

並列分散問い合わせを実現した pgpool-IIのご紹介

SRA OSS, Inc. 日本支社

技術部

浅羽 義之

y-asaba@sraoss.co.jp

PostgreSQLに対する要求

- 1 台では処理能力が限界にきている
 - システムのボトルネックになっている！
 - ▶ コネクションプーリングによる負荷の軽減
 - ▶ クラスタリングによる負荷分散
- 信頼性の向上
 - システムを止めたくない！
 - ▶ レプリケーションによるデータ冗長化

これらの要求に答えるため、pgpoolを開発

pgpoolとは？

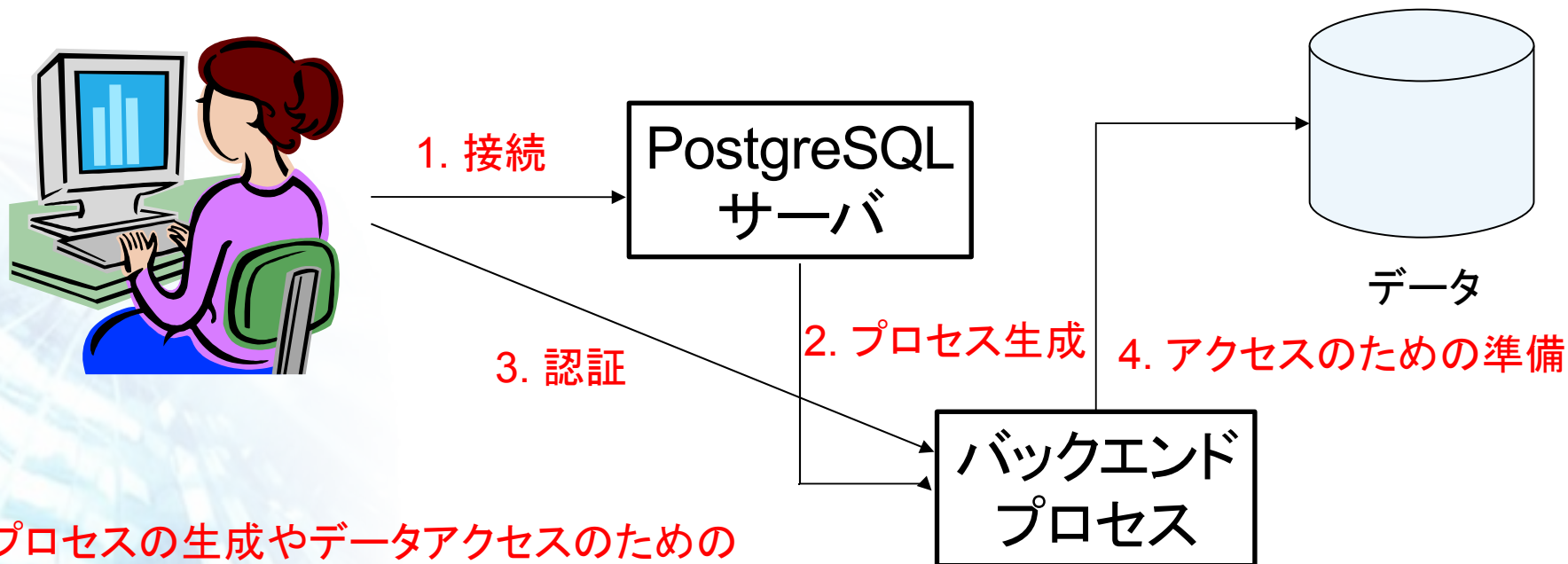
- 2003年開発スタート
 - 弊社石井が個人で開発
 - 最初はコネクションプーリング機能のみ(pgpoolという名前の由来はここからきている)
- ライセンス
 - BSDライセンス
 - ソースコードが公開され、自由に改造してもよい
- 現在の開発体制
 - pgpool Global Development という開発団体に移行

pgpoolの機能

- コネクションプーリング
 - 接続時のオーバーヘッドを軽減
- レプリケーション(データの複製を作る)
 - 2台までの制限あり
- ロードバランス
 - レプリケーション設定時にのみ検索を振り分けることが可能

