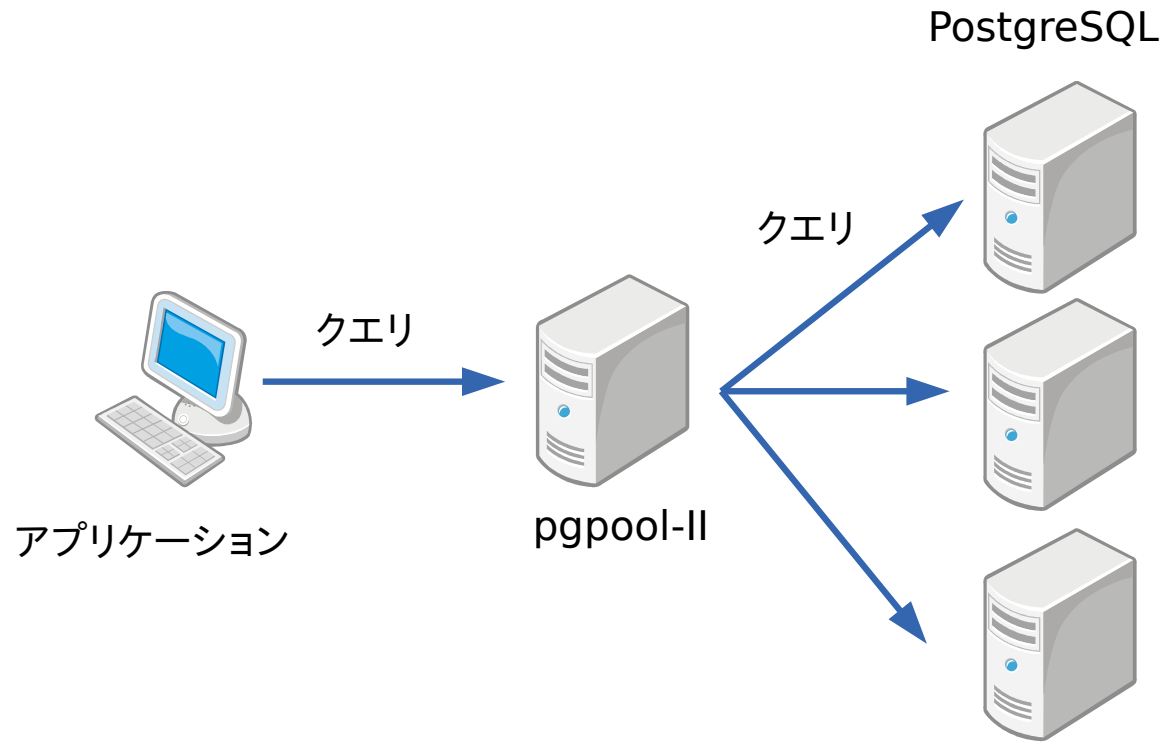
- 今回は、pgpool-II 3.5の新機能、改善点についてご紹介します
  - PostgreSQL 9.5 対応
  - 組み込み HA 機能（watchdog）の改善
  - 拡張プロトコル使用時の性能改善

pgpool-II について

pgpool-II の機能

- アプリケーションと複数 PostgreSQLの間に入って、様々な機能を提供
  - アプリケーションからは1台の PostgreSQL のように見える

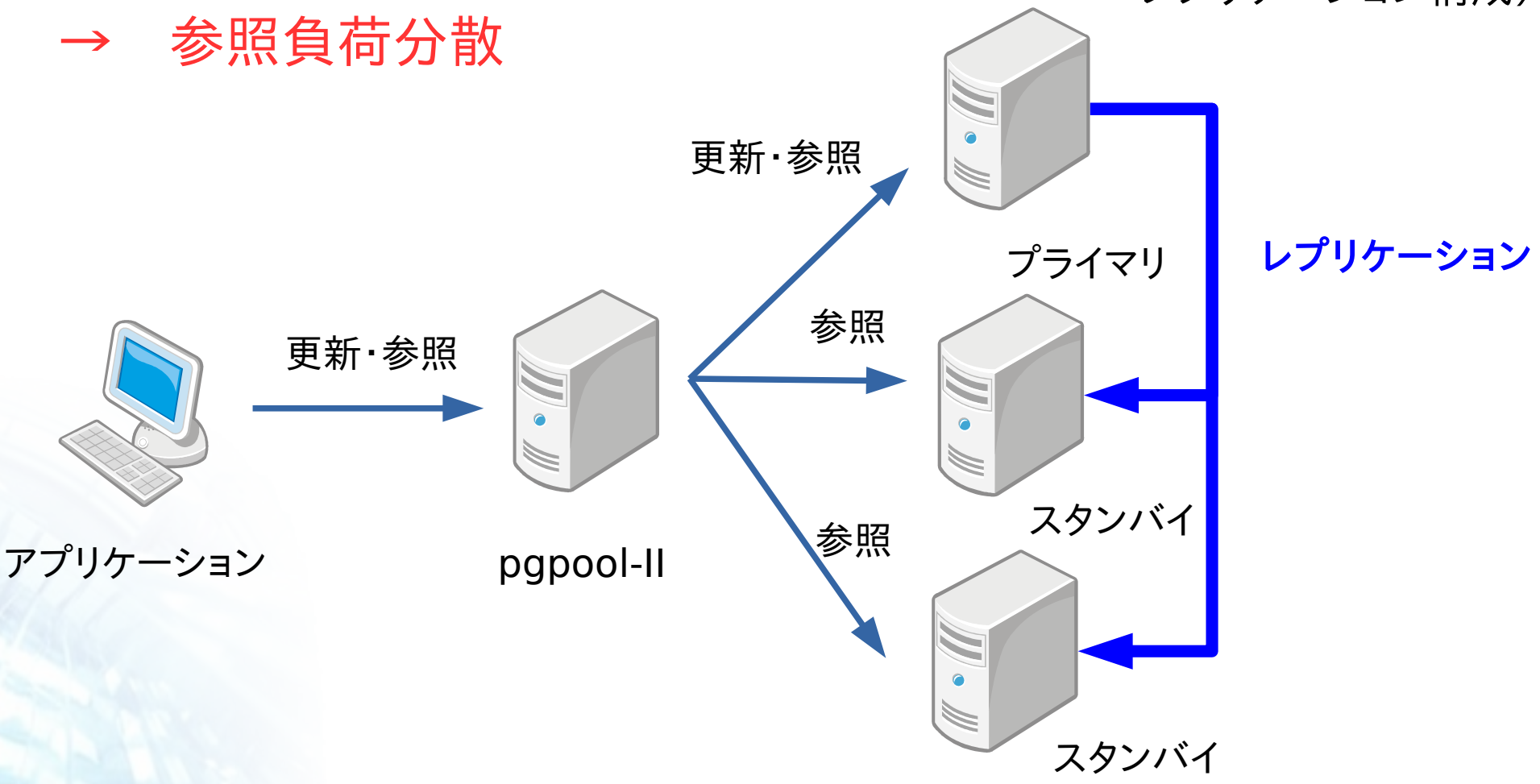
- 性能向上
  - コネクションプーリング
  - 参照負荷分散
  - クエリキャッシュ
- 高可用性
  - 自動フェイルオーバ
  - watchdog
- クラスタ管理
  - オンラインリカバリ
  - ネイティブレプリケーション
- クラスタとアプリケーションの親和性
  - クエリの自動振り分け



