

pgpool-IIで高可用性・高性能 システムを構築しよう

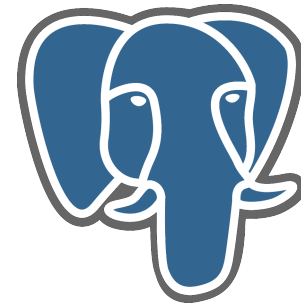
SRA OSS, Inc. 日本支社

取締役支社長

石井 達夫

SRA OSS, Inc.のご紹介

- 1999年よりPostgreSQLサポートを中心にOSSビジネスを開始、2005年に現在の形に至る
- OSSを利用したいユーザ/Slerに必要なサービスを提供
 - OSSサポート
 - PostgreSQLは24時間サポートを提供
 - 有効契約件数500以上の実績
 - 技術者トレーニングサービス
 - OSSの導入、設計コンサルティングサービス
 - <http://www.sraoss.co.jp>



PowerGres

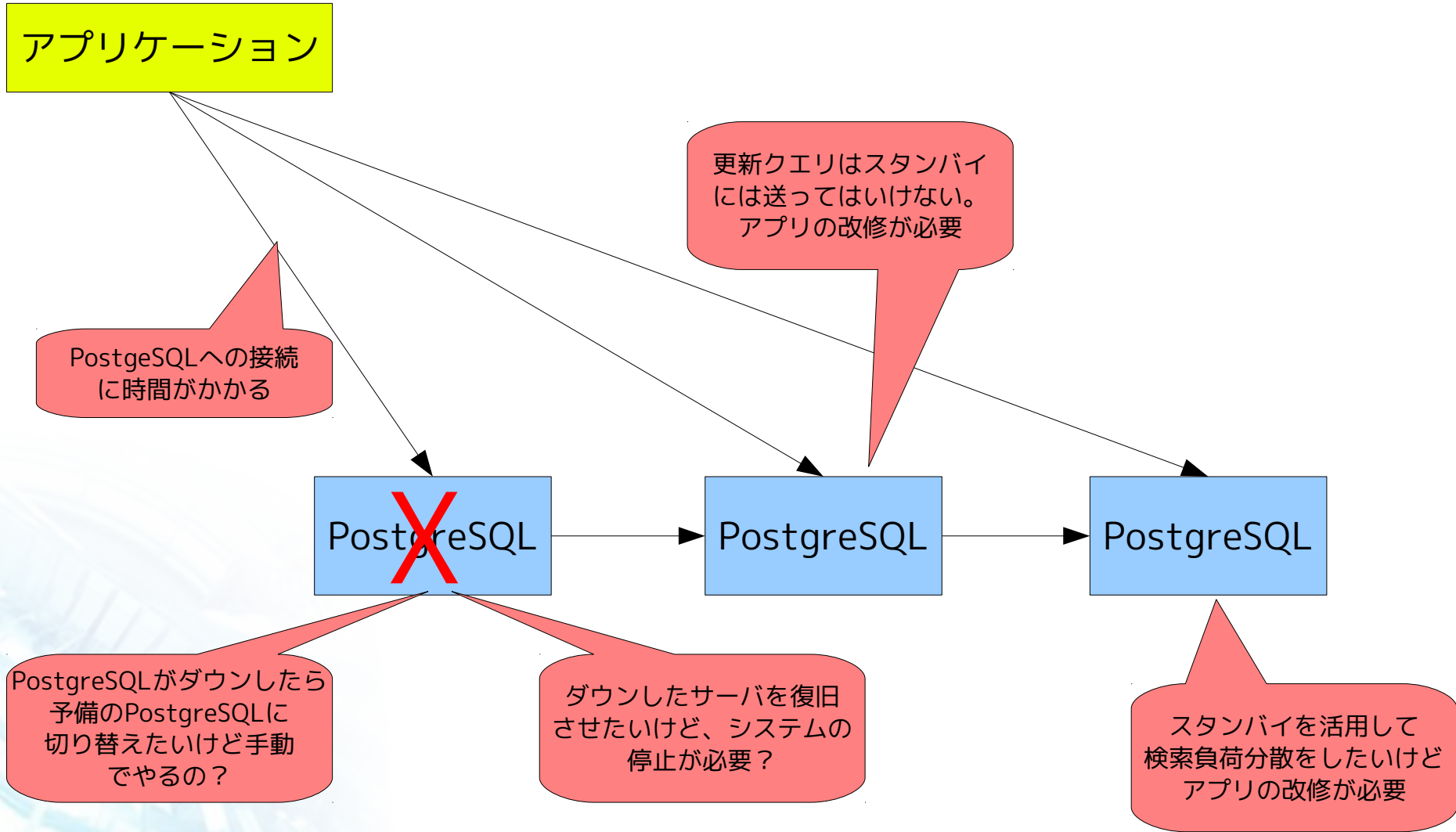


本日のメニュー

- pgpool-IIとは
- pgpool-II設定の勘所
- pgpool-IIの構成パターン
- シンプル2サーバ構成
- マルチマスタ的構成
- まとめ

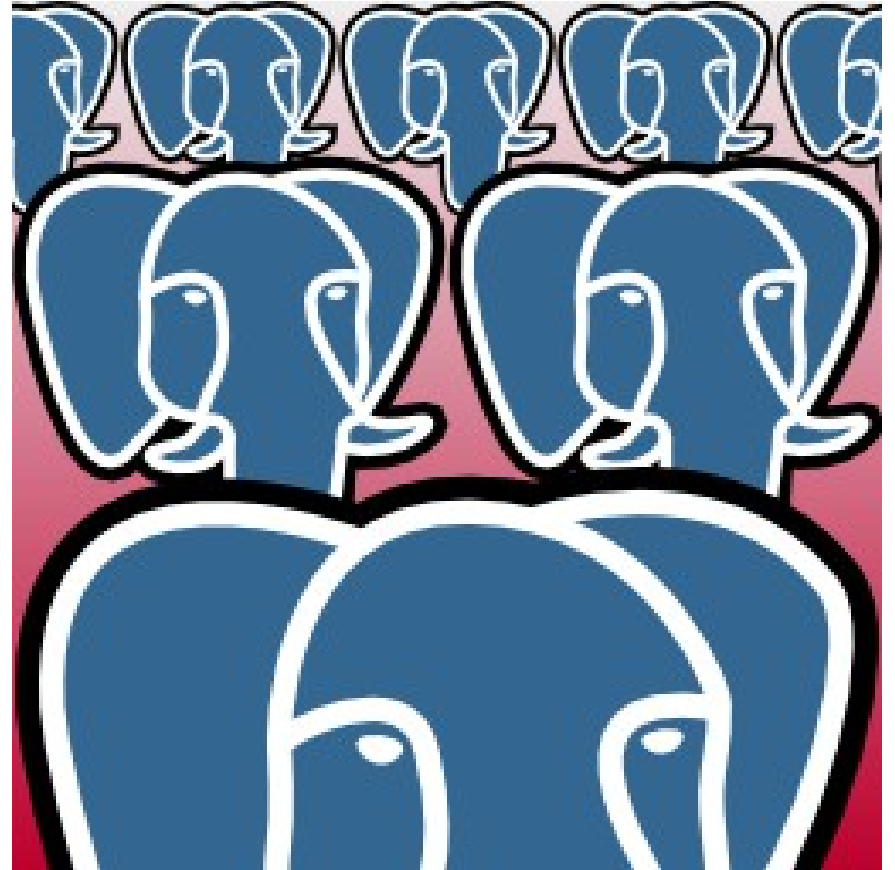


Why pgpool-II?