コネクションプーリングの利点

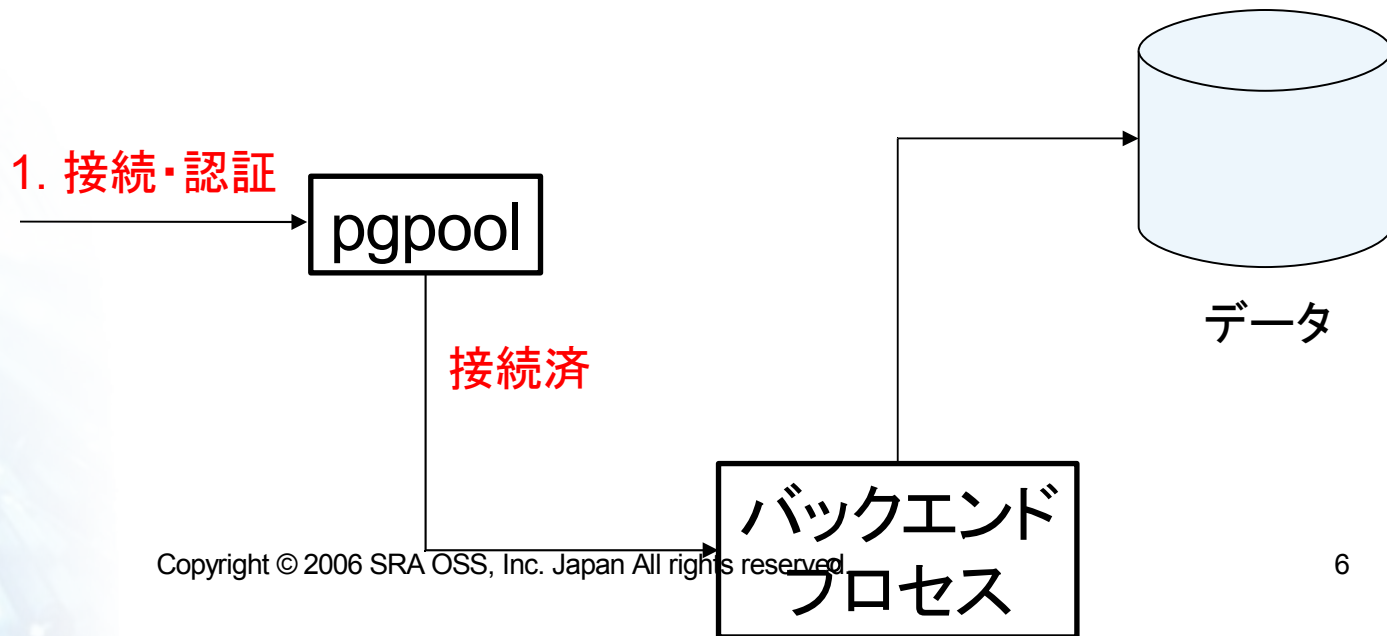
- 接続オーバーヘッドの軽減
 - PostgreSQLの接続処理の流れ



プロセスの生成やデータアクセスのための
初期化処理は結構重い

コネクションプーリングの利点

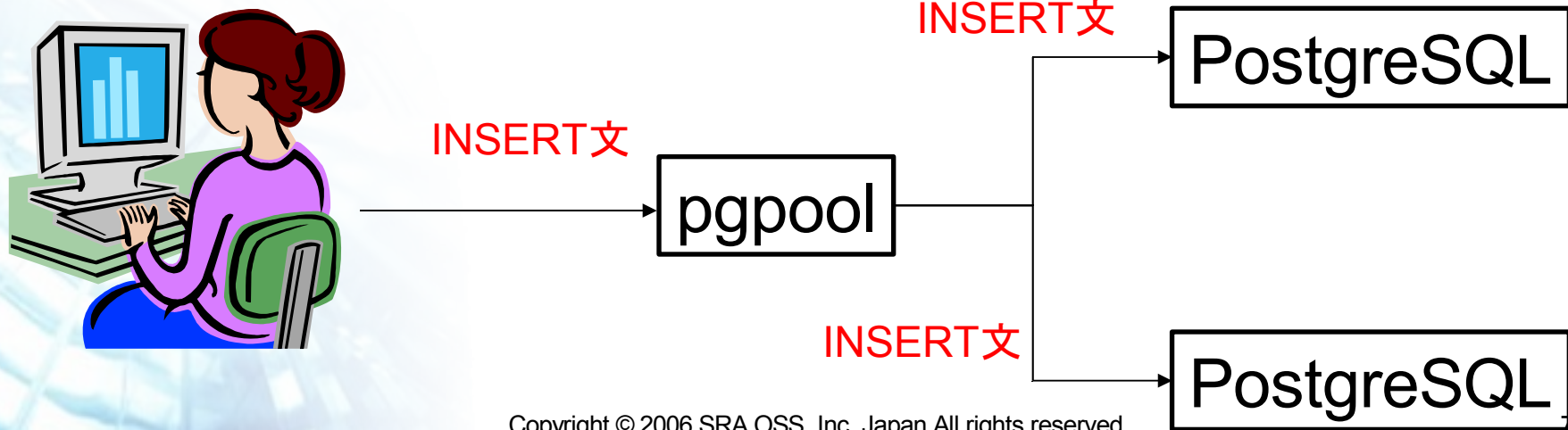
- 接続オーバーヘッドの軽減
 - pgpoolによるコネクションプールの接続処理の流れ
 - クライアントとPostgreSQLの間にpgpoolが入る
 - プロセス生成とディスクアクセスのための準備処理が省略可能