クエリの自動振り分け

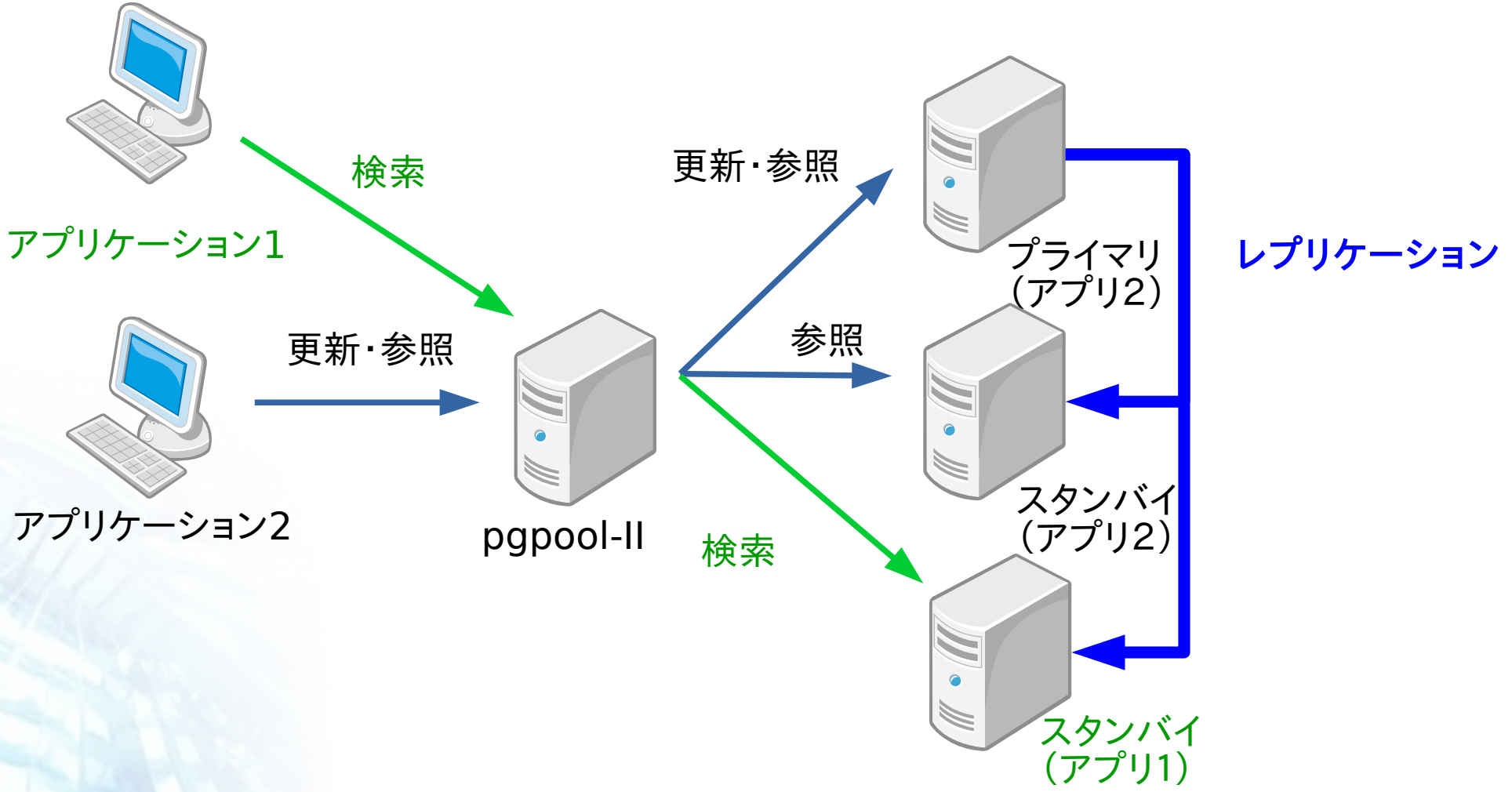
- 更新クエリはプライマリサーバへ
- 参照クエリはサーバ間で振り分け

→　参照負荷分散



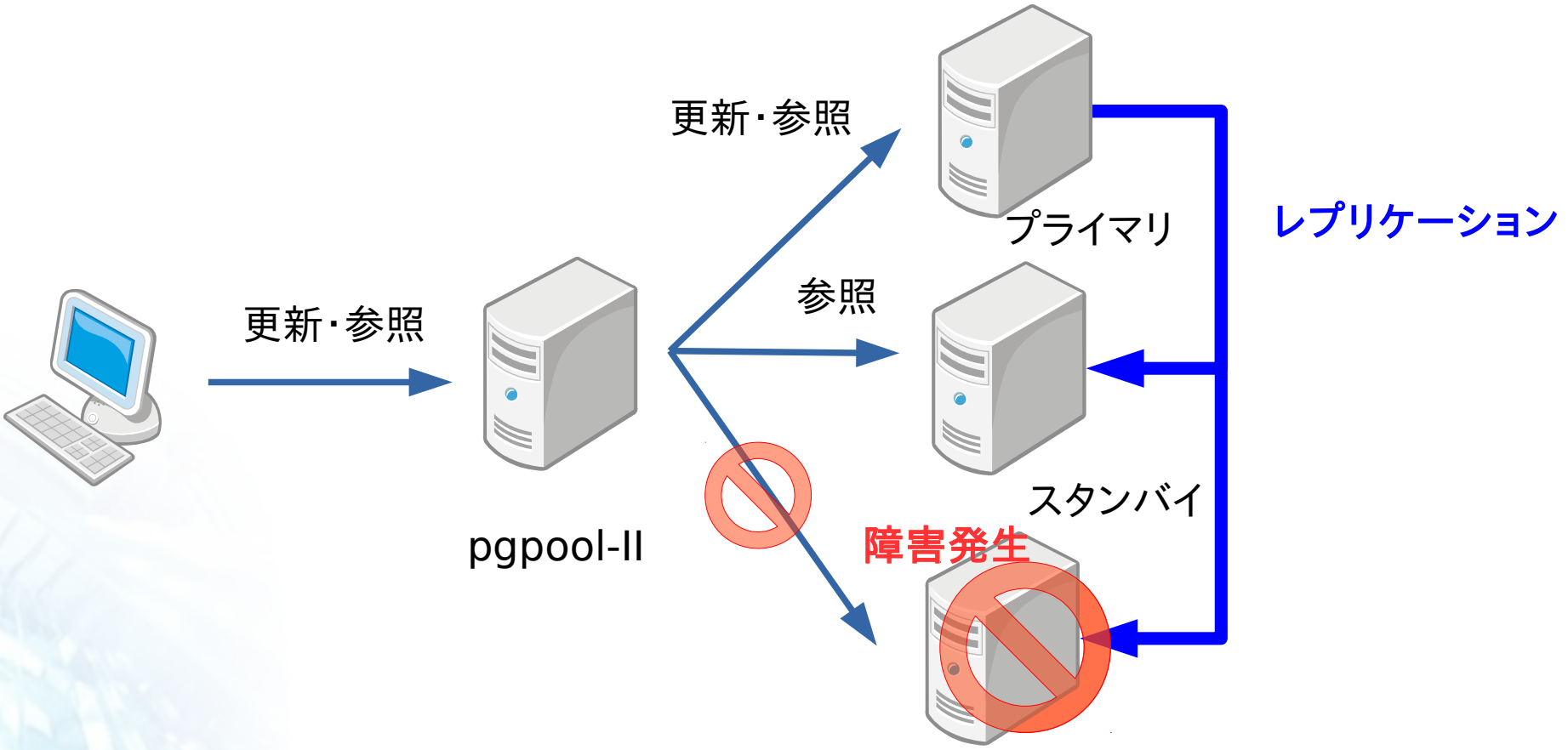
クエリの自動振り分け

- アプリケーション名や、DB名に基づくクエリの振り分けも可能(pgpool-II 3.4～)



