

pgpool-II 2.0について

SRA OSS, Inc. 日本支社
石井 達夫

- ◆ pgpool Global Development Groupが開発
- ◆ BSDライセンスで配布される
- ◆ 多彩な機能
 - ◆ コネクションプーリング, レプリケーション, 負荷分散, パラレルクエリ
- ◆ 設定が容易. GUI管理ツールも附属
- ◆ PostgreSQLに手を入れていないので, バージョン追従が容易
- ◆ 使用言語を選ばない

GUI管理ツール: pgpoolAdmin

pgpool.conf設定 - Mozilla Firefox

ファイル(E) 編集(E) 表示(V) 移動(G) ブックマーク(B) ツール(I) ヘルプ(H)

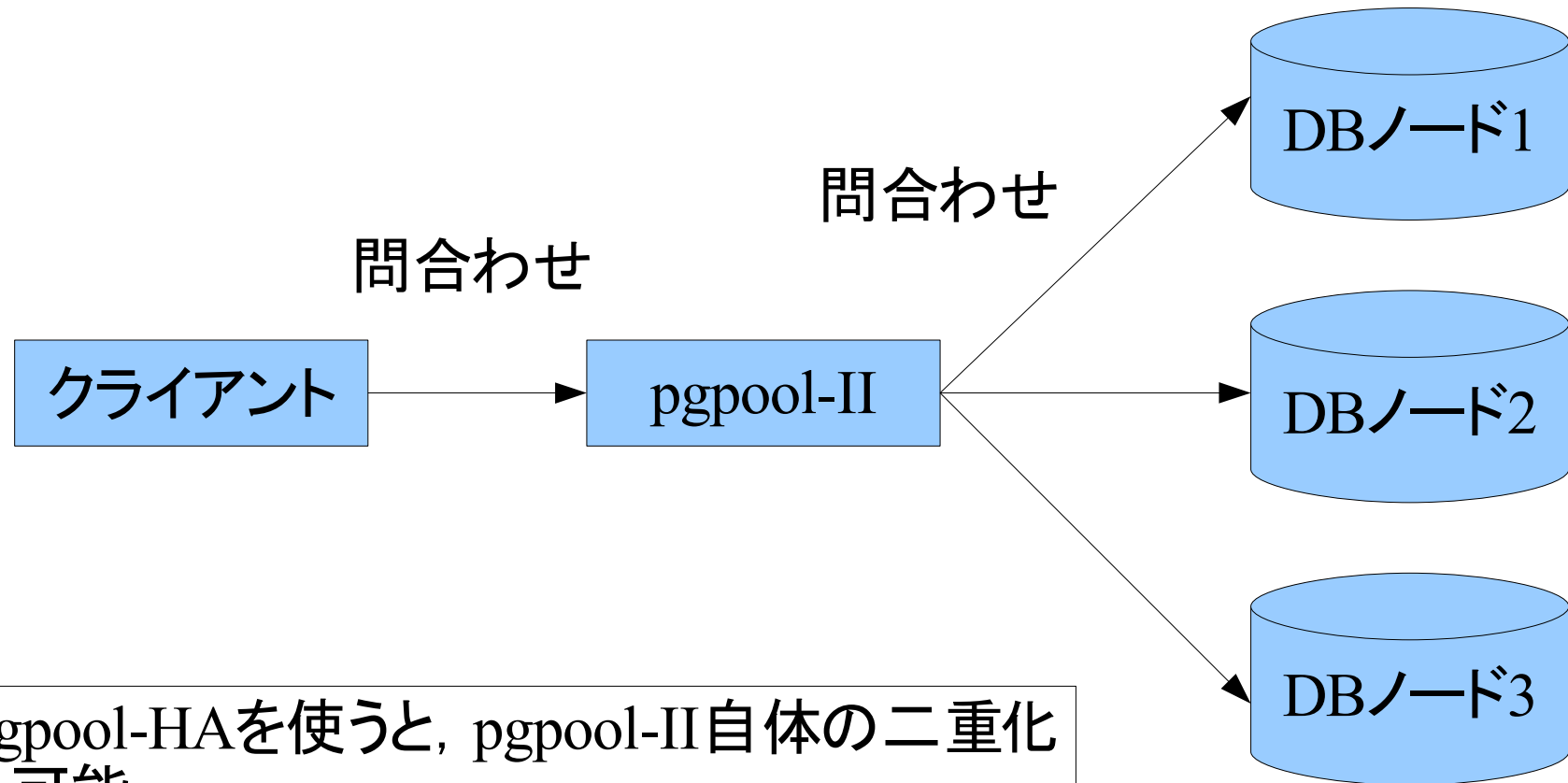
pgpoolAdmin

pgpool.conf設定

ステータス
クエリキャッシュ
マスターDB
pgpool.conf設定
設定
パスワード変更
ヘルプ
ログアウト

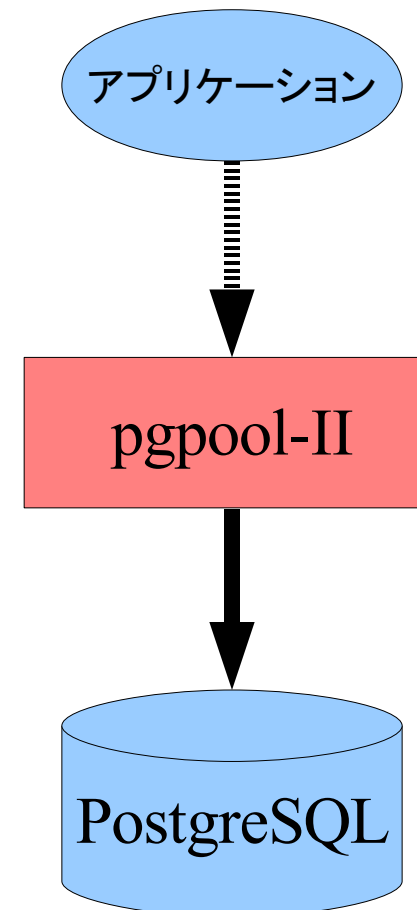
項目	値
listen_addresses	localhost
port	9999
socket_dir	/tmp
backend_socket_dir	/tmp