レプリケーションの利点

- 信頼性向上

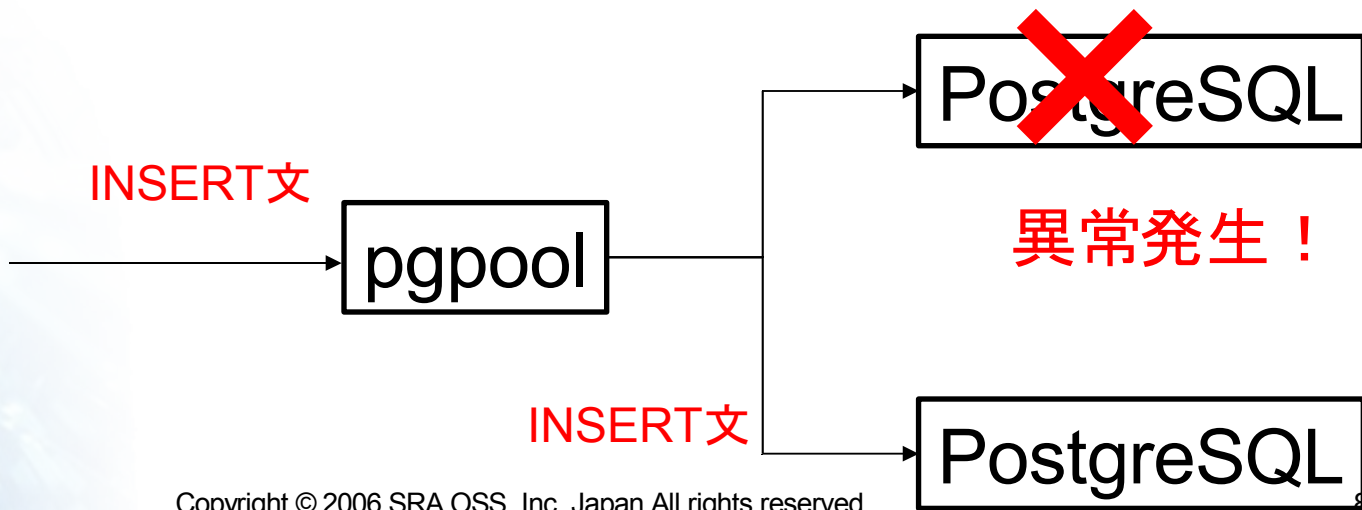
- 2台に更新系クエリを送信することで、データの同期を取る
 - 異常が発生した場合にも運用が可能(縮退運転)
 - 常に同期を取るため、通常より更新系クエリは時間がかかる



レプリケーションの利点

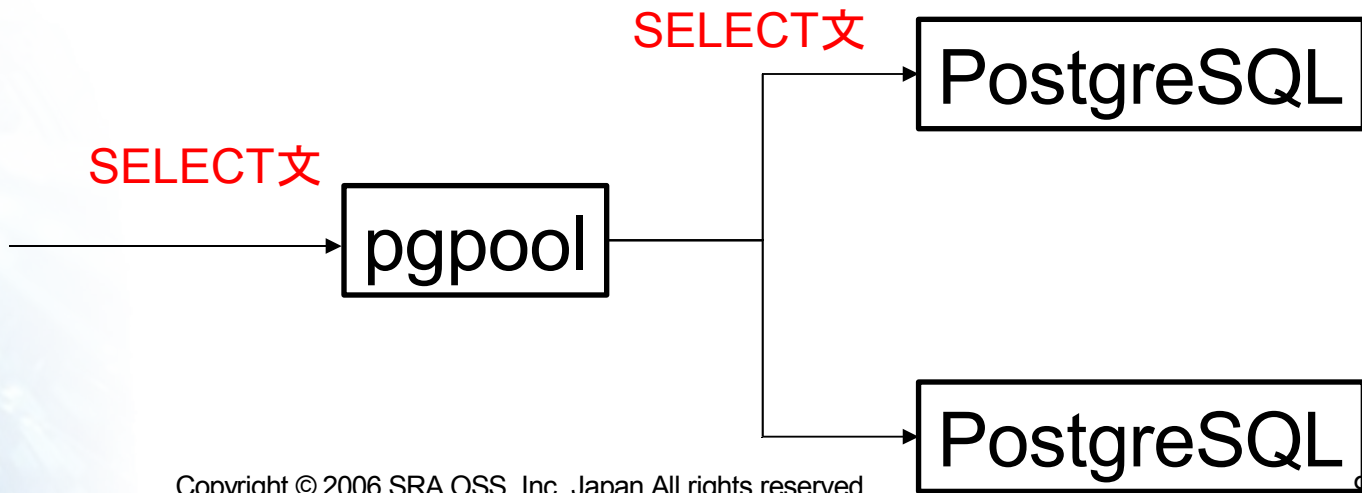
- 信頼性向上

- 2台に更新系クエリを送信することで、データの同期を取る
 - 異常が発生した場合にも運用が可能(縮退運転)
 - 常に同期を取るため、通常より更新系クエリは時間がかかる