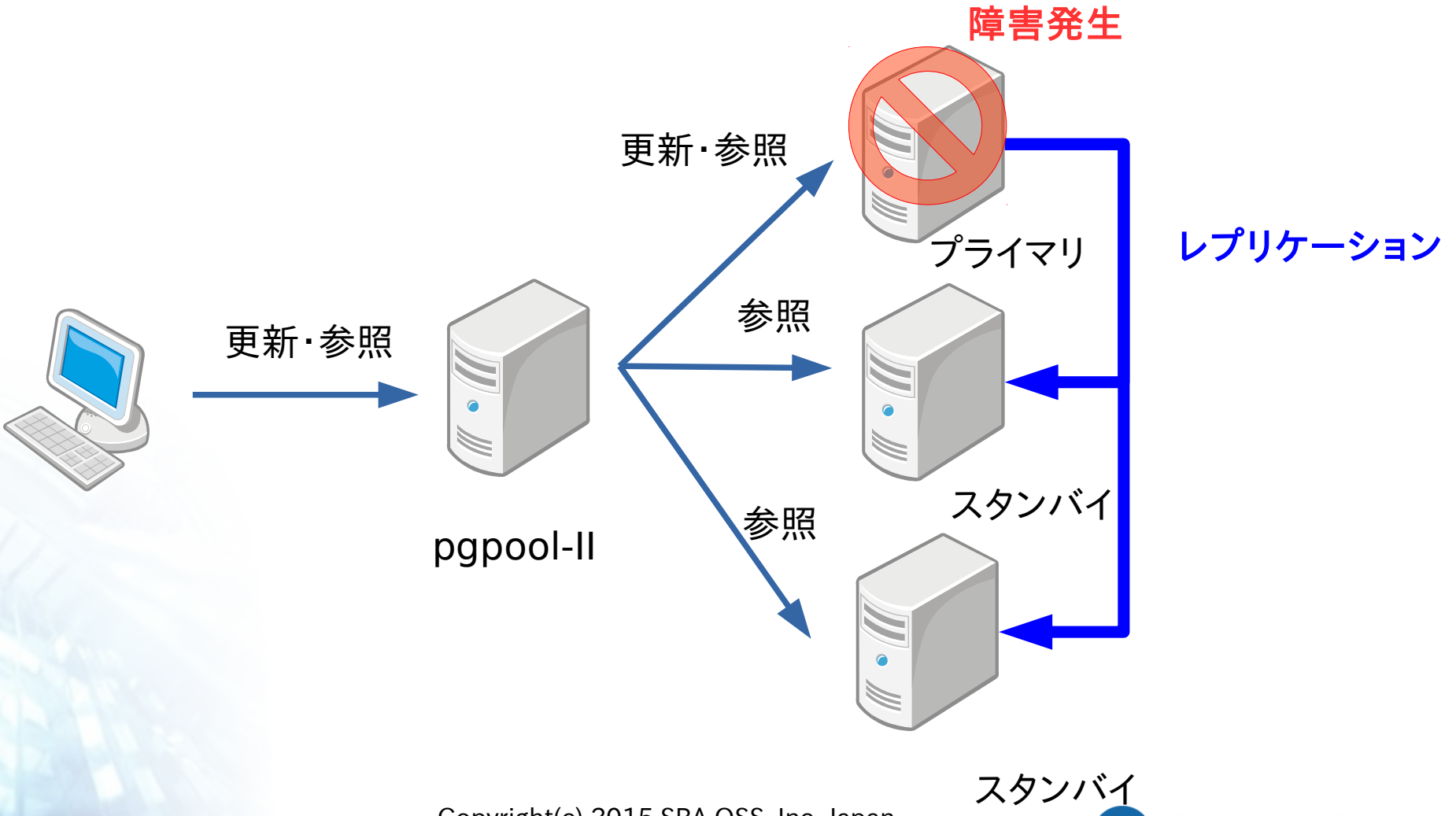
自動フェイルオーバ

- DBサーバの障害を自動検出（ヘルスチェック機能）
  - ダウンしたPostgreSQLを切り離す
    → 負荷分散の対象から外れる



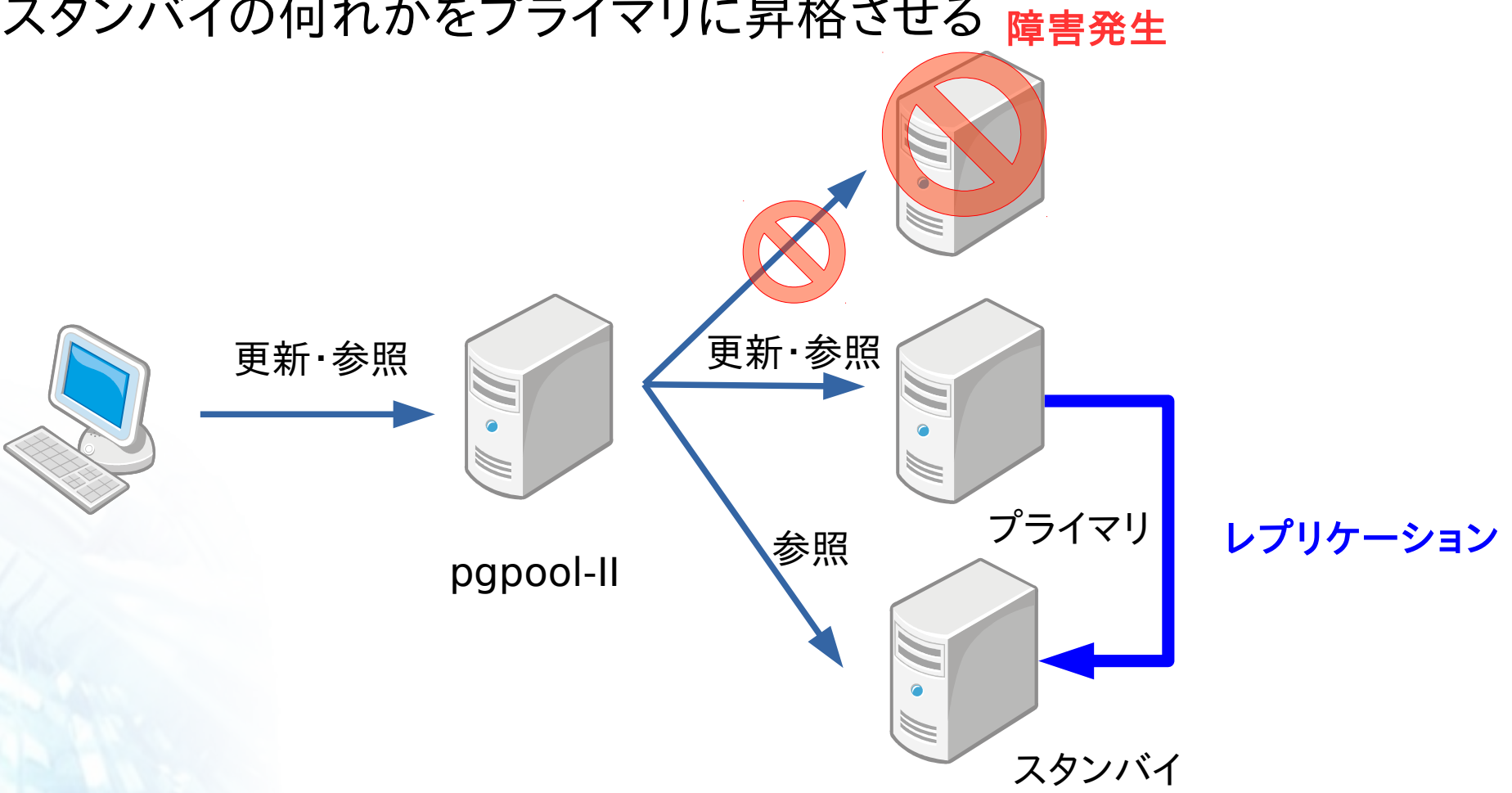
自動フェイルオーバ

- プライマリサーバに障害が発生した場合



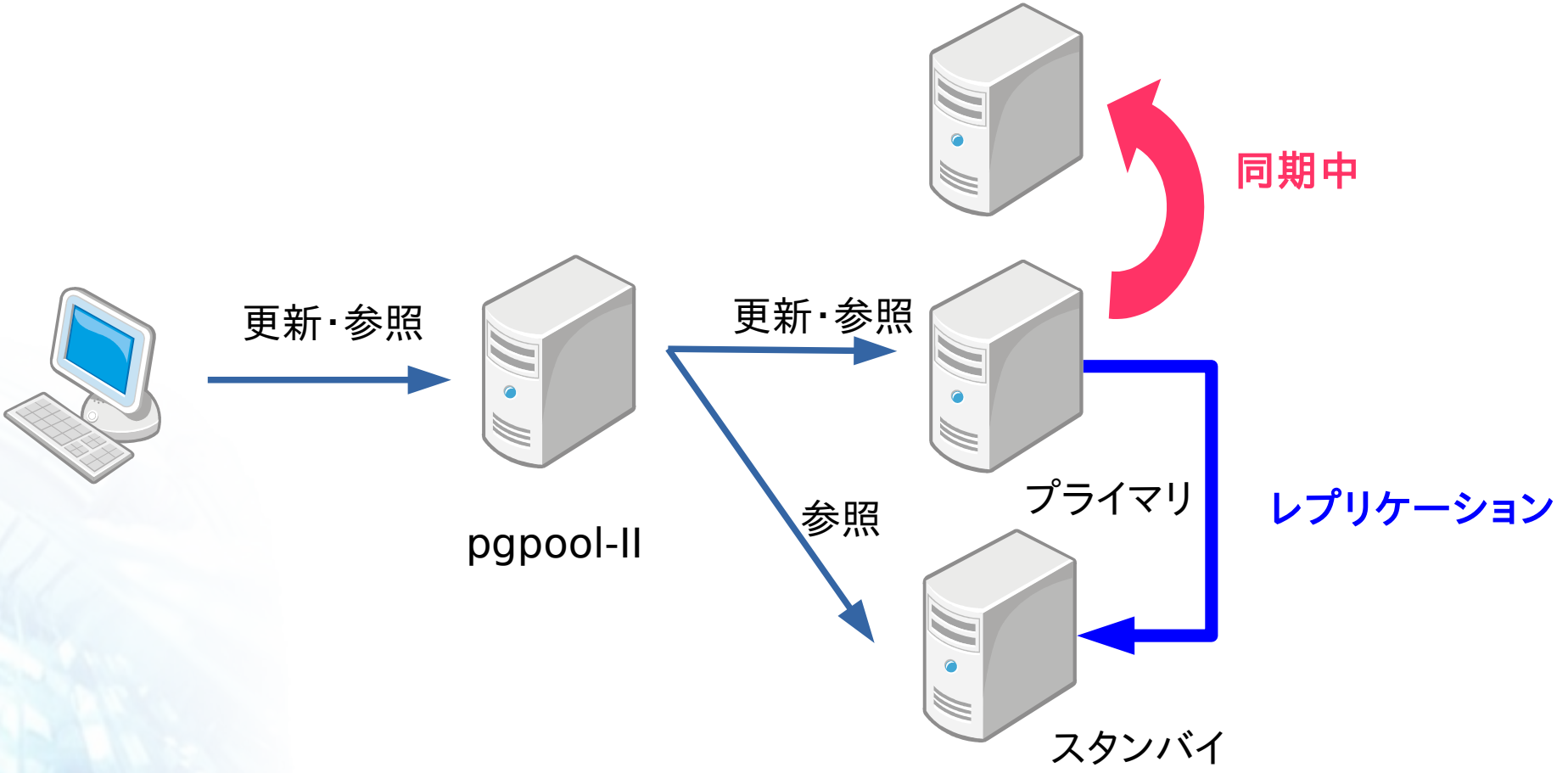
自動フェイルオーバ

- プライマリサーバに障害が発生した場合
  - 負荷分散の対象から切り離す
  - スタンバイの何れかをプライマリに昇格させる