num_init_children	1
max_pool	4
child_life_time	300
connection_life_time	0
child_max_connections	0
logdir	/tmp
replication_mode	false
replication_strict	true
replication_timeout	5000

完了



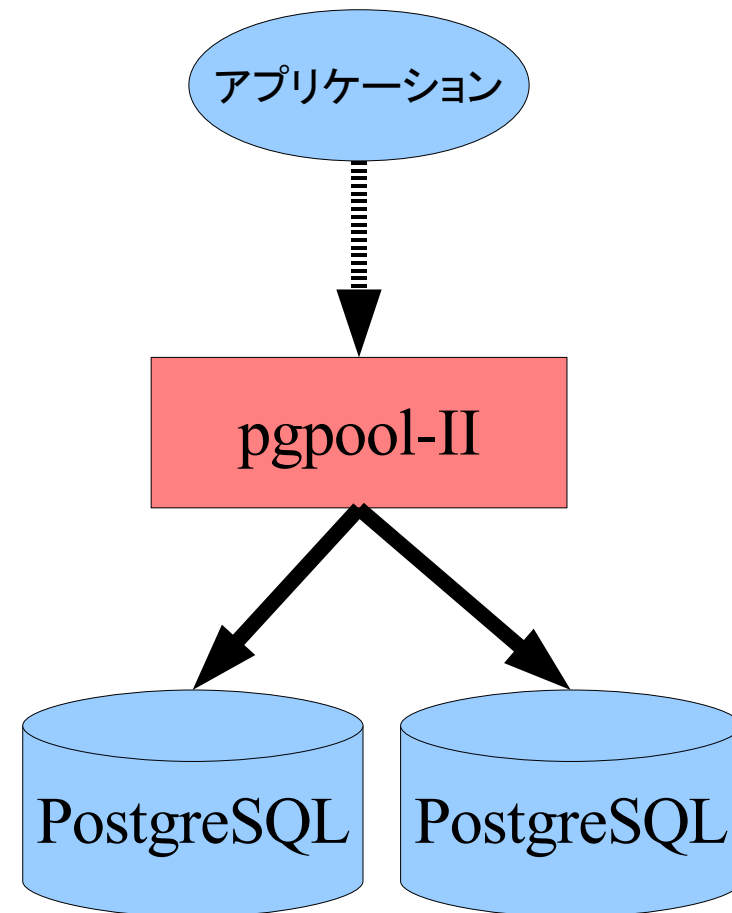
pgpool-HAを使うと, pgpool-II自体の二重化も可能

- ◆ コネクションプーリング
 - ◆ データベースへの接続オーバーヘッドを軽減
 - ◆ PostgreSQLへのコネクションを保持しておき、再利用する
 - ◆ Webシステムのように、頻繁に接続/切断を繰り返すシステムで効果あり



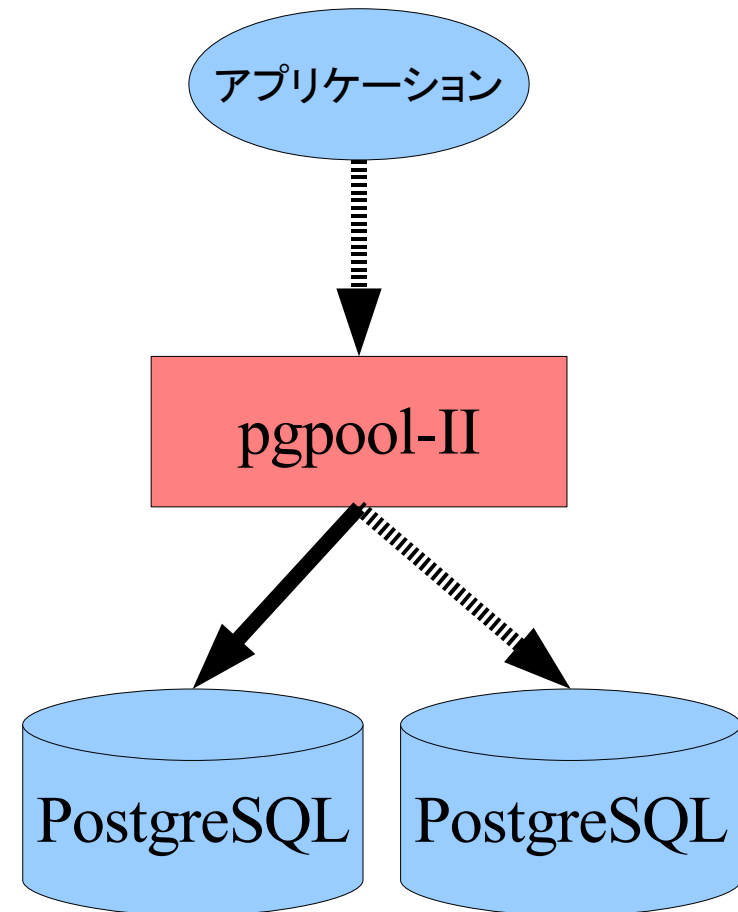
◆レプリケーション

- ◆ SQL文を複製してデータベースのコピーをリアルタイムに作る
- ◆ 片方のDBが障害を起しても自動的に切り離して運用を継続
- ◆ 障害復旧後は運用を停めずにDBを同期, 復帰させることが可能(オンラインリカバリ)
 - ◆ オンラインリカバリの手法をユーザが選択可能なので, PITRを利用して高速リカバリ可能