レプリケーションの利点

- ロードバランス
 - 同じデータを2台で持つので、検索についてはどちらかに問い合わせすればよいので、負荷分散が可能
 - ロードバランス先の割合を指定可能



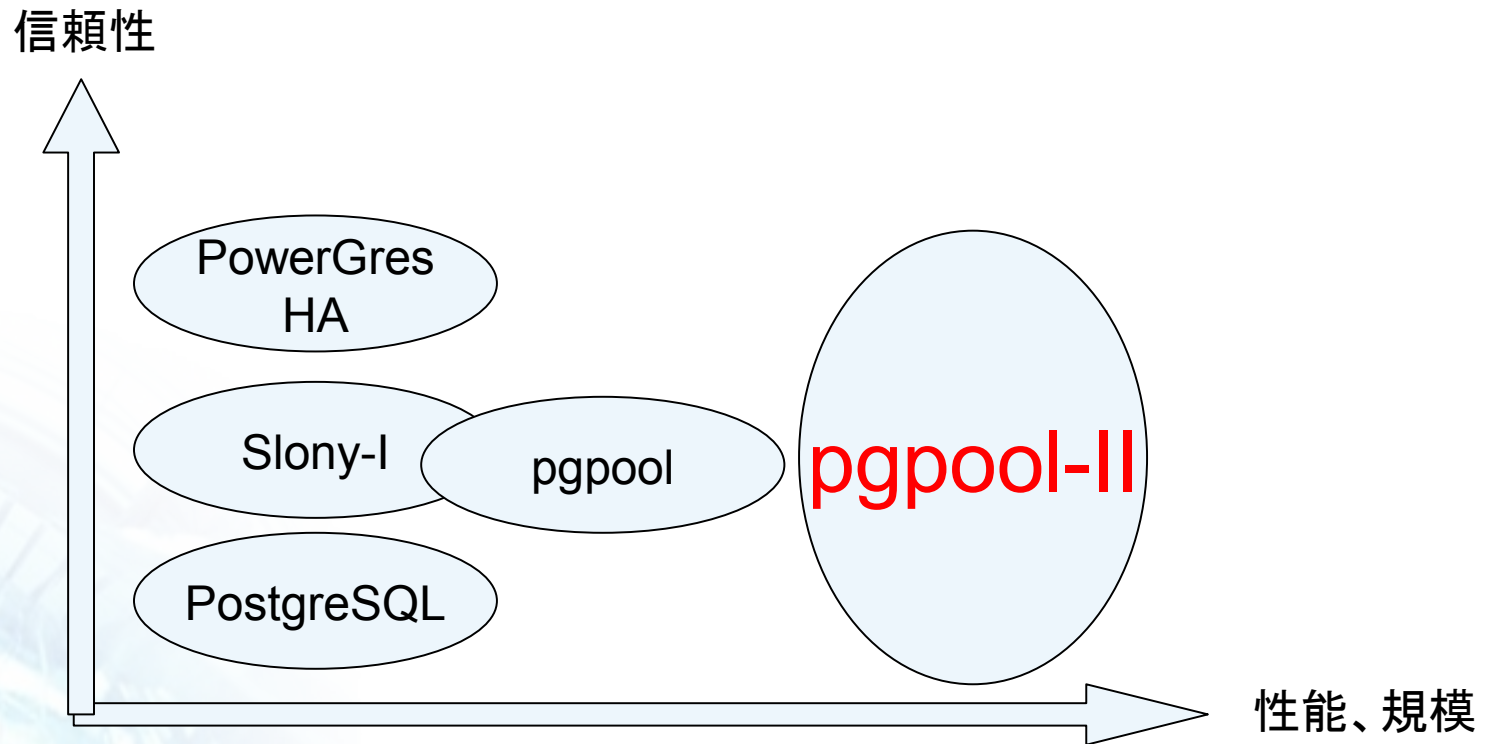
pgpool-IIとは？

- IPA「オープンソースソフトウェア活用基盤整備事業」の支援により SRA OSS, Inc. 日本支社で開発
- 現在は pgpool Global Development にソースコードを寄贈
 - 弊社の開発メンバーも何人か所属
 - ライセンスはpgpoolと同じ

pgpool-IIの機能

- pgpoolの後継プロダクト
 - pgpoolの機能を継承
 - レプリケーションを2台までしかできない制限を解除
- pgpool-II新機能
 - パラレルクエリ
 - クエリキャッシュ
 - 管理ツールによるpgpoolの管理