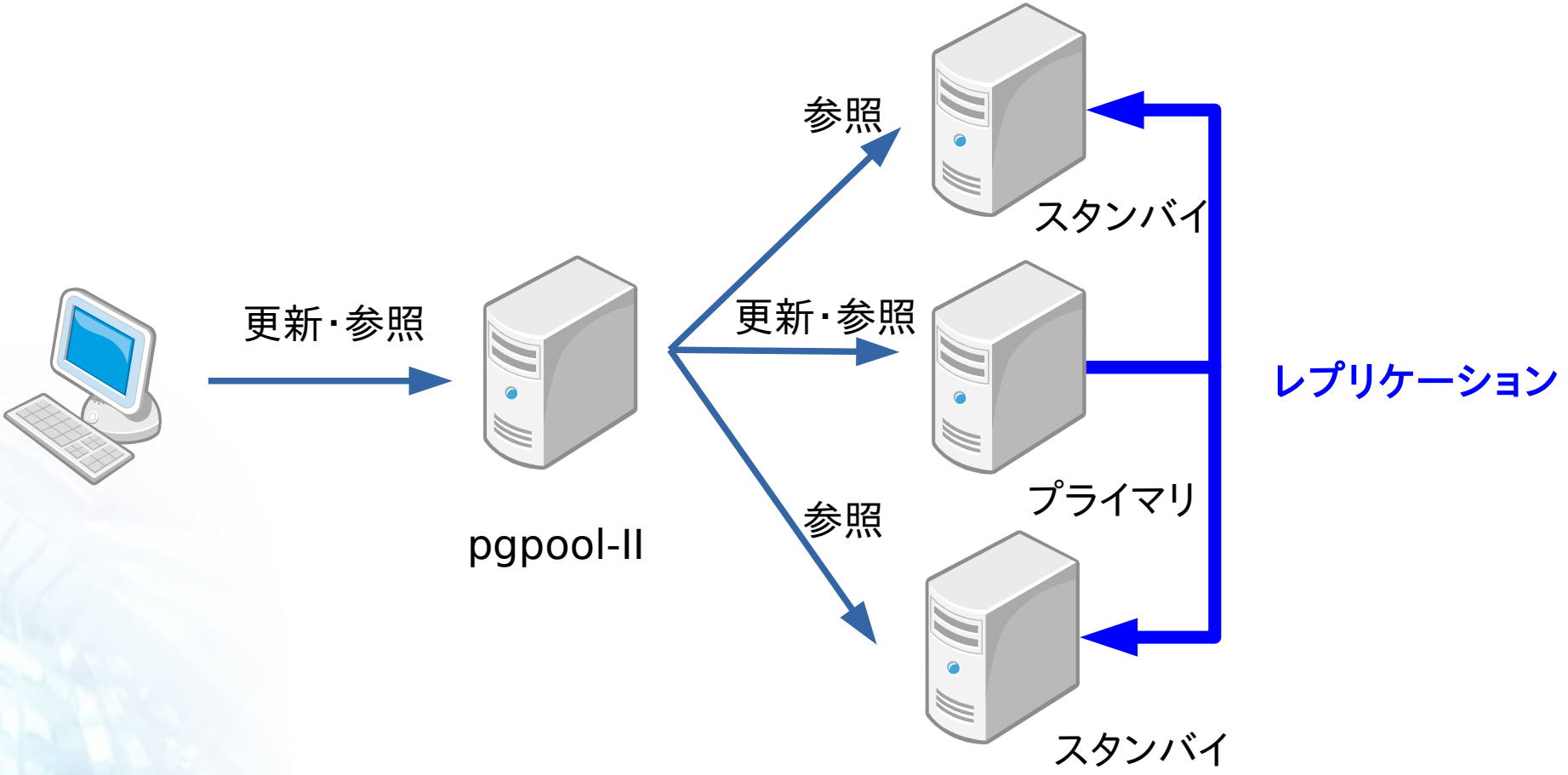
オンラインリカバリ

- ダウンしたスタンバイをプライマリに再同期させる
- システム稼働中でも実行可能



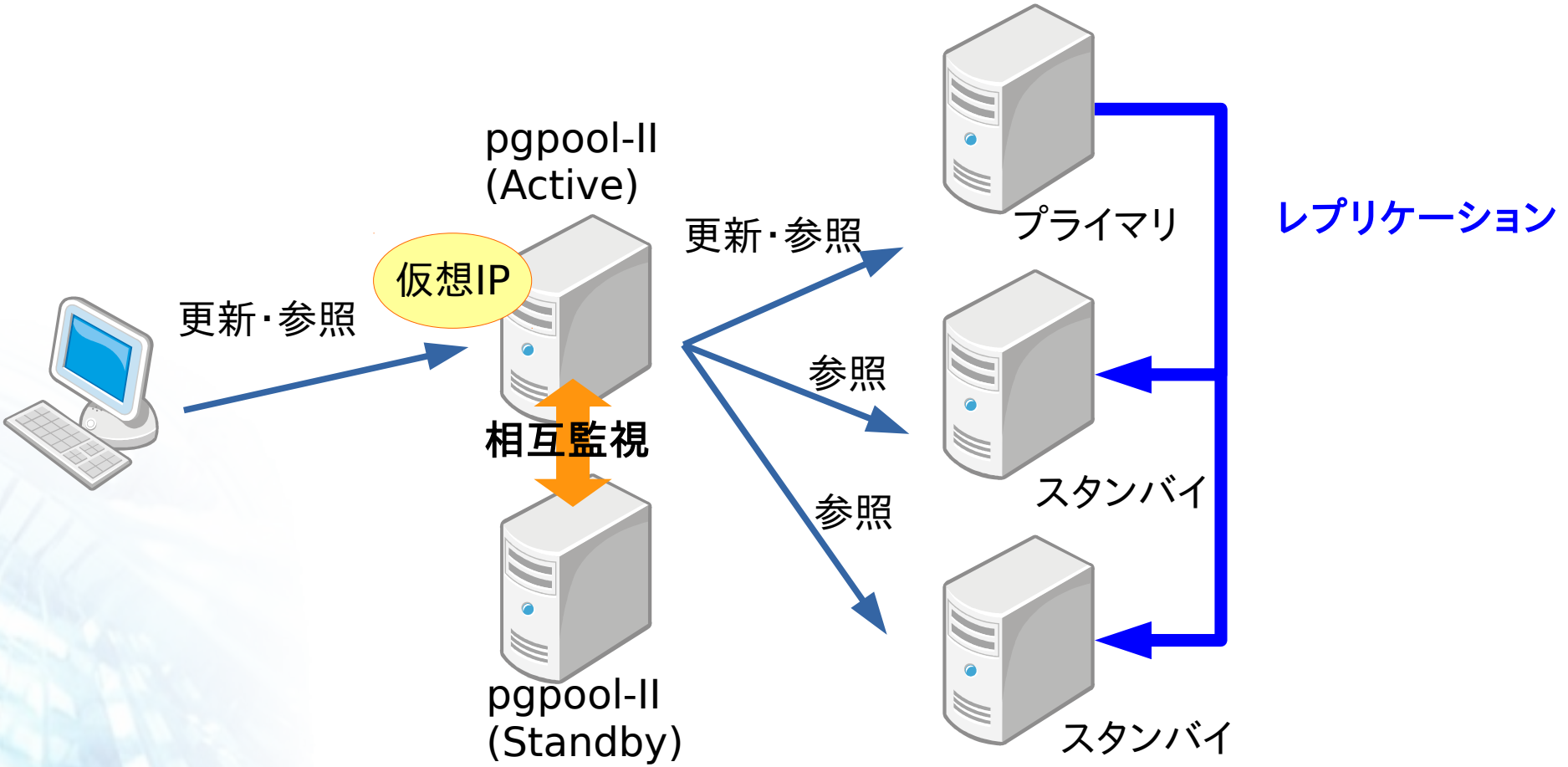
オンラインリカバリ

- 同期完了後、マスタからのレプリケーションが再開
- 自動的に負荷分散の対象となる



watchdog

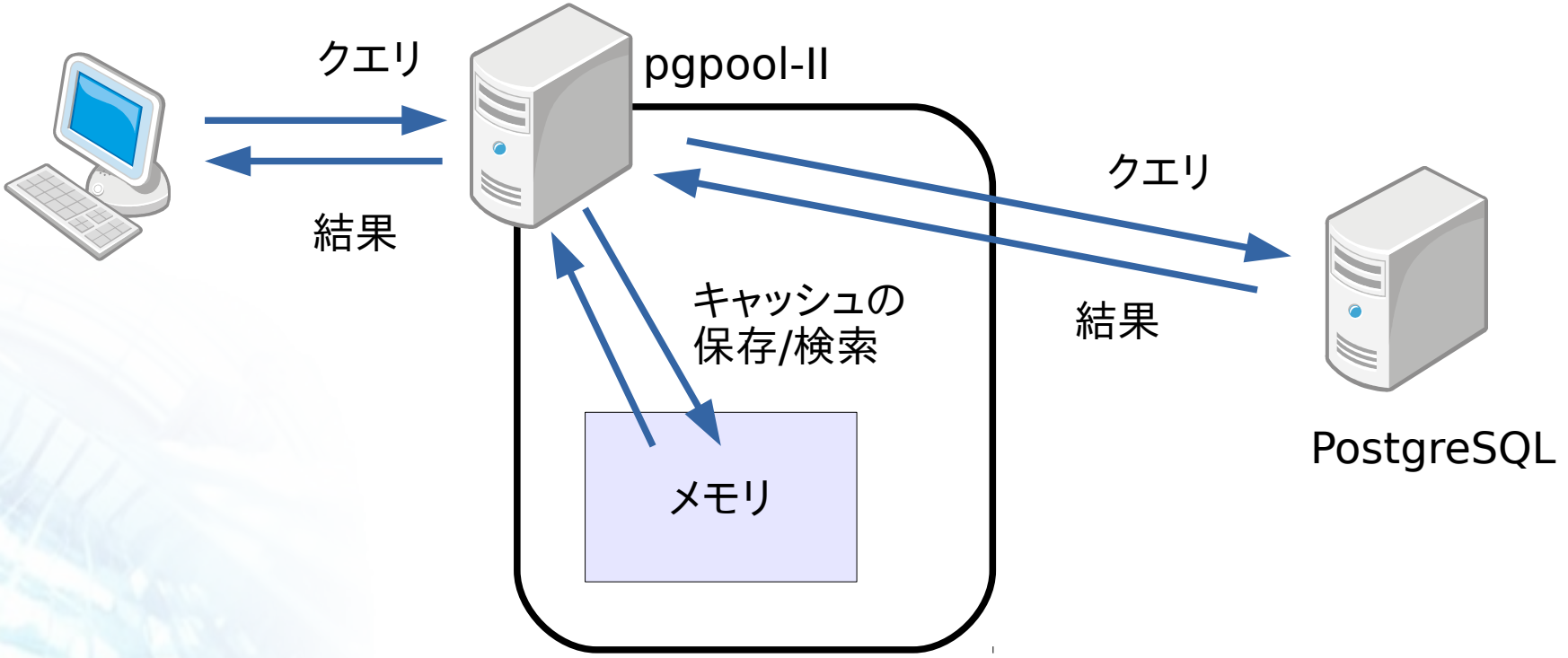
- pgpool-II 組み込みのHA機能
  - pgpool-II を Active/Standby 構成にすることで
    - 単一障害点となることを防止



SRA OSS, INC.