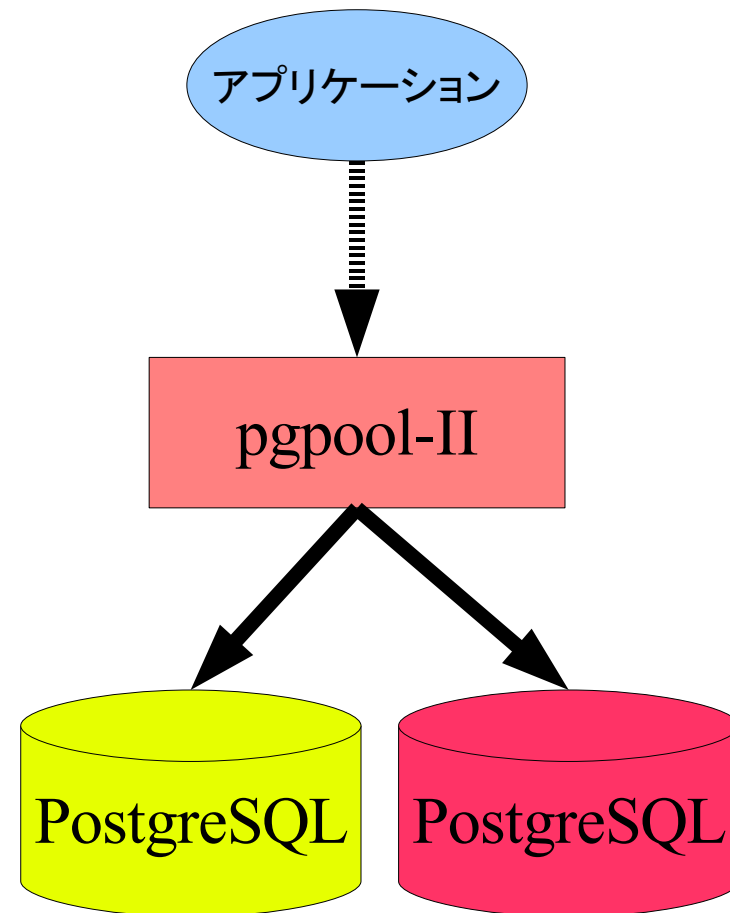
◆ 負荷分散

- ◆ 検索問い合わせをランダムに決めたPostgreSQLに振り向ける
- ◆ 負荷をPostgreSQL間で分かち合うので、検索性能が向上
- ◆ TPC-Cなどの、OLTP(On Line Transaction Processing)系の負荷に向いている



◆ パラレルクエリ

- ◆ データを分割してPostgreSQL間で分担
- ◆ 検索問い合わせを複数のPostgreSQLで一斉に実行, 結果をpgpool-IIでまとめる
- ◆ 問い合わせを並列に処理するので高速
- ◆ TPC-H/DBT-3などOLAP(On Line Analitical Processing)系の負荷に最適