pgpool-IIのターゲット市場



パラレルクエリ

- データを複数サーバに分散して配置
 - 複数のサーバに配置されたテーブルを仮想的に1つのテーブルとして扱う
 - 分散ルールはユーザが定義



分散ルールを
格納

INSERT文

PostgreSQL

pgpool

分散ルールから
振り分け先を決定

INSERT文

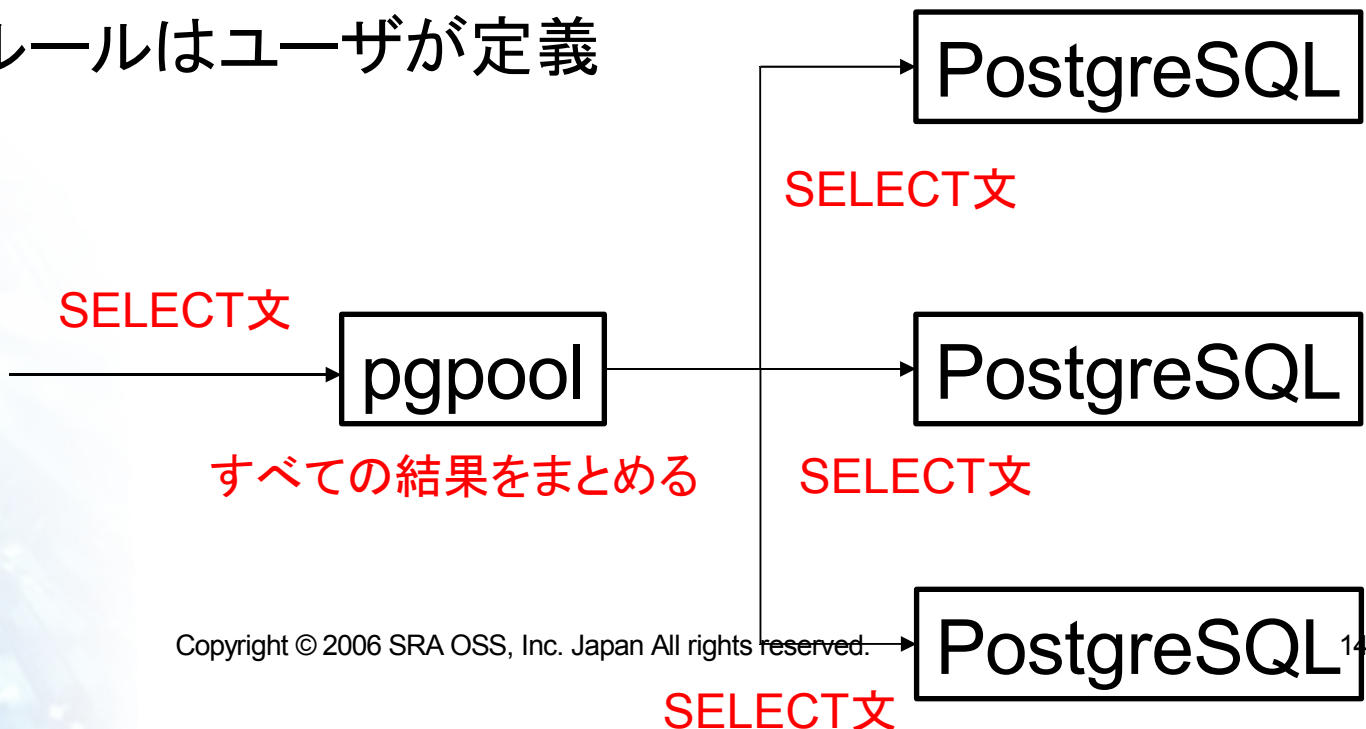
PostgreSQL

PostgreSQL

PostgreSQL

パラレルクエリ

- データを複数サーバに分散して配置
 - 複数のサーバに配置されたテーブルを仮想的に1つのテーブルとして扱う
 - 分散ルールはユーザが定義



パラレルクエリの利点・欠点

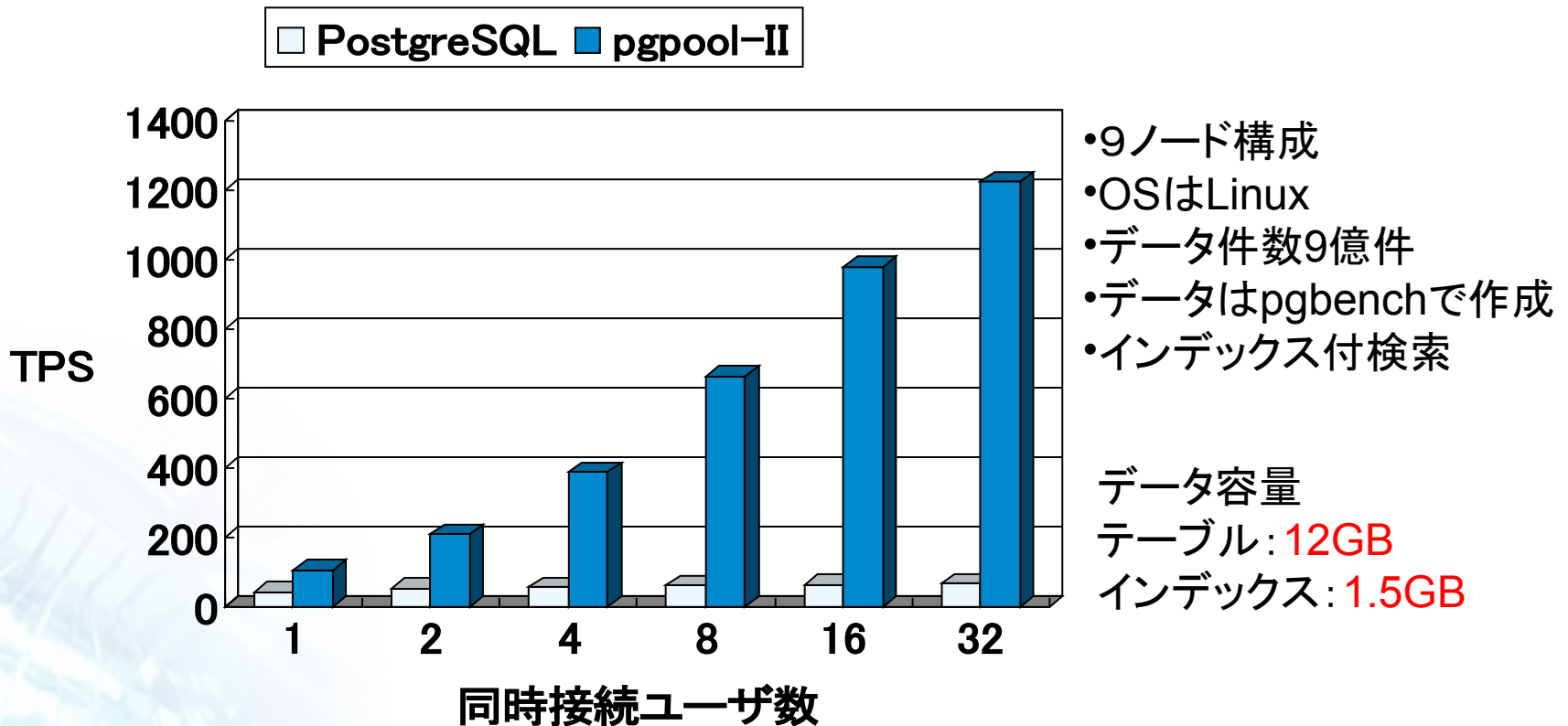
- 利点

- 大容量のテーブルへの検索の高速化
 - PostgreSQL単体の場合、I/Oの負荷がすべて集中する

- 欠点

- データ量が多くないとそれほど効果があがらない