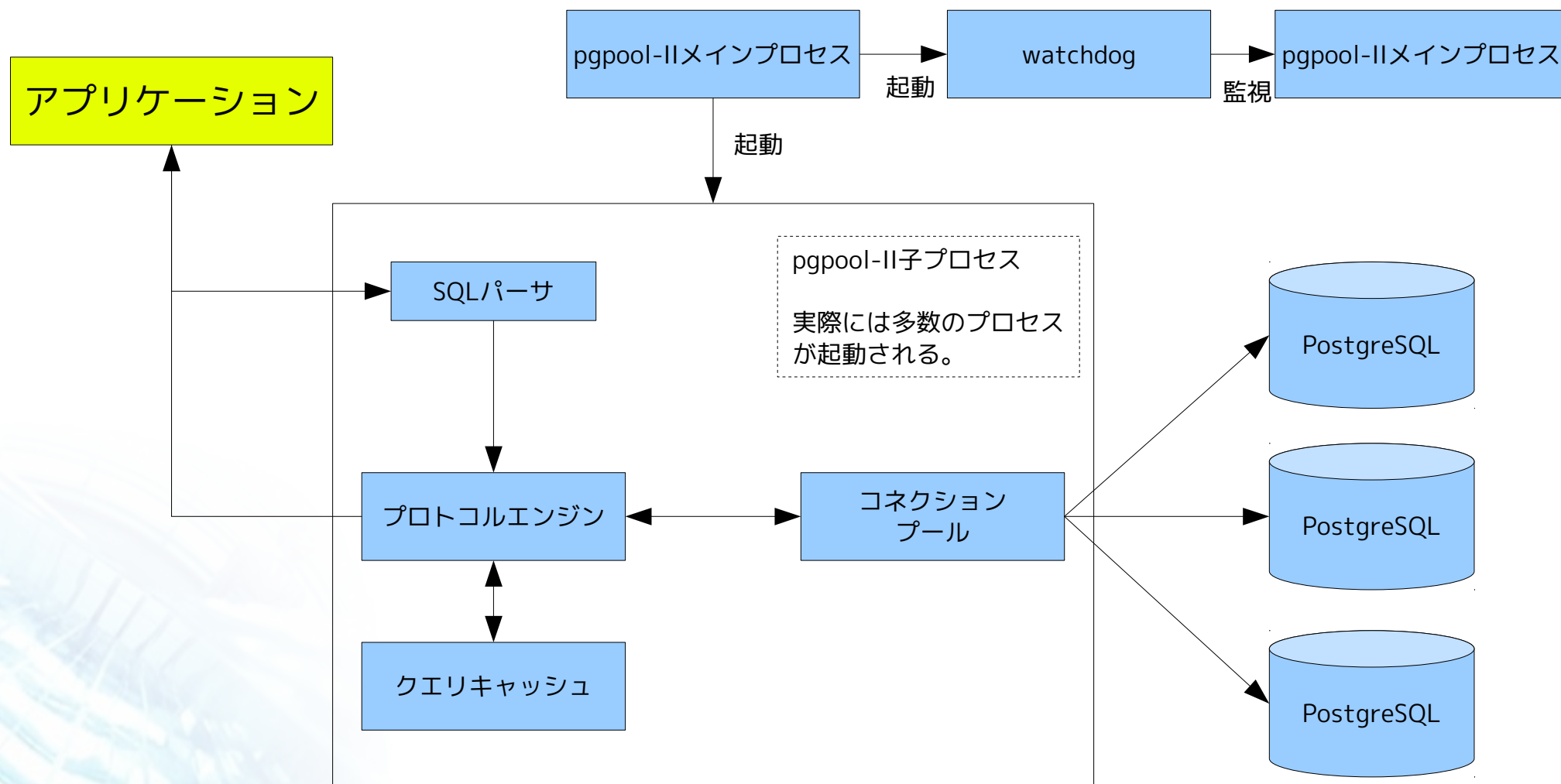


pgpool-IIとは

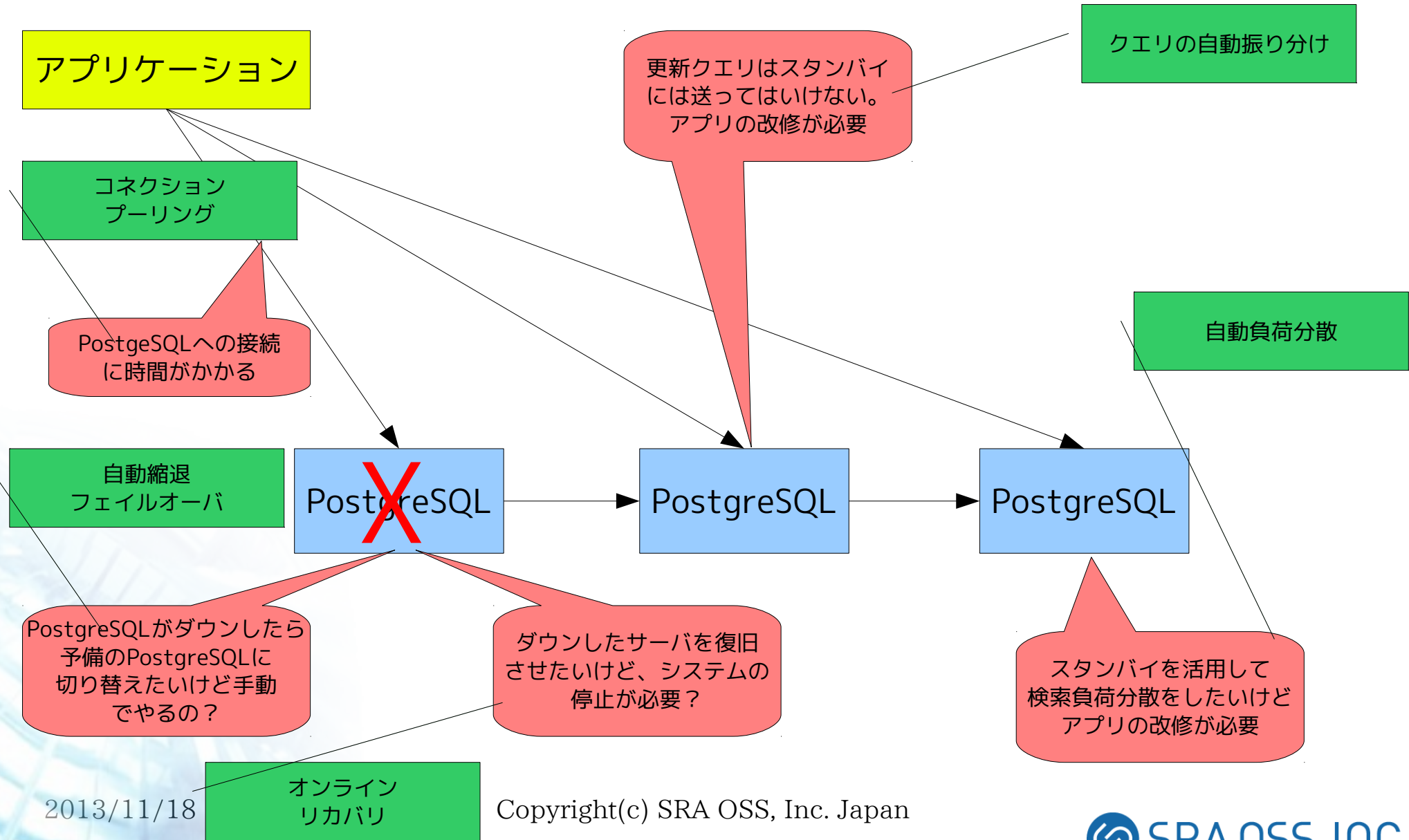
- PostgreSQL専用のクラスタリングミドルウェア
- OSSとして公開
- PostgreSQLから独立したproxyとして動作、アプリケーションの改修は最低限
- 性能や可用性を高める多くの機能
- 動作環境はLinux/UNIX (Windowsは未対応)。すべてC言語で書かれており、移植性は高い
- メジャーバージョンアップは年に1回、マイナーバージョンアップは2-3ヶ月に1回、最新バージョンは3.3.1