pgpool-II 2.0

◆ 2007/11/16 2.0 リリース

◆ その後マイナーバージョンアップして2.0.1がリリース済

- <http://pgfoundry.org/frs/download.php/1521/pgpool-II-2.0.1.tar.gz>

◆ 様々な改良を追加

◆ レプリケーション

- ◆ 信頼性向上、高速化、オンラインリカバリ

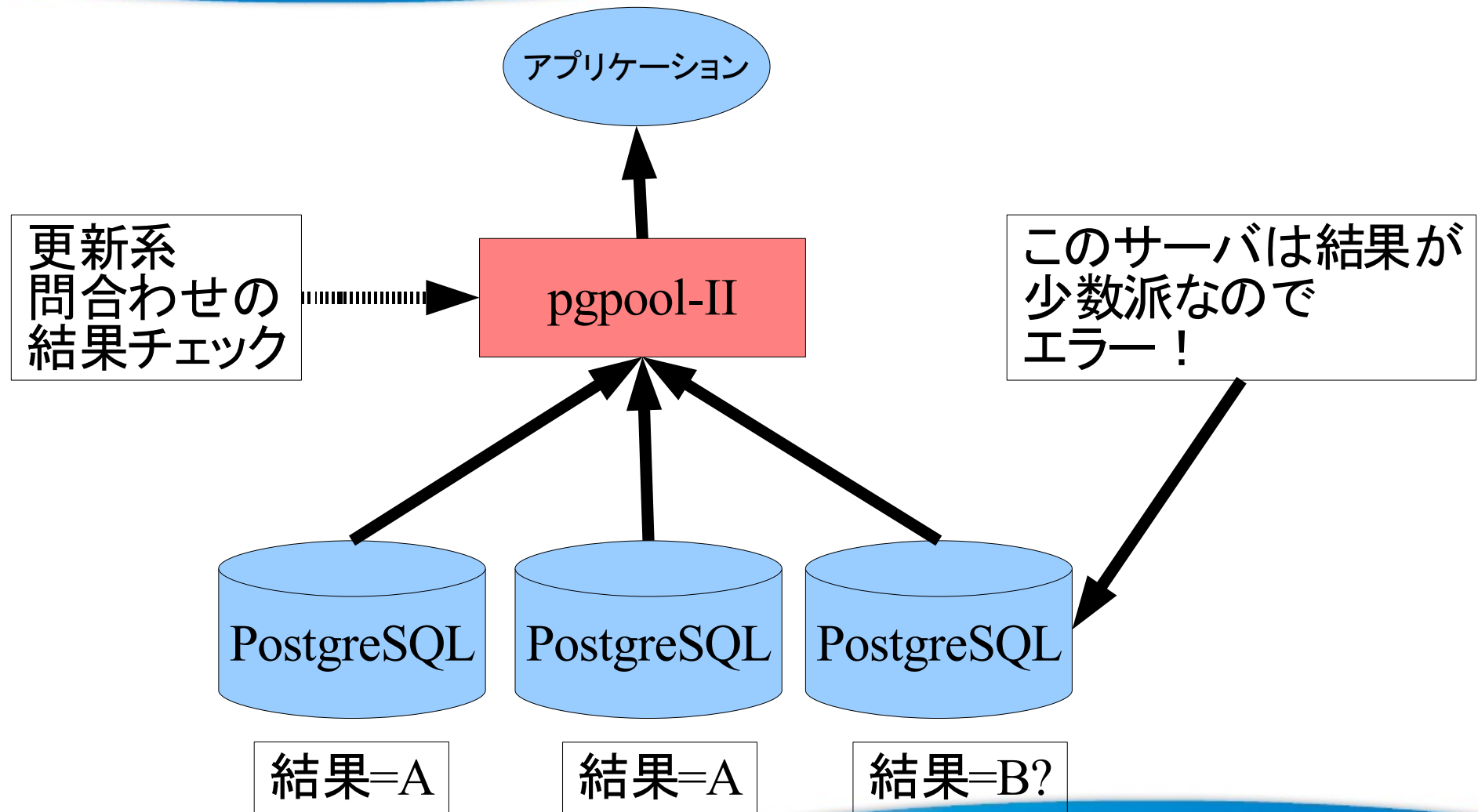