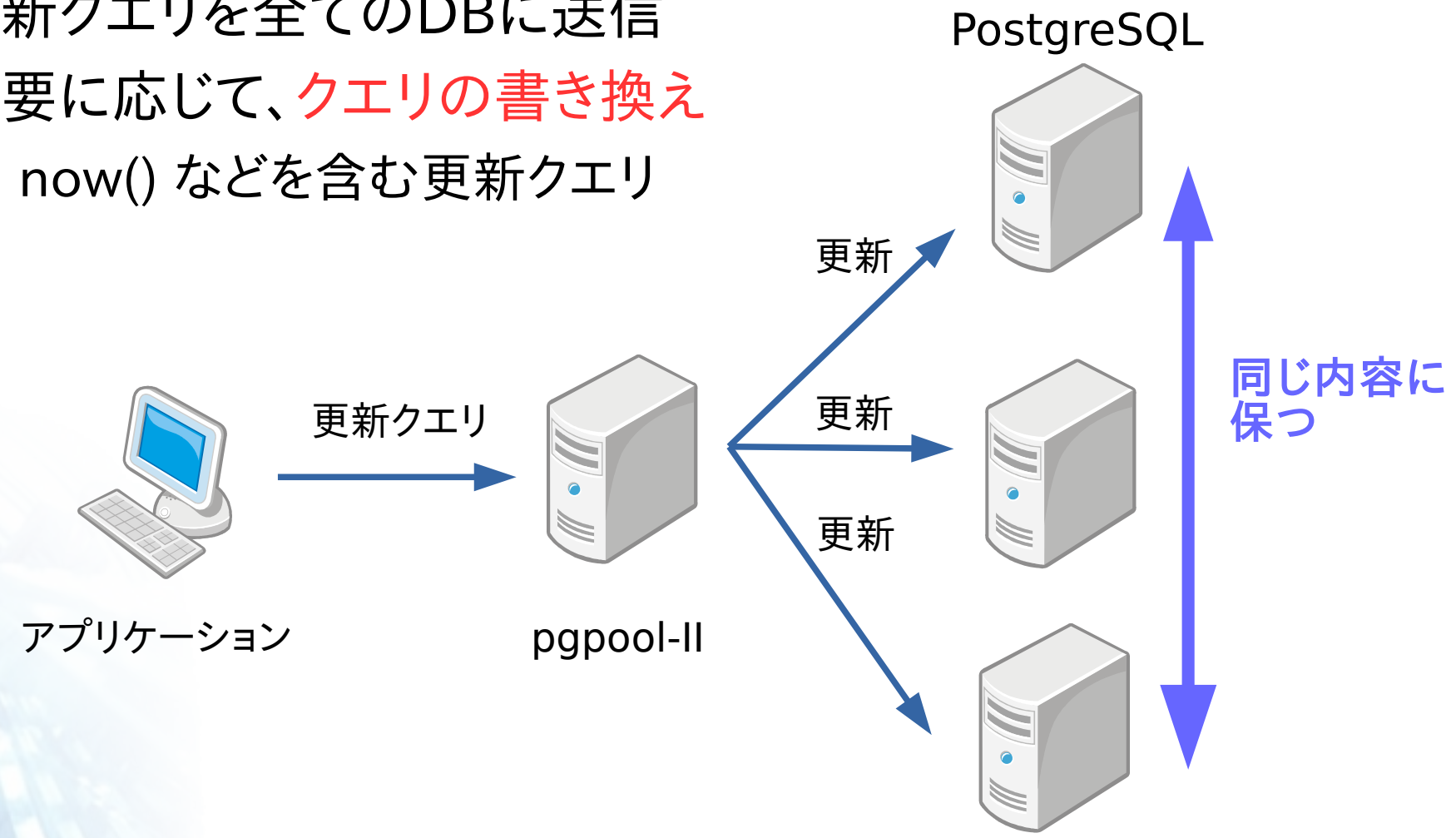
クエリキャッシュ

- SELECTクエリの結果をメモリ内にキャッシュする機能
  - 同じクエリが来たときに再利用する
  - DBへのアクセスが減り、応答速度が向上



ネイティブレプリケーション

- PostgreSQL のストリーミングレプリケーションを用いずに同期レプリケーションを実現するモード
  - 更新クエリを全てのDBに送信
  - 必要に応じて、クエリの書き換え
    - now() などを含む更新クエリ



次期バージョン pgpool-II 3.5 について

pgpool-II 3.5

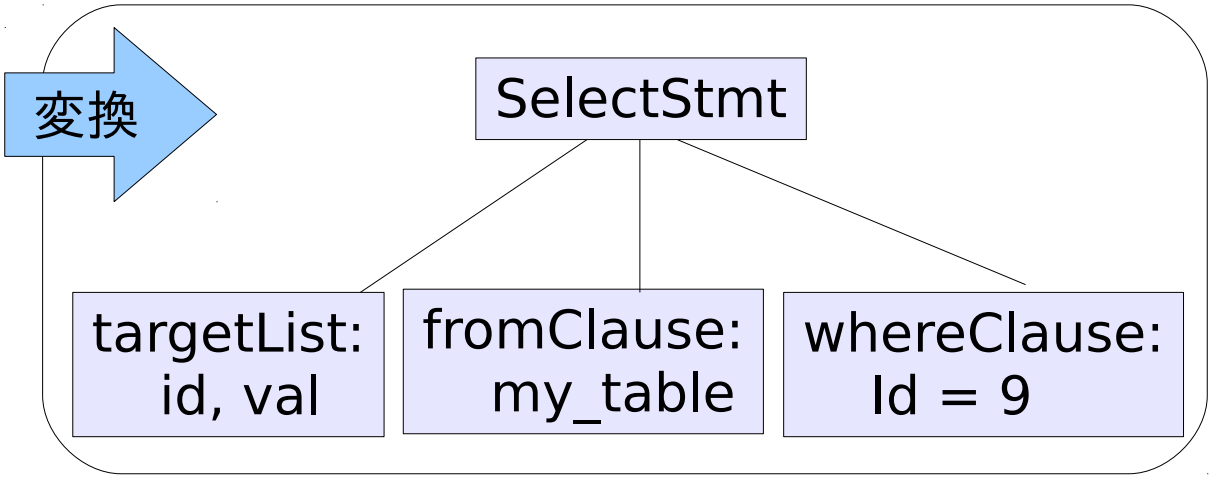
- 2015年秋にリリース予定
- 主な新機能
  - PostgreSQL 9.5 対応
  - watchdog機能の改善
  - 性能改善
  - pcpコマンドのオーバホール
  - パラレルクエリモードの廃止

PostgreSQL 9.5 対応

PostgreSQL 9.5 SQLパーサの移植(1)

- SQL パーサ
  - SQL 文字列を解析してパーツツリーに変換するモジュール

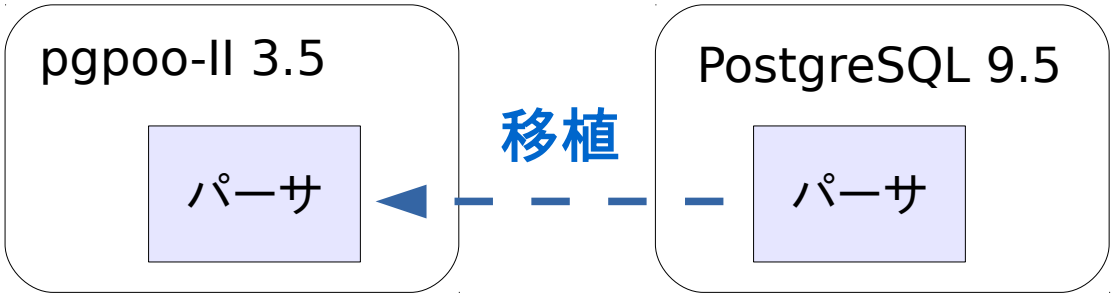
```
SELECT id, val FROM my_table
  WHERE id = 9;
```



- 参照負荷分散、クエリキャッシュ、クエリ書き換えなどで利用
  - 負荷分散可能なクエリか?
  - キャッシュ可能なクエリか?
  - 書き換えが必要なクエリか?