pgpool-IIの構造



pgpool-IIの機能

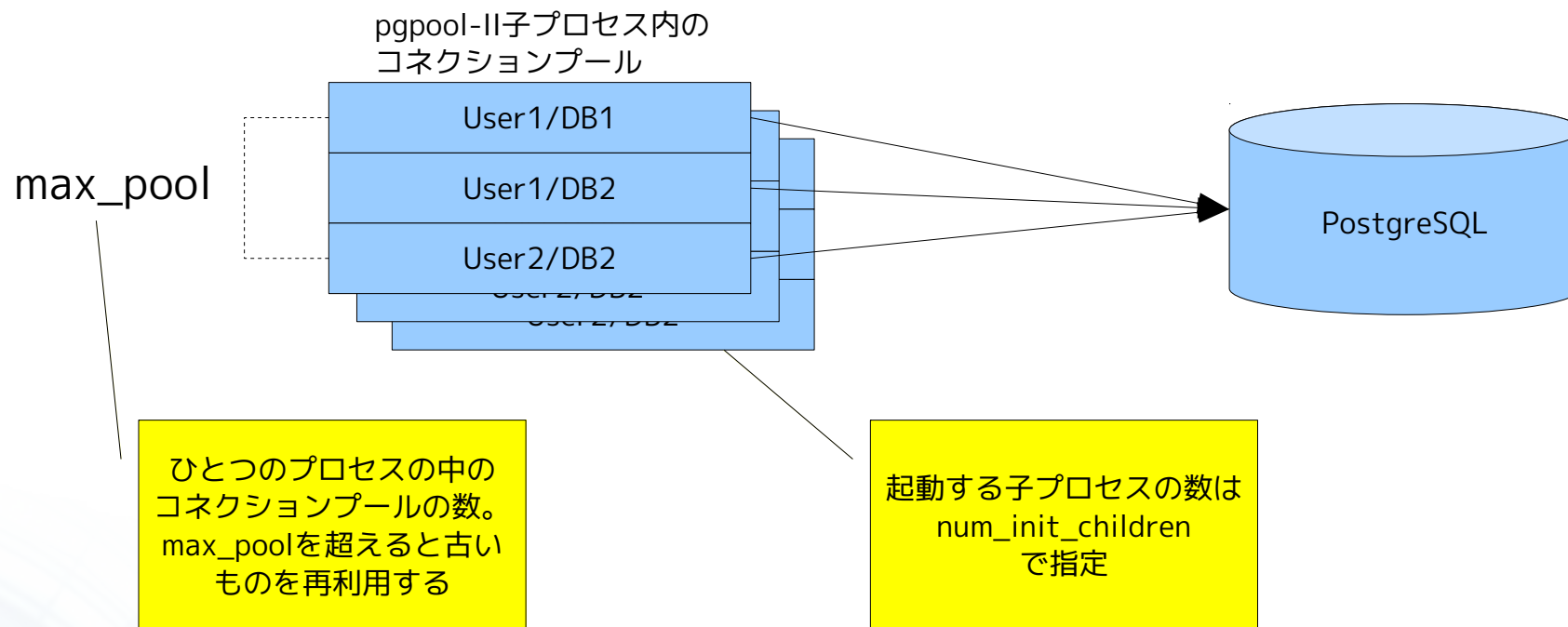


pgpool-II設定の勘所

ネイティブレプリケーションモード vs. ストリーミングレプリケーションモード

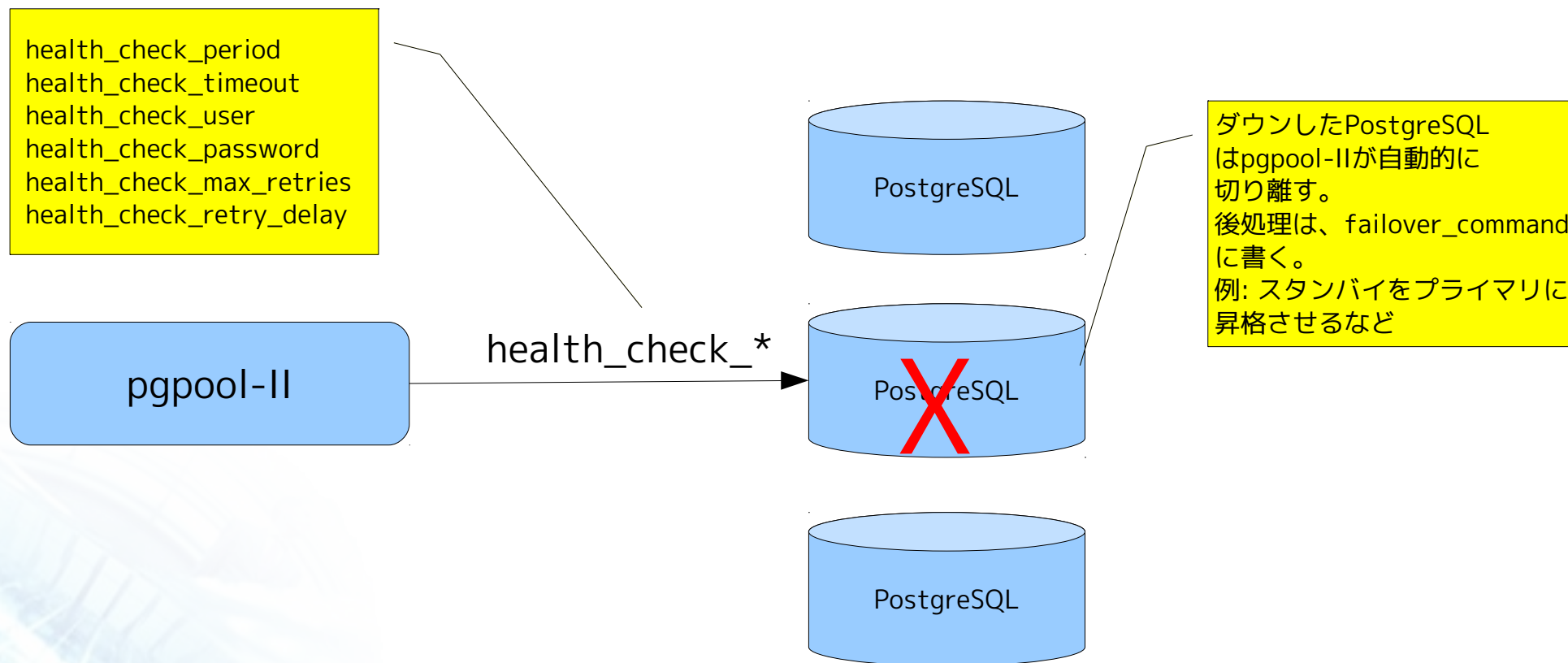
- ネイティブレプリケーションモード
 - 更新クエリをすべてのPostgreSQLに投げることでレプリケーションを行う
 - SQLの制限がある(たとえばINSERT INTO ... SELECT)
 - 更新の伝搬の遅延がない
 - 設定が簡単
 - 更新性能が50%に落ちる
- ストリーミングレプリケーションモード
 - レプリケーションはPostgreSQLのストリーミングレプリケーションに任せる。pgpool-IIIは、更新クエリをプライマリサーバにのみ投げる
 - SQLの制限が少ない
 - 更新の伝搬の遅延がある
 - 設定が複雑
 - 更新性能はあまり落ちない(ストリーミングレプリケーションの設定次第)
- 更新遅延が問題になる場合はネイティブレプリケーションモード、それ以外はストリーミングレプリケーションモードがお薦め
- どちらのモードでも、watchdog、検索負荷分散、クエリキャッシュなどの主要な機能が使える