◆ パラレルクエリ

- ◆ 高速化

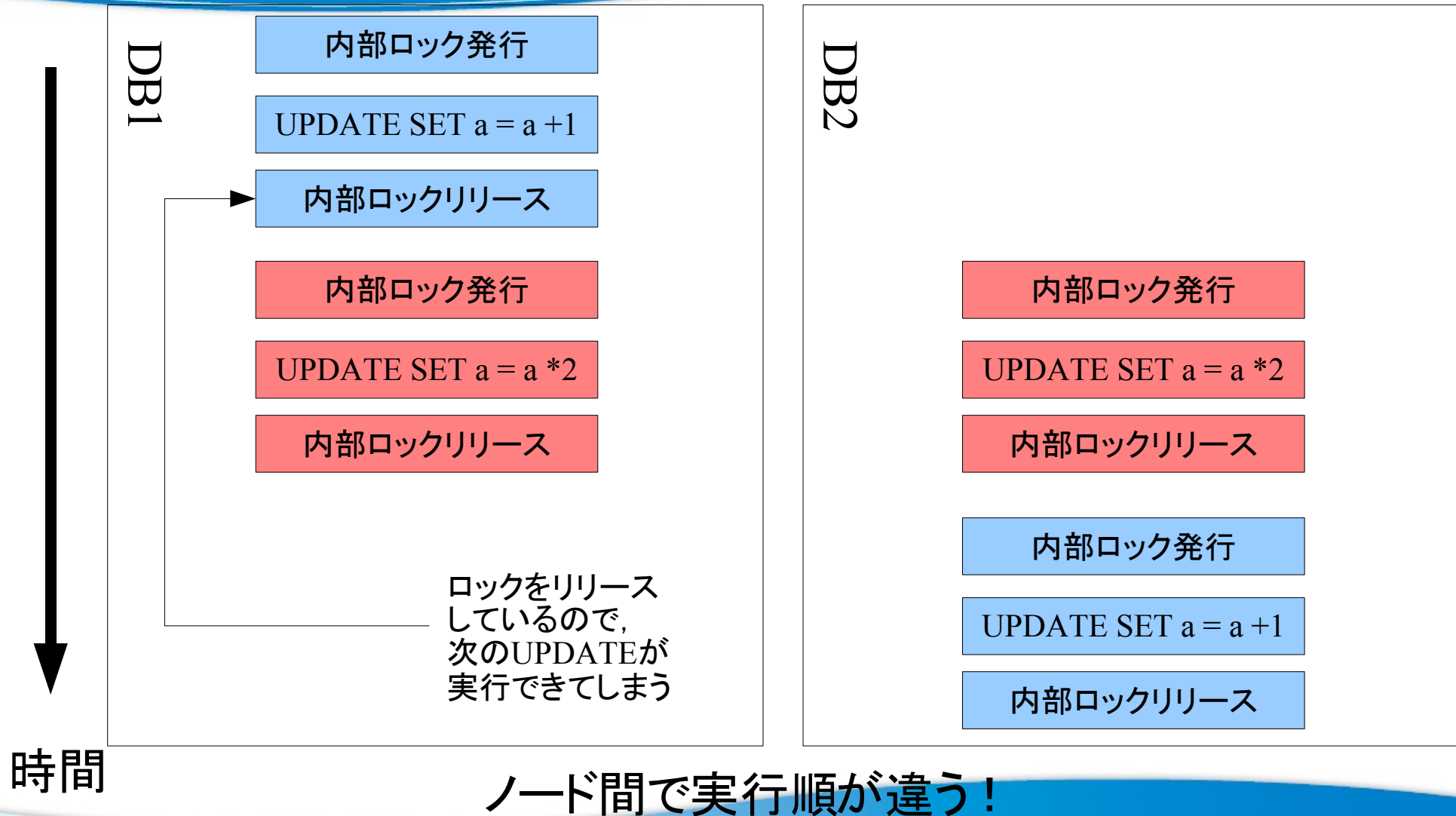
◆ 全般

- ◆ 他のシステムとの連携

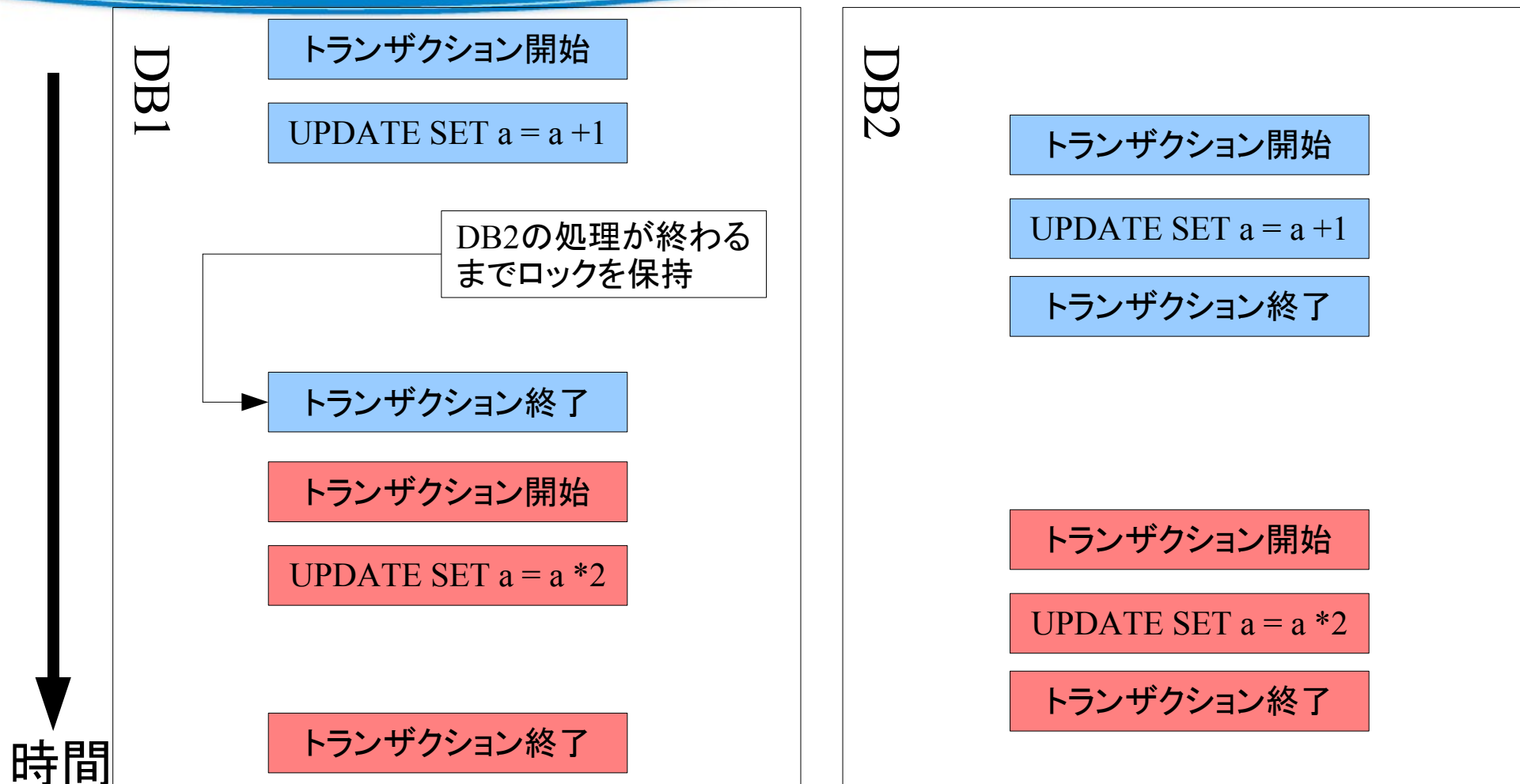
レプリケーションの信頼性向上: 多数決方式の採用



レプリケーションの信頼性向上： 明示的なトランザクション発行



レプリケーションの信頼性向上： 明示的なトランザクション発行