 - pgpoolで検索結果の統合やクエリの書き換え処理などを実行するため、一台で十分なパフォーマンスが出ている場合はpgpoolが入ることによりオーバヘッドが増える

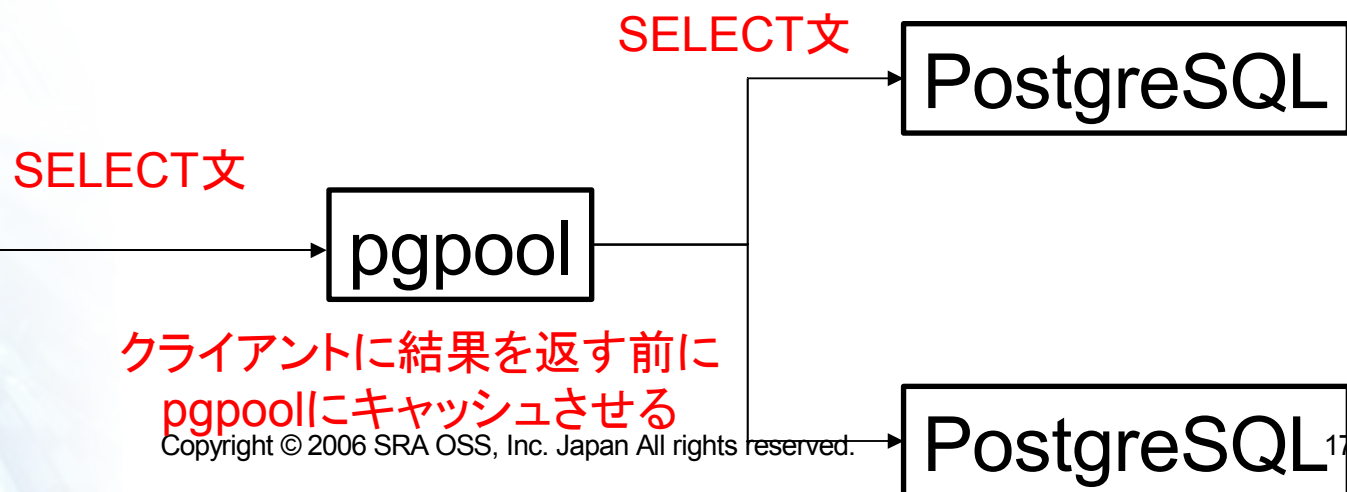
pgpool-IIのパラレルクエリの効果



最大18倍(32接続の場合)

クエリキャッシュ

- 問い合わせた結果をpgpoolにキャッシュさせる
 - 同じ問い合わせの場合、PostgreSQLには問い合わせず、pgpoolが直接結果を返す
 - 制限事項: キャッシュを自動的に削除する仕組みを実装していない



クエリキャッシュ

- 問い合わせた結果をpgpoolにキャッシュさせる
 - 同じ問い合わせの場合、PostgreSQLには問い合わせず、pgpoolが直接結果を返す
 - 制限事項: キャッシュを自動的に削除する仕組みを実装していない



SELECT文

pgpool

PostgreSQL

クライアントにキャッシュ
されている結果を返す

Copyright © 2006 SRA OSS, Inc. Japan All rights reserved.