まずはコネクション関係の設定



以下の関係を満たすようにする(クエリキャンセルを考慮する場合は左辺を倍にする)
$$\text{num_init_children} * \text{max_pool} \leq (\text{max_connections} - \text{superuser_reserved_connections})$$

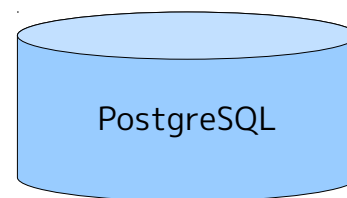
フェイルオーバー関連設定



オンラインリカバリ設定

リカバリ時に起動されるコマンドを
ユーザ定義できる(サンプル提供あり)

- プライマリのベースバックアップを取る
- リカバリされる側にリストアする
- recovery.confを設定する
- リカバリされる側のPostgreSQLを起動
- バックアップ/リカバリ中に行われた更新を
ロールフォワードして同期する



リカバリ中も
更新処理を受け付ける
ことができる

オンラインリカバリ関連の
パラメータ

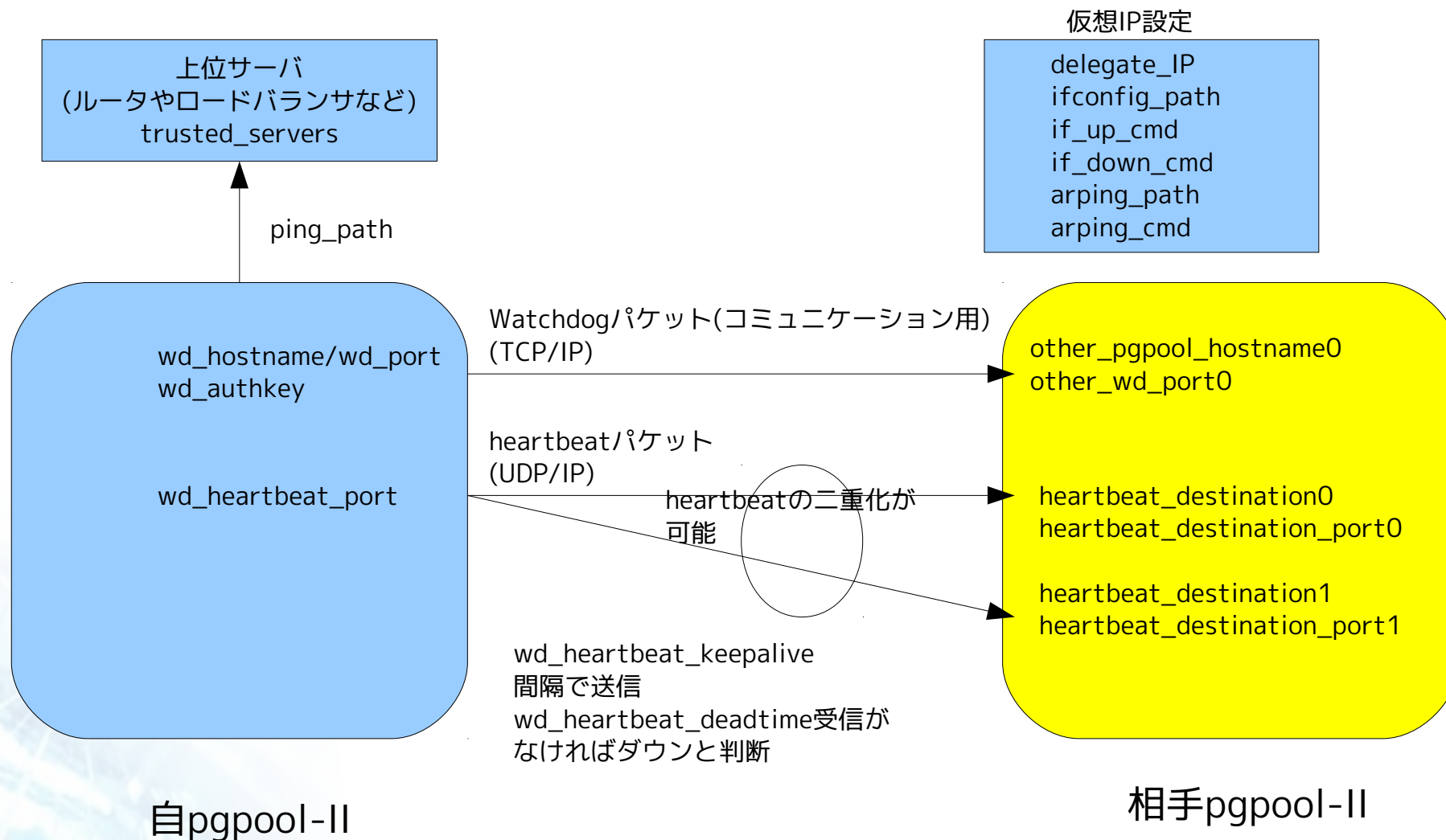
```
recovery_user
recovery_password
recovery_1st_stage_command
recovery_2nd_stage_command
recovery_timeout
```