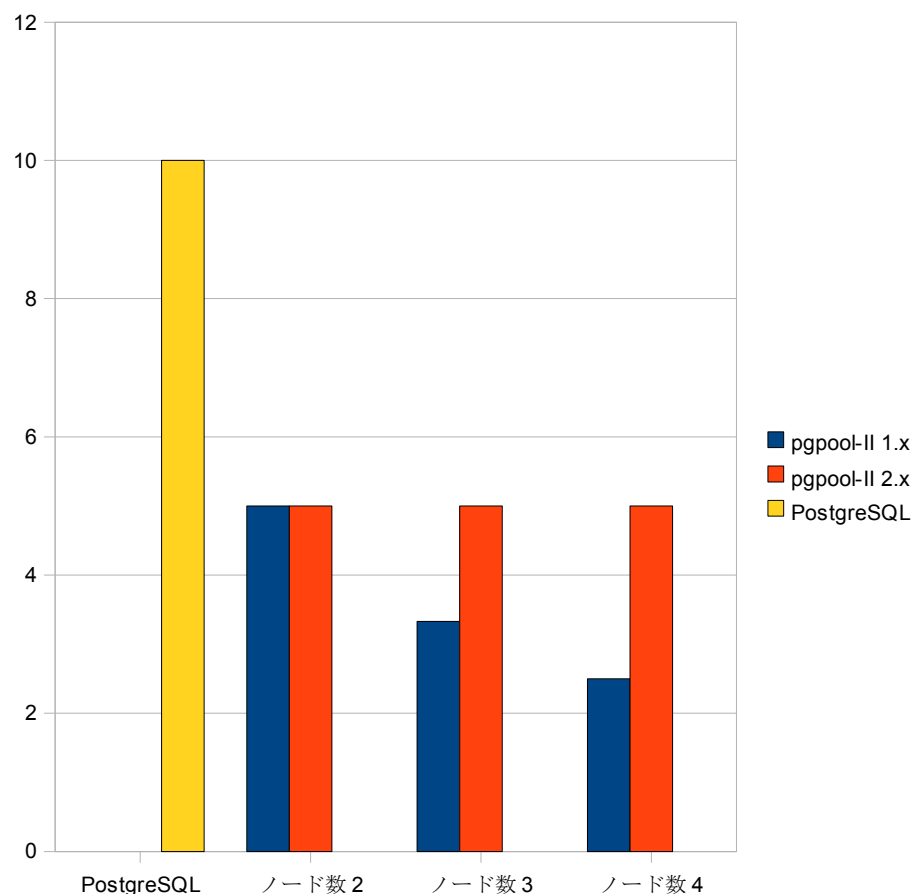
明示的なトランザクションを使うことでノード間の実行順を保証

レプリケーションの信頼性向上: 更新行数の確認

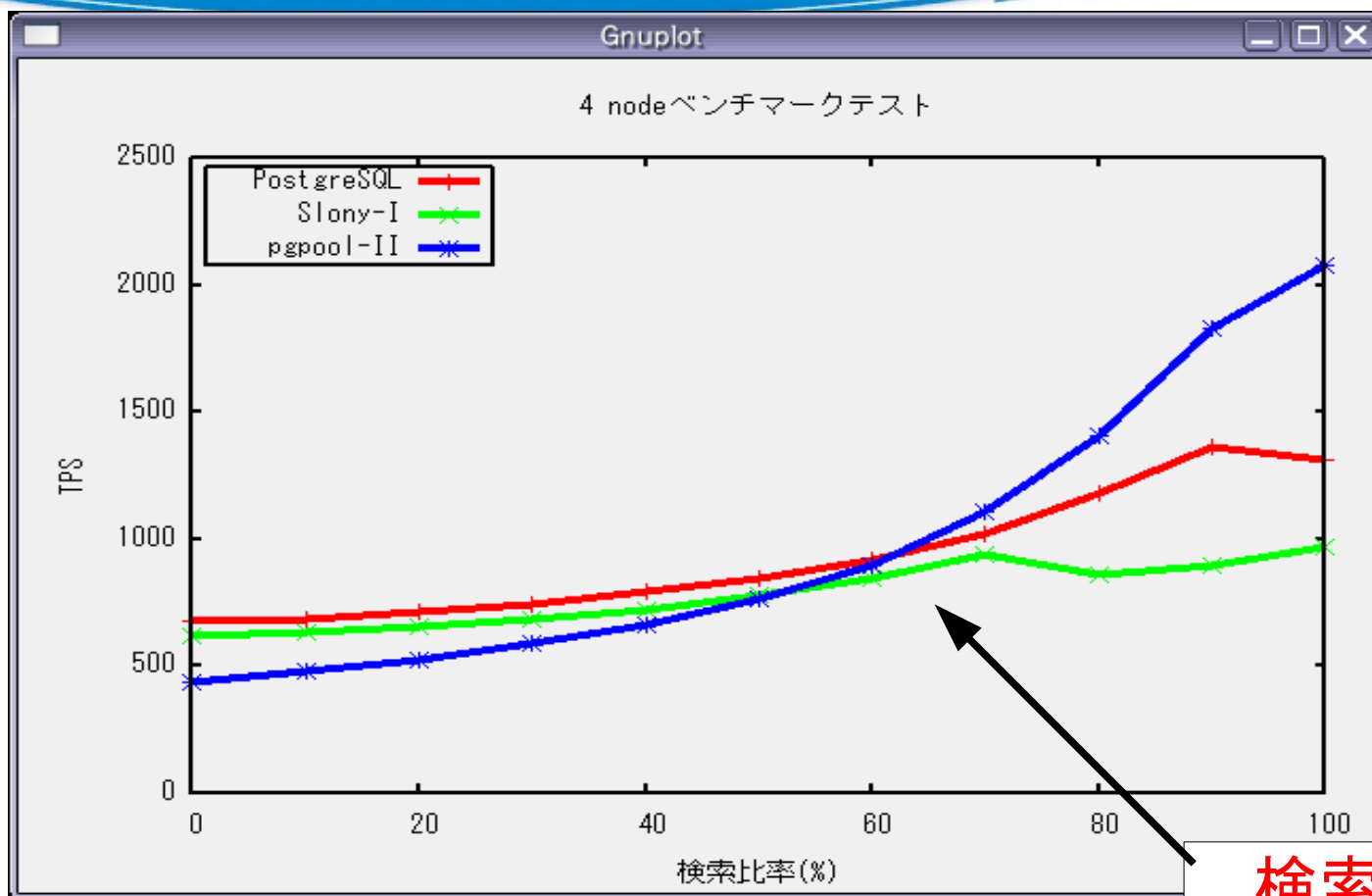
- ◆ pgpool-II 1.xでは更新の成功, 失敗のみを判断
- ◆ pgpool-II 2.0では, 加えて更新件数もチェック
 - ◆ 件数が一致していない場合はエラー扱い
 - ◆ 回復はオンラインリカバリで