PostgreSQL 18

pgpool管理ツール

- PHPによるWebベースの管理ツール
 - 設定ファイルの編集
 - 統計情報の表示
 - ノード管理
 - ステータス表示
 - ノードの追加、切り離し

pgpoolAdmin

pgpool Administration Tool

▶ pgpoolステータス

▶ ノードステータス

▶ クエリキャッシュ

▶ 分散ルール

▶ pgpool.conf設定

▶ 管理ツール設定

▶ パスワード変更

▶ ログアウト

[ヘルプ](#)

pgpoolステータス

[サマリー](#) [プロセス情報](#) [ノード情報](#)

ノード情報

IPアドレス	ポート	ステータス
192.168.187.11	5432	ノード稼働中。接続有り
192.168.187.12	5432	ノード稼働中。接続有り
192.168.187.13	5432	ノード稼働中。接続有り

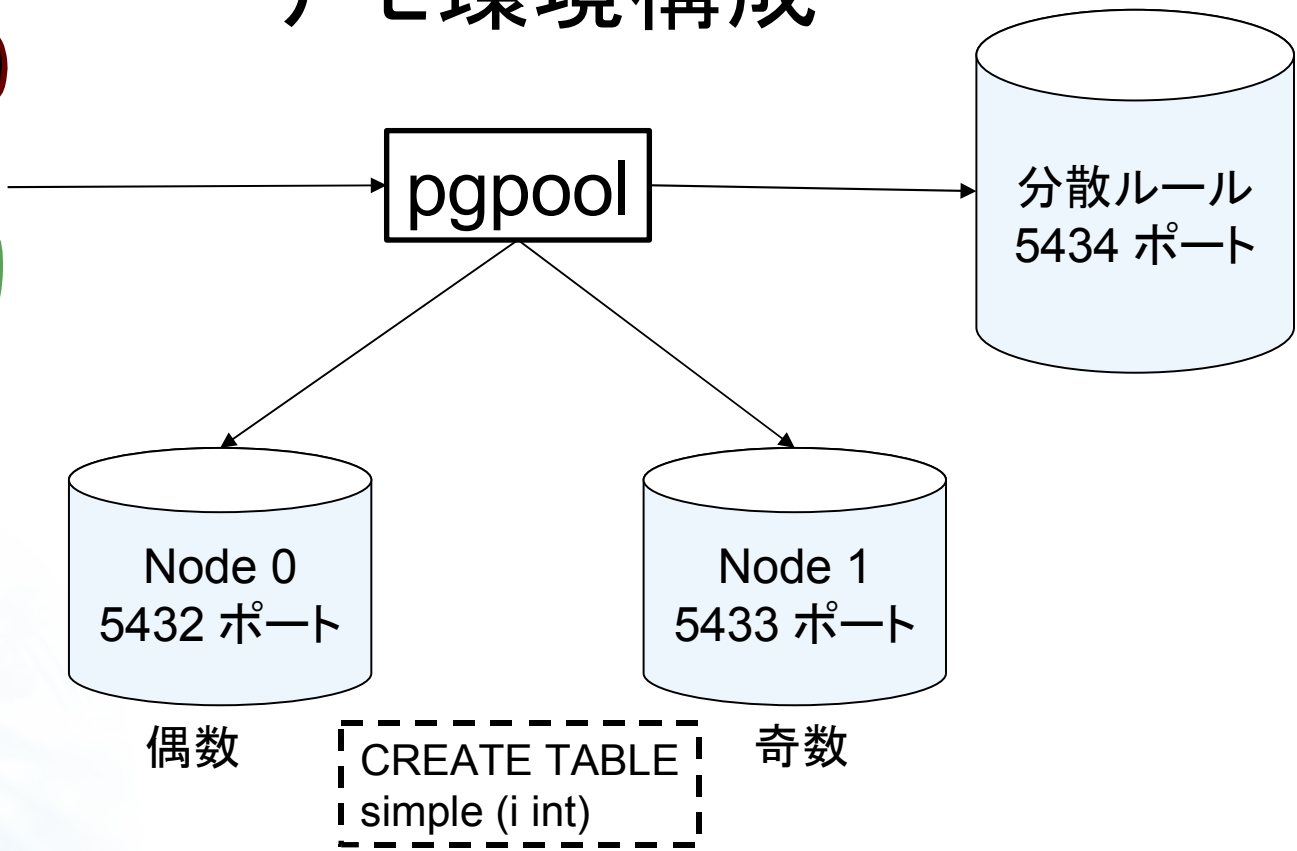
[サマリー](#) [プロセス情報](#) [ノード情報](#)

pgpool

[pgpool停止](#) [pgpool再起動](#)