同じ方法で新しいPostgreSQLを
運用を止めずに追加できる。
負荷に応じてPostgreSQLを追加
する仕組みを作れば
自動スケールアウトも可能。

watchdogとは

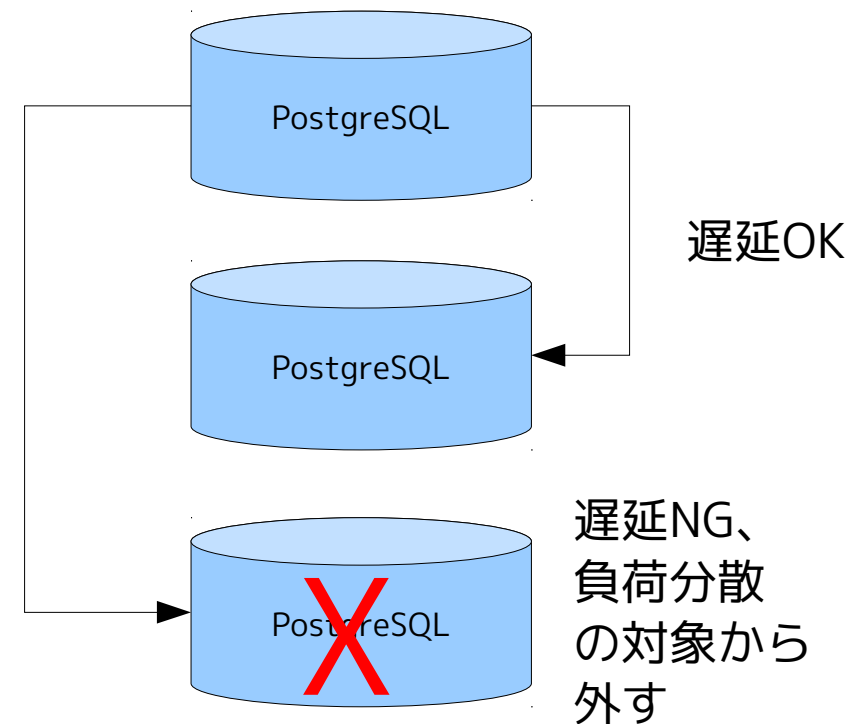
- pgpool-II組み込みのHA(High Availability)機能+複数pgpool-IIの連動機能
- 稼働系のpgpool-IIがダウンした時に待機系のpgpool-IIがVIPを引き継ぐことによってpgpool-II自体の可用性を向上させることができる
- pgpool-II組み込みなので、導入が簡単
- watchdog自体は、pgpool-IIから起動されるプロセスとして動く
- watchdogが対応可能な障害
 - pgpool-IIが動いているサーバ自体がダウンした時
 - pgpool-II親プロセスがダウンした時
 - watchdogプロセス自体がダウンしたら自動再起動される
- DB障害はpgpool-II親プロセスが監視
- その他watchdogがやること
 - オンラインリカバリ、フェイルオーバー処理の排他制御
 - DBステータスの同期

watchdog設定項目



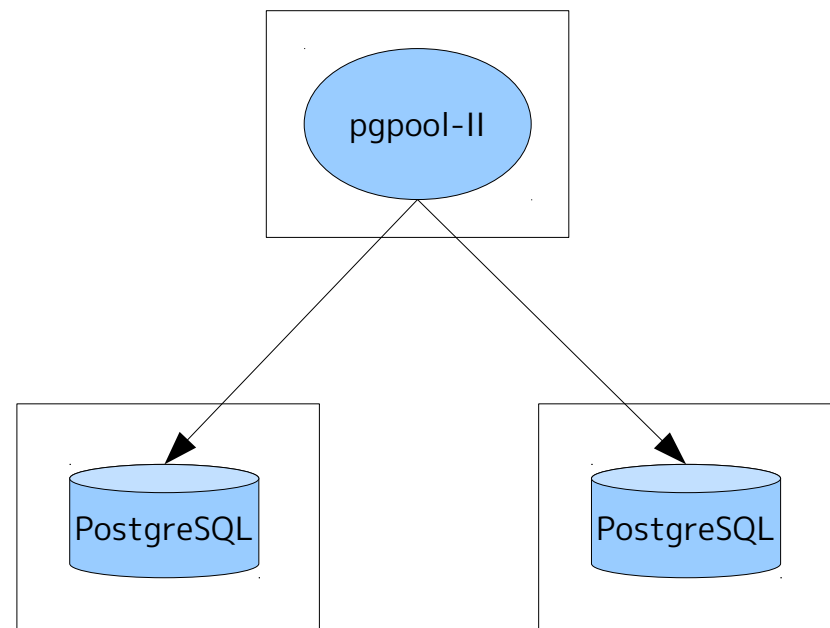
レプリケーション遅延の監視

- ストリーミングレプリケーションモードにおいて、レプリケーションの遅延を定期的に監視
- 遅延がしきい値を超えたら負荷分散の対象から外す、ログを出力する、などの対応が可能
- `sr_check_period`
- `sr_check_user`
- `sr_check_password`
- `delay_threshold`



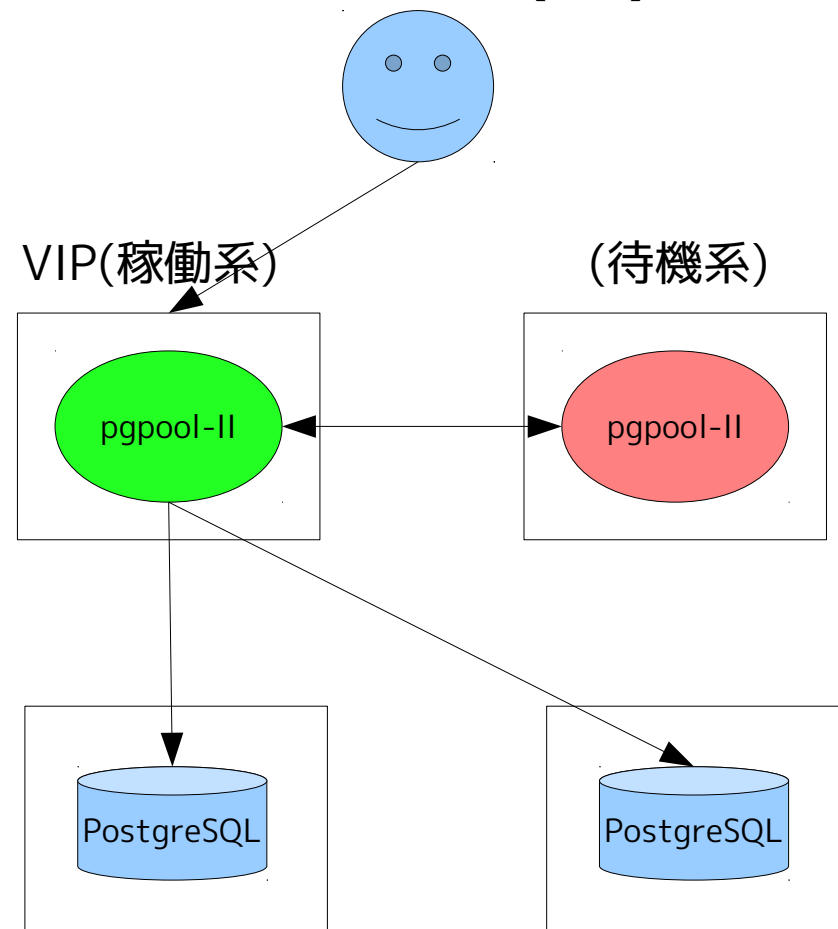
pgpool-IIの構成パターン(1)