PostgreSQL 9.5 SQLパーサの移植(2)

- pgpool-II 3.5 では PostgreSQL 9.5 の SQL パーサを移植



  - 参照負荷分散、クエリキャッシュが、新しい SELECT 構文に対応
    - GROUPING SET, CUBE, ROLLUP,
    - TABLESAMPLE
  - クエリ書き換えが、新しい構文に対応
    - INSERT … ON CONFLICT
    - UPDATE tab SET (col1,col2,...) = (SELECT ...), ...

watchdog 機能の改善

Watchdog 機能の改善

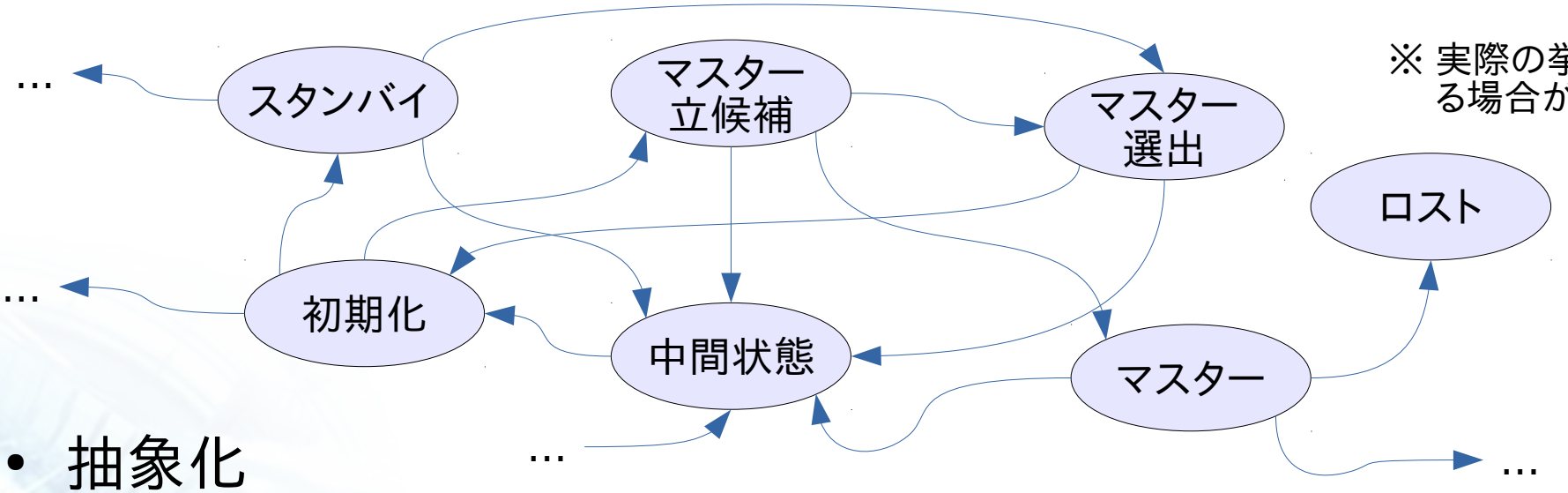
- 大幅なコードの改変
  - 元のコードがほとんど残されないほど

- 主な内容
  - 内部コードの改善
  - スプリットブレイン対策
  - Watchdog 内のプロセス間通信方式の変更
  - ノード間で設定パラメータの一貫性を検証

※ 現時点での予定

Watchdogの改善
ロバスト性の向上(1)

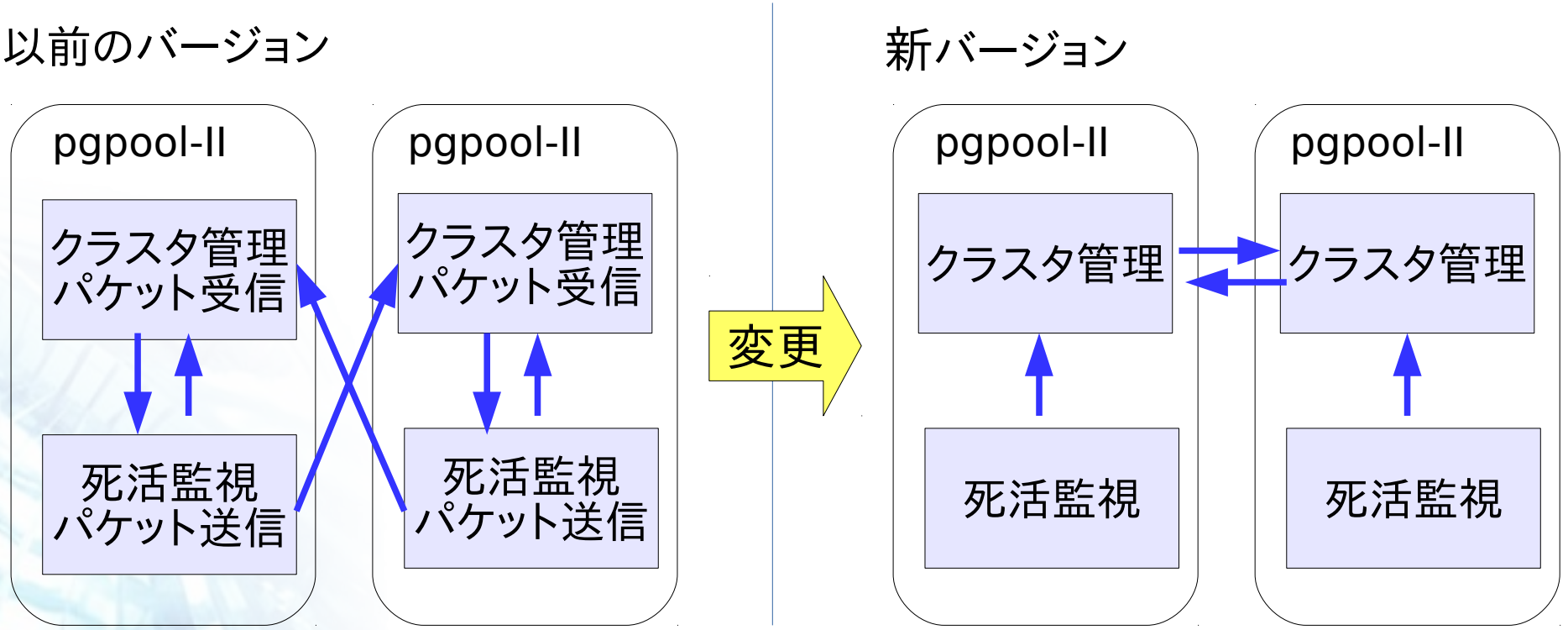
- watchdog のクラスタ管理を行うコア部分のコード
  - 「状態マシン」モデルで全面的に書き換え



- 抽象化
  - コードが理解しやすく、デバッグが容易に → 保守性の向上
  - 将来の機能拡張が容易に → 拡張性の向上

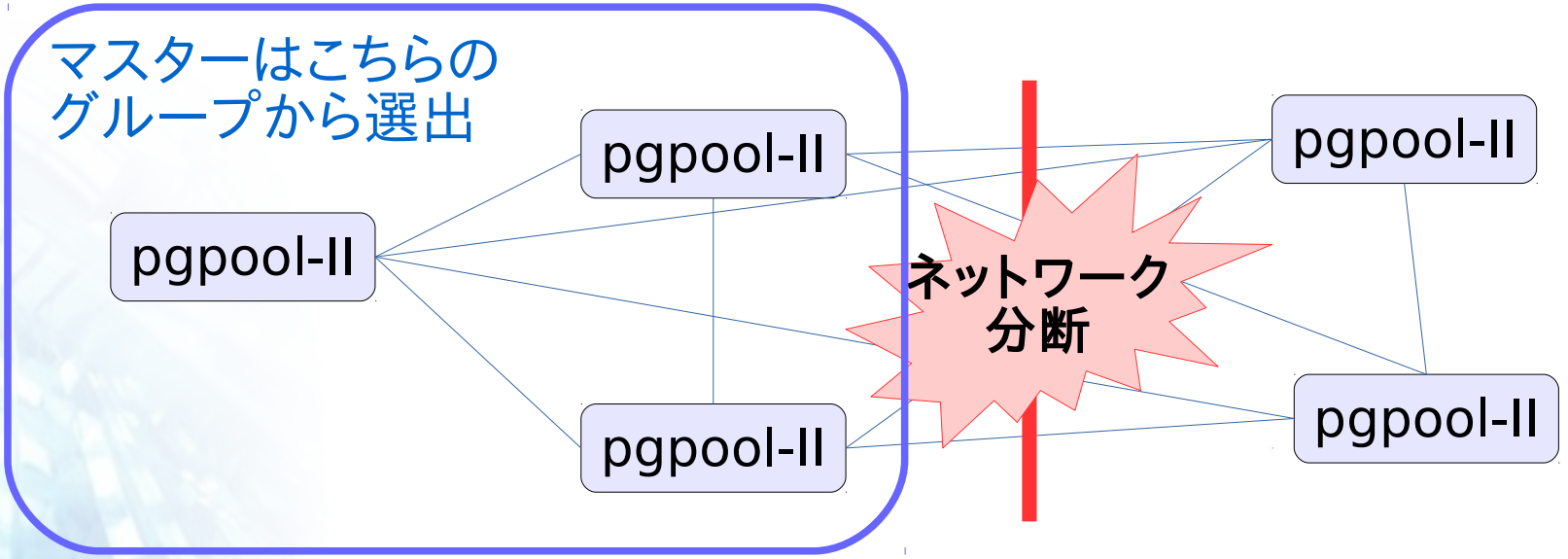
Watchdogの改善
ロバスト性の向上(2)