デモを行います

デモ環境構成



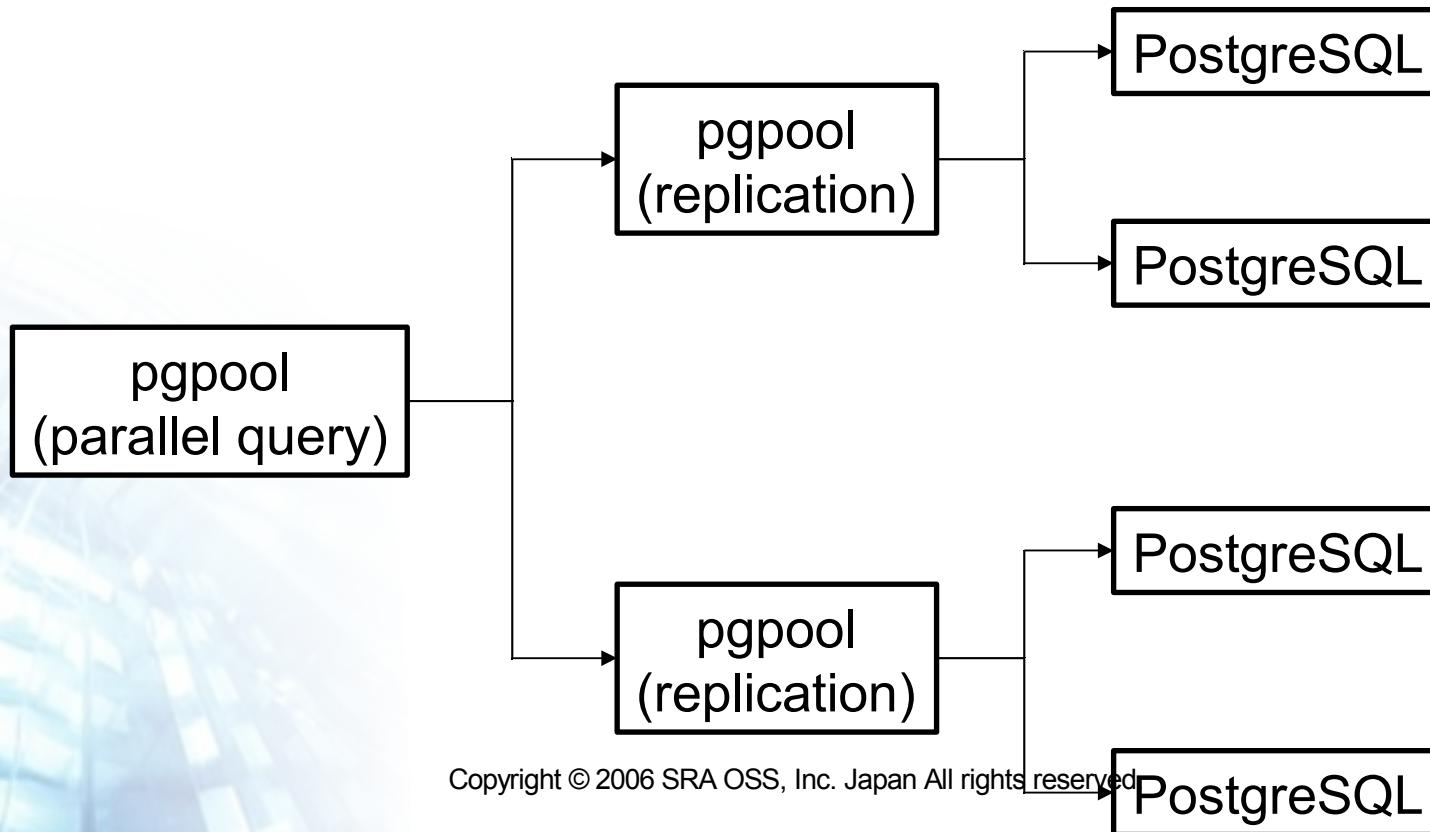
偶数か奇数かでデータを振り分ける

構成例

- データ量がそれほど多くない場合
 - レプリケーション+ロードバランス構成
- データ量が多く、検索が遅い場合
 - パラレルクエリ

パラレルクエリ+レプリケーション

- 分散させたデータを冗長化する構成例



制限事項

- オンラインリカバリができない
 - ノード間のデータの再同期には、PostgreSQLを停止させ、rsync等で手動でコピーする必要がある
- 一部のSQL(nowなど)には対応していない
- JDBCなどで使う問い合わせには今のところ対応できない(プリペアドステートメント等)

参考URL

- pgpool-II開発サイト
 - <http://pgfoundry.org/projects/pgpool>
- pgpoolコミュニティサイト
 - <http://pgpool.sraoss.jp/>
- 日経IT Proでの紹介記事
 - <http://itpro.nikkeibp.co.jp/article/COLUMN/2006>