- pgpool-II用に1台、PostgreSQL用に2台のサーバを用意(3台以上も可能)
- シンプルでわかりやすい
- pgpool-IIがダウンすると、単一障害点(SPOF: Single Point Of Failure)になる



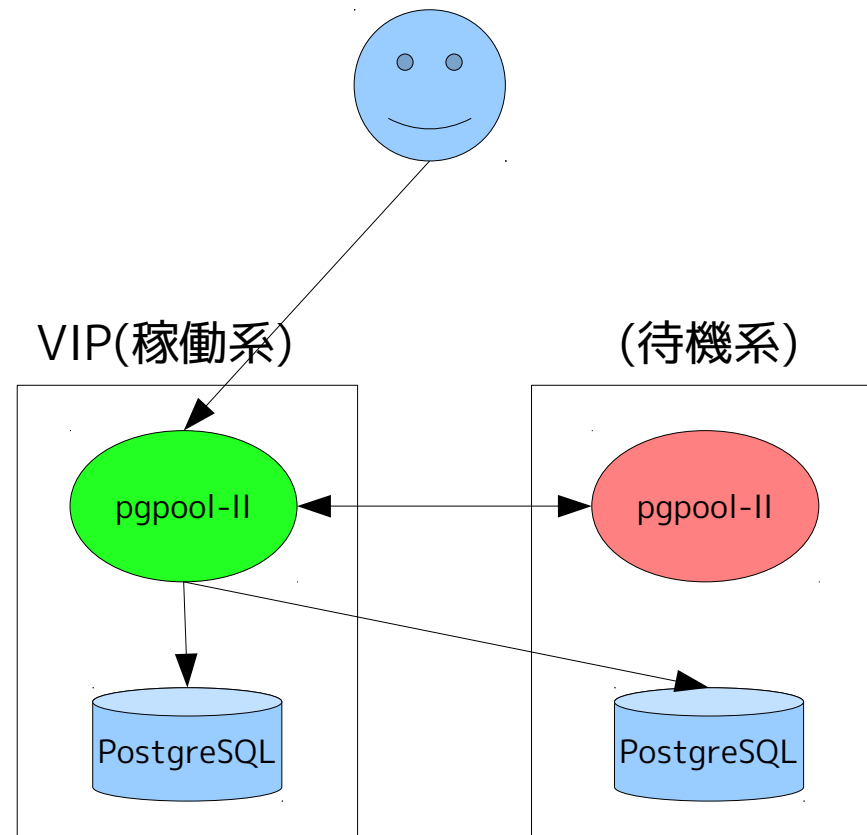
pgpool-IIの構成パターン(2)

- pgpool-II用に2台、PostgreSQL用に2台のサーバを用意(3台以上も可能)
- pgpool-IIを冗長化することによって、単一障害点を克服
- VIP経由でpgpool-IIにアクセス
- サーバが4台必要



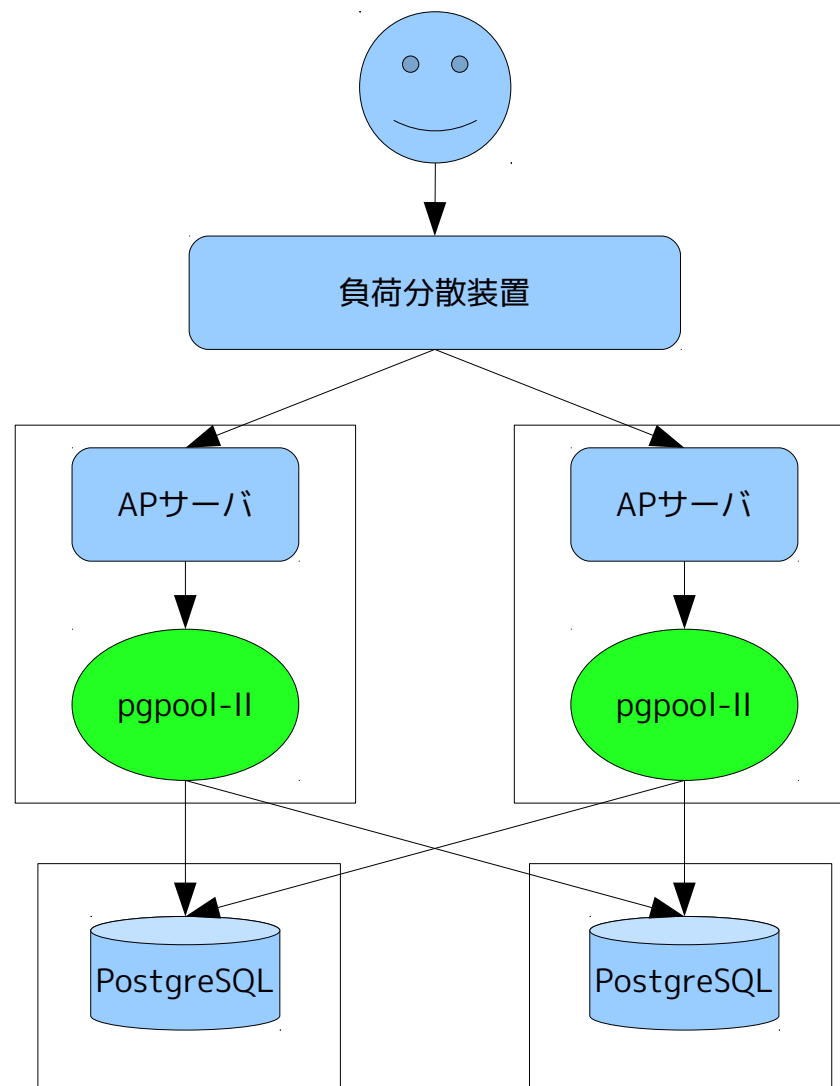
pgpool-IIの構成パターン(3)