- ◆ pgpool-II 1.xでは、更新性能はノード数に比例して低下
 - ◆ ノード数が2なら1/2, 3なら1/3, 4なら1/4...
- ◆ pgpool-II 2.0では、更新性能はノード数に関わらず1/2

更新性能の比較



pgbench によるベンチマーク結果



4ノードでの
レプリケーション
処理テスト

検索比率が60%越えると
pgpool-II が速い。

オンラインリカバリとは？

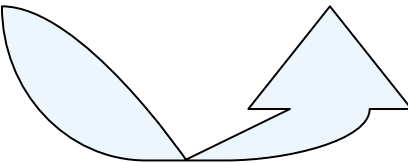
- ◆ pgpoolを停止させずにデータを同期させてノードを復帰させる機能

pgpool-II 1.xは
すべてを停止させて
データを同期

~~pgpool~~

~~PostgreSQL~~

~~PostgreSQL~~



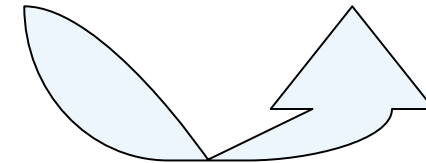
データ同期

pgpool-II 2.0は
停止させずに
データを同期

pgpool

PostgreSQL

PostgreSQL



データ同期

- ◆ 2段階に分けてデータを同期
 - ◆ **ファーストステージ**
 - ◆ 並行して DB の参照・更新が**可能**
 - ◆ **セカンドステージ**
 - ◆ すべての接続が終了するまで待機
 - **接続リクエストをすべてブロック**
- ◆ リカバリ方法
 - ◆ ユーザ自身でどのようにリカバリするかを決めることができる
 - ◆ pgpool-IIのソースコードにサンプルスクリプトを同梱

- ◆ PITRを使ったオンラインリカバリ
 - ◆ 差分バックアップにより、接続をブロックする時間を短時間にすることが可能
 - ◆ ブロックする時間はpostmasterがアーカイブログからリカバリする時間
- ◆ ファーストステージ
 - ◆ ベースバックアップを取得し、ダウンしたホストへコピー
- ◆ セカンドステージ
 - ◆ `SELECT pg_switch_xlog()` を実行し、最新のWALログをアーカイブさせる

- ◆ `pcp.conf`
 - ◆ `pcp_recovery_node` コマンドを使うので設定は必須
- ◆ `pgpool.conf`
 - ◆ `recovery_user`
 - ◆ リカバリ中にPostgreSQL(template1データベース)へ接続するためのユーザ名
 - ◆ `recovery_password`
 - ◆ `recovery_user`のパスワード
 - ◆ `recovery_1st_stage_command`
 - ◆ ファーストステージで実行するコマンド
 - ◆ コマンドは\$PGDATA以下におく必要がある(セキュリティ上の問題)
 - ◆ `recovery_2nd_stage_command`
 - ◆ セカンドステージで実行するコマンド
 - ◆ コマンドは\$PGDATA以下におく必要がある(セキュリティ上の問題)
 - ◆ `backend_data_directoryN`(Nは数字)
 - ◆ データベースクラスタディレクトリ名

◆ 管理ツール上から実施



The screenshot shows the pgpoolAdmin web interface in a Mozilla Firefox browser window. The page title is 'pgpoolステータス - Mozilla Firefox'. The interface includes a sidebar with navigation links and a main content area with tabs for 'サマリー', 'プロセス情報', and 'ノード情報'. The 'ノード情報' tab is active, displaying a table of node status.