- クラスタ管理を行うコア部分を単一プロセスに統一
  - Watchdog 処理の一貫性の向上
  - 死活監視をコア部分と分離可能に



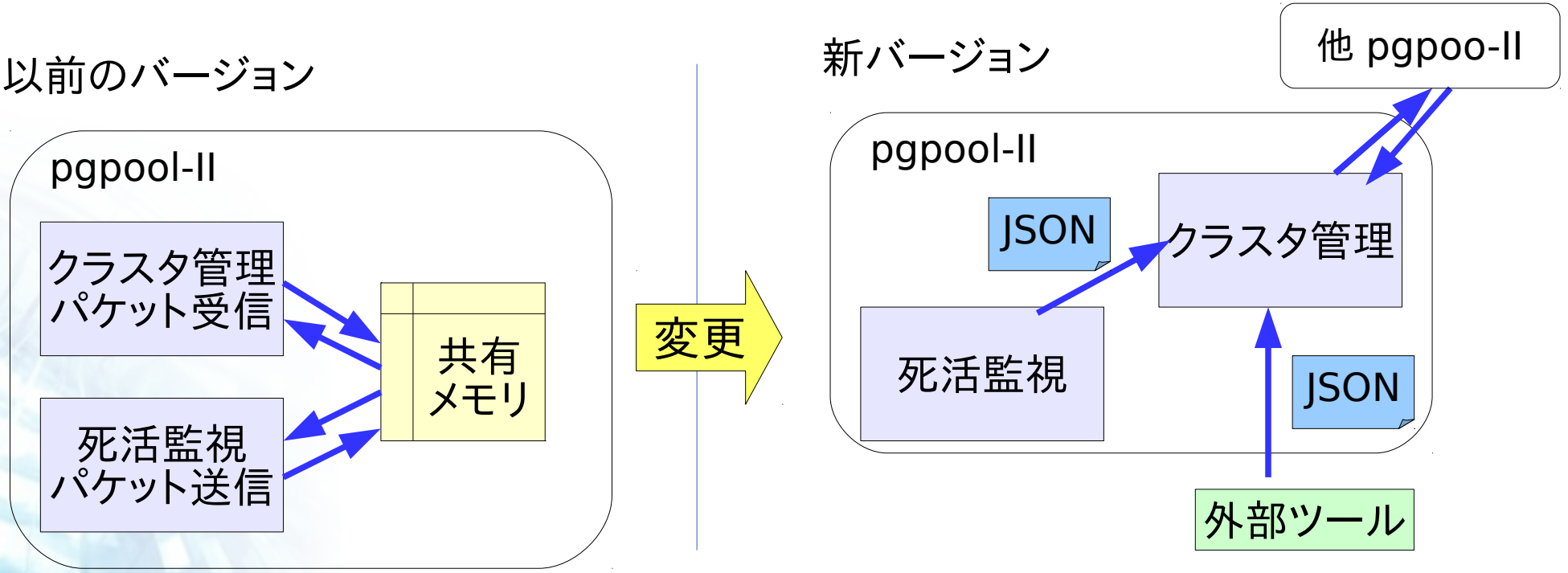
Watchdogの改善
ロバスト性の向上（3）

- スプリットブレイン対策の改善
  - ネットワークが分離された時に、どの pgpool-II がマスターとなるか決められなくなる問題
  - Quorum のサポート
    - クラスタに参加している全ノードのうち半数以上が自分と同じネットワークに属しているかどうか、をチェックする



Watchdogの改善
プロセス間連携

- watchdog 内部で行われる、プロセス間通信方式の変更
  - UNIX ドメインソケット & JSON 形式データ
- これにより、外部のサードパーティツールとの連携が可能に
  - 例）外部ツールを使った死活監視など



Watchdogの改善
その他

- ノード間で設定パラメータの一貫性を検証
  - pgpool-II 間で重要なパラメータの値に一貫性を持たせる
  - 設定ミスに起因する問題を軽減

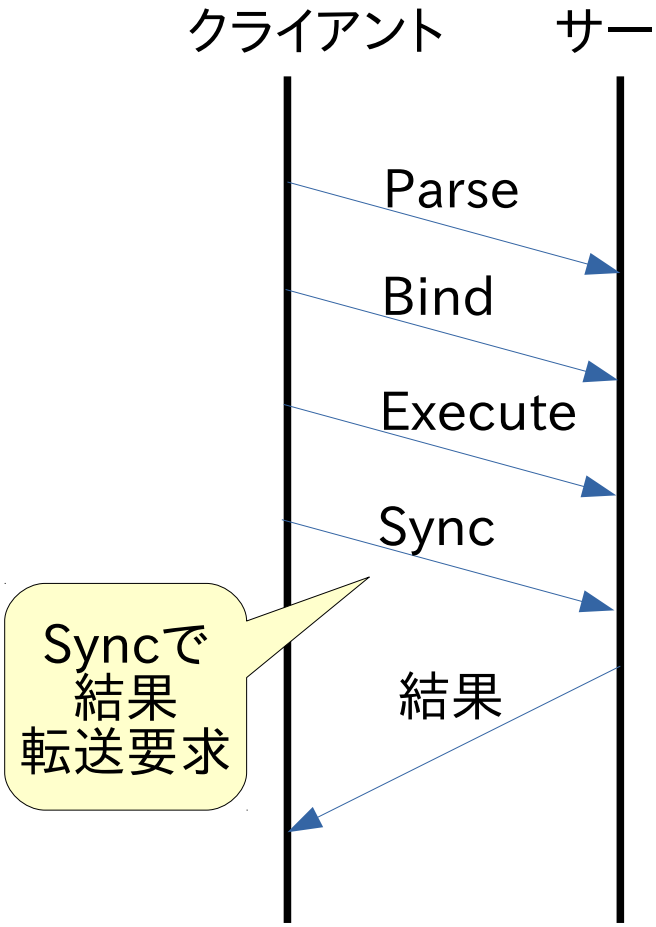
- ノードの優先度
  - 各ノードに異なる優先度を付与
  - 高い優先度のノードは、マスターに選ばれやすくなる