- pgpool-II用とPostgreSQLを同一サーバに同居
- pgpool-IIを冗長化することによって、単一障害点を克服
- VIP経由でpgpool-IIにアクセス
- サーバが2台で済む
- 費用対効果、可用性重視の方にお勧め
- 今回お勧めパターンの一つ



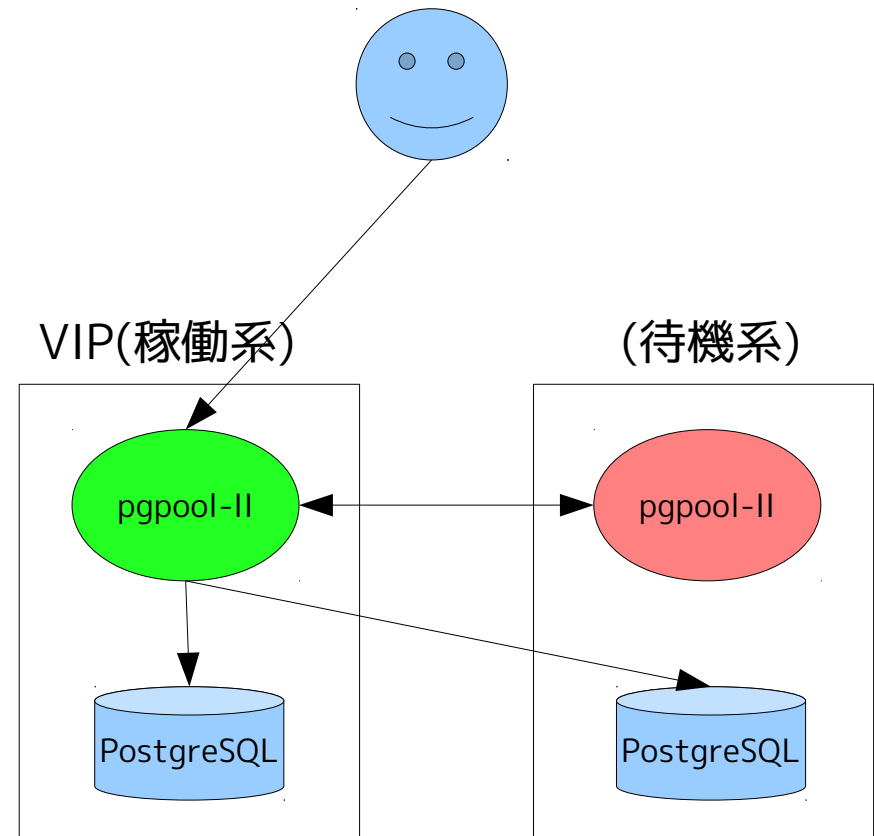
pgpool-IIの構成パターン(4)

- APサーバとpgpool-IIを同居させたペアを複数用意し、どのペアにもアクセスできるマルチマスタ的構成
- pgpool-IIを冗長化することによって、単一障害点を克服
- APサーバとpgpool-IIのペアを増やしてAPサーバの処理性能をスケールアウトできる
- PostgreSQLの数を増やしてDB検索性能をスケールアウトできる
- 性能、可用性重視の方にお勧め
- 今回お勧めパターンの一つ
- Pgpool-IIをPostgreSQL側に置く構成も考えられるが、APサーバとPostgreSQLの間の通信量が多いので、この構成のほうが有利



pgpool-IIインストーラ

- 構成パターン(3)の導入にはpgpool-II 3.3から提供されているインストーラを利用するのがお薦め
- 質問に答えていくだけで、適切な設定が完了する
- インストーラが設定する構成
 - ストリーミングレプリケーションモードまたはネイティブレプリケーションモードが選択できる
 - フェイルオーバースクリプト
 - オンラインリカバリスクリプト
 - ストリーミングレプリケーションの設定
 - クエリキャッシュの有無
 - PgpoolAdmin (管理用GUI)



構成パターン(3)

まとめ

- pgpool-IIの概要、構造、機能
- pgpool-II設定の勘所
 - ネイティブレプリケーションモード vs. ストリーミングレプリケーションモード
 - コネクション関係の設定
 - フェイルオーバ関連設定
 - オンラインリカバリ設定
 - Watchdog
- pgpool-IIの構成パターン
 - シンプル2サーバ構成
 - マルチマスタ的構成
 - その他



参考URL

- pgpool-IIオフィシャルサイト
 - <http://www.pgpool.net>
- PostgreSQLエンタープライズコンソシアム(2012年度の実施報告書にpgpool-IIのベンチマークあり)
 - <http://www.pgecons.org>
- Let's Postgres (pgpool-II関連の記事あり)
 - <http://lets.postgresql.jp/>

ご清聴ありがとうございました