IPアドレス	ポート	ステータス	ウェイト	
	5432	ノード稼働中。接続有り	0.333	切断
	5433	ノードダウン	0.333	リカバリ
	5434	ノード稼働中。接続無し	0.333	切断

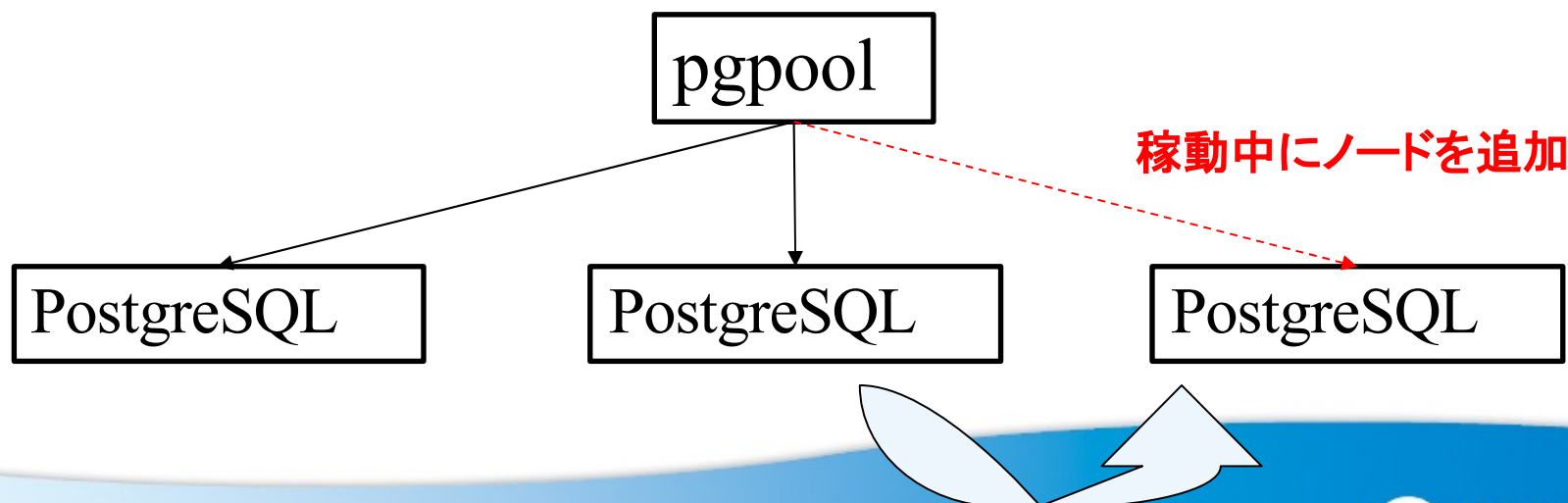
Below the table, there are buttons for 'pgpool停止', 'pgpool再起動', and '設定リロード'. The 'リカバリ' button for the node with IP 5433 is highlighted with a red box.

◆ ノードの動的追加

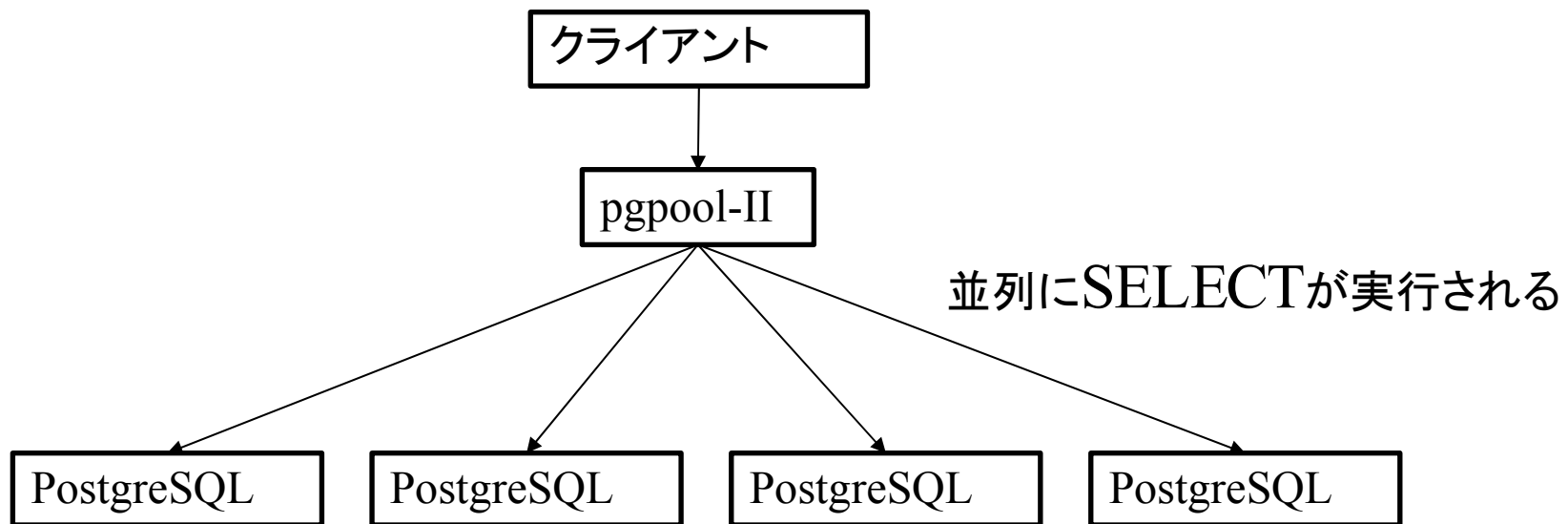
- 2.0 から pgpool.conf のリロードが可能

◆ backend_* を追加し、オンラインカバリすることで負荷状況に応じてノードを追加ができる(**スケールアップ**)

◆ リロードするとノードは「ダウン」状態で追加される



- ◆ 複数のノードでリクエスト(SELECT)を処理
 - ◆ 1つのテーブルを複数サーバに分割(**テーブルパーティショニング**)しておき、SELECTの結果をpgpool-IIでまとめることで、仮想的に1つのテーブルとして扱う