拡張問い合わせプロトコル
性能改善

拡張問い合わせプロトコル

- 単純問い合わせ

- 拡張問い合わせ

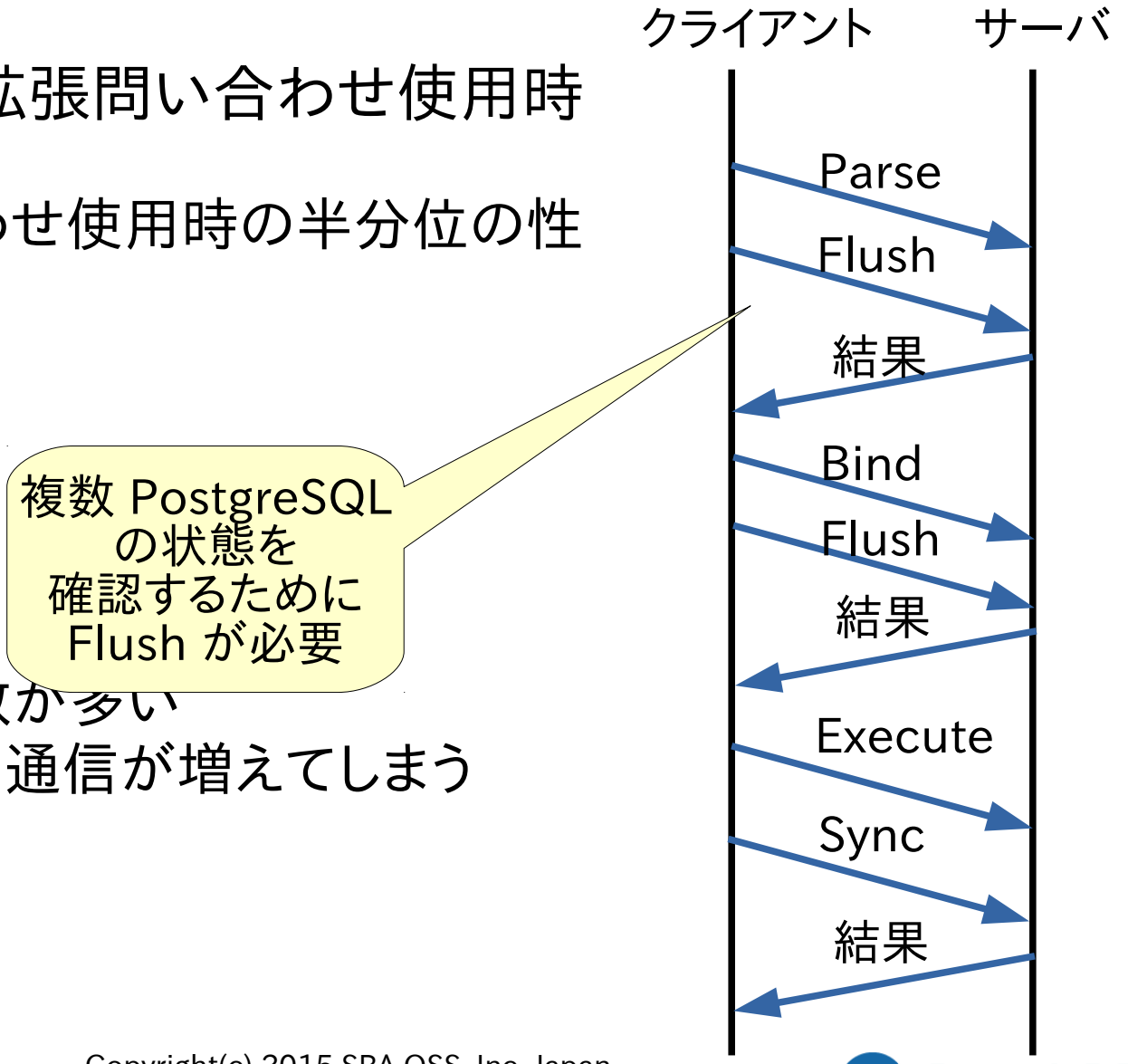


- 複数の段階に分けて処理
  - SQLの解析
  - パラメータ値の結び付け
  - 実行
- Java アプリケーションの JDBC ドライバで使用される

一部やり取りを省略しています

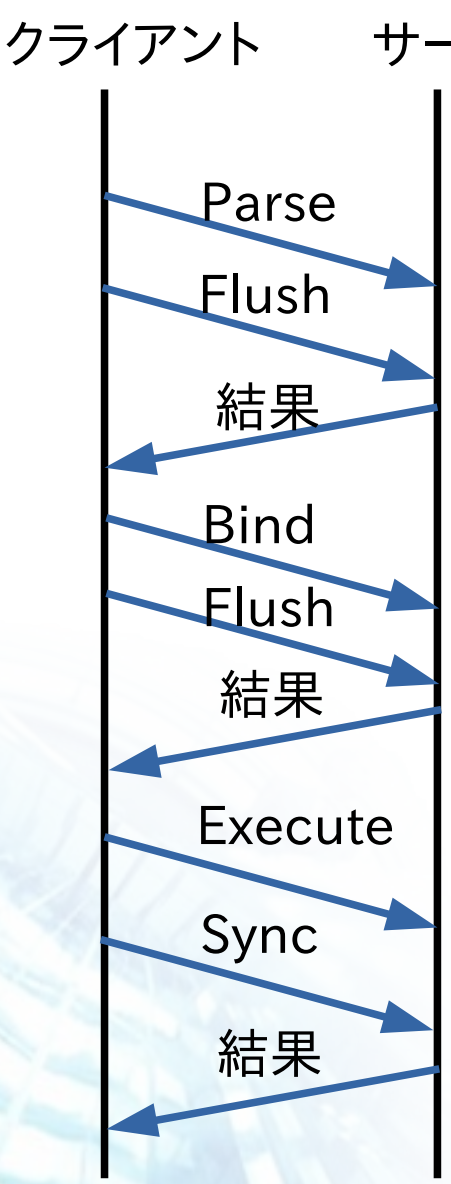
pgpool-II の拡張問い合わせ性能問題

- 今の pgpool-II は拡張問い合わせ使用時の性能が悪い
  - 最悪単純問い合わせ使用時の半分位の性能となってしまう



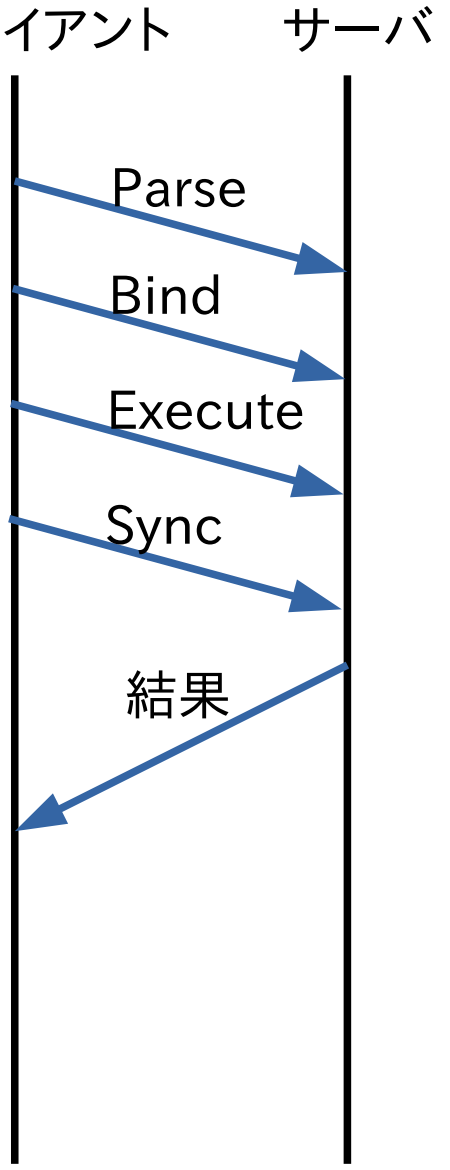
- 性能劣化の原因
  - Flush の発行回数が多い
  - PostgreSQL との通信が増えてしまう

拡張問い合わせ性能の改善



ストリーミングレプリケーションでは、Flushを省略できる場合がある

改良！

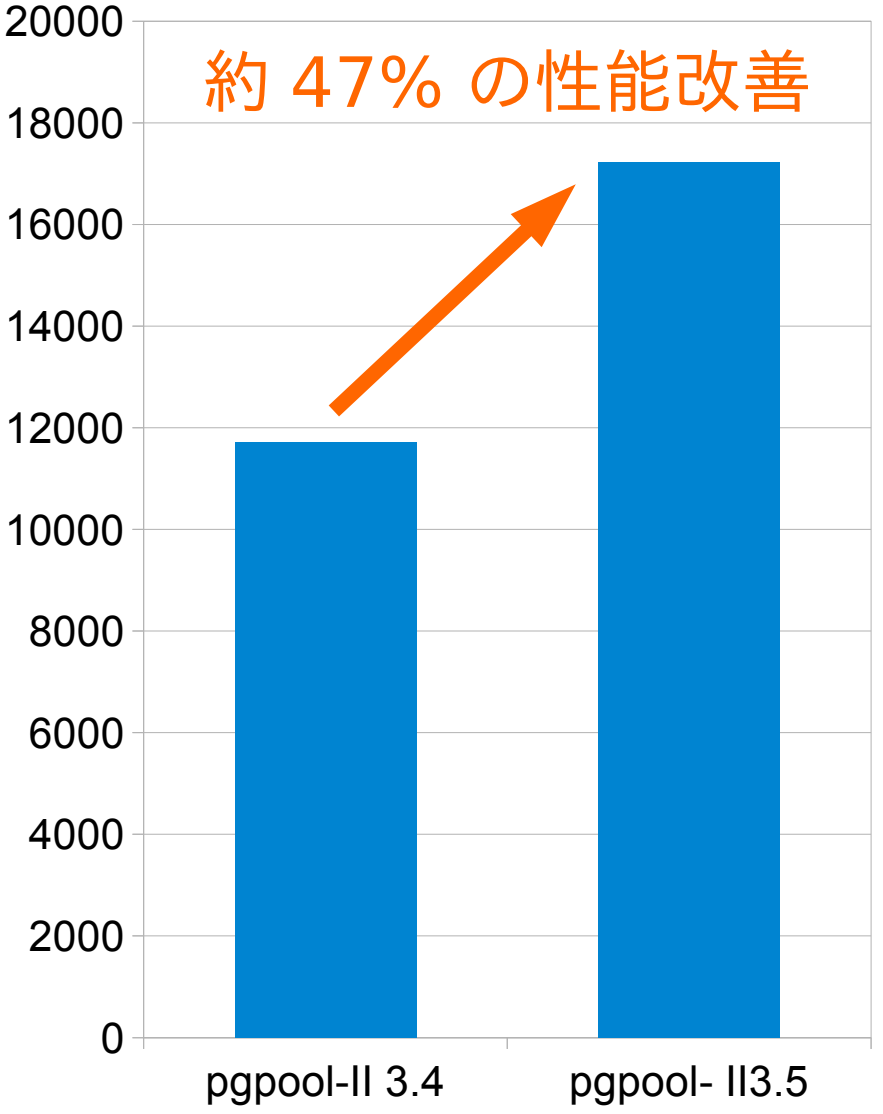


一部やり取りを省略しています

SRA OSS, INC.

性能比較

- pgbench による計測結果
  - 3.4 vs 3.5（開発中）
  - 1秒あたりの SELECT 処理回数 (TPS) を比較
  - バックエンドは2ノードのストリーミングプリケーション構成



Let's note CF-SX3
CORE i7 x 2@2.1GHz
Mem 16GB
SSD 512GB

pgbenchコマンドライン:
pgbench -S -n -M extended -c 8 -j 4 -T 30 test

まとめ

- pgpool-II 3.5 の新機能
  - PostgreSQL 9.5 のパーサ取り込み
    - クエリ振り分け、クエリキャッシュ、クエリ書き換えが新しい構文に対応
  - Watchdog 改善
    - ロバスト性の向上
    - プロセス間通信の変更
    - ノード間のパラメーター貫性
  - 性能改善
    - 拡張問い合わせ性能が 47% 向上
- この秋のリリースに向けて、活動中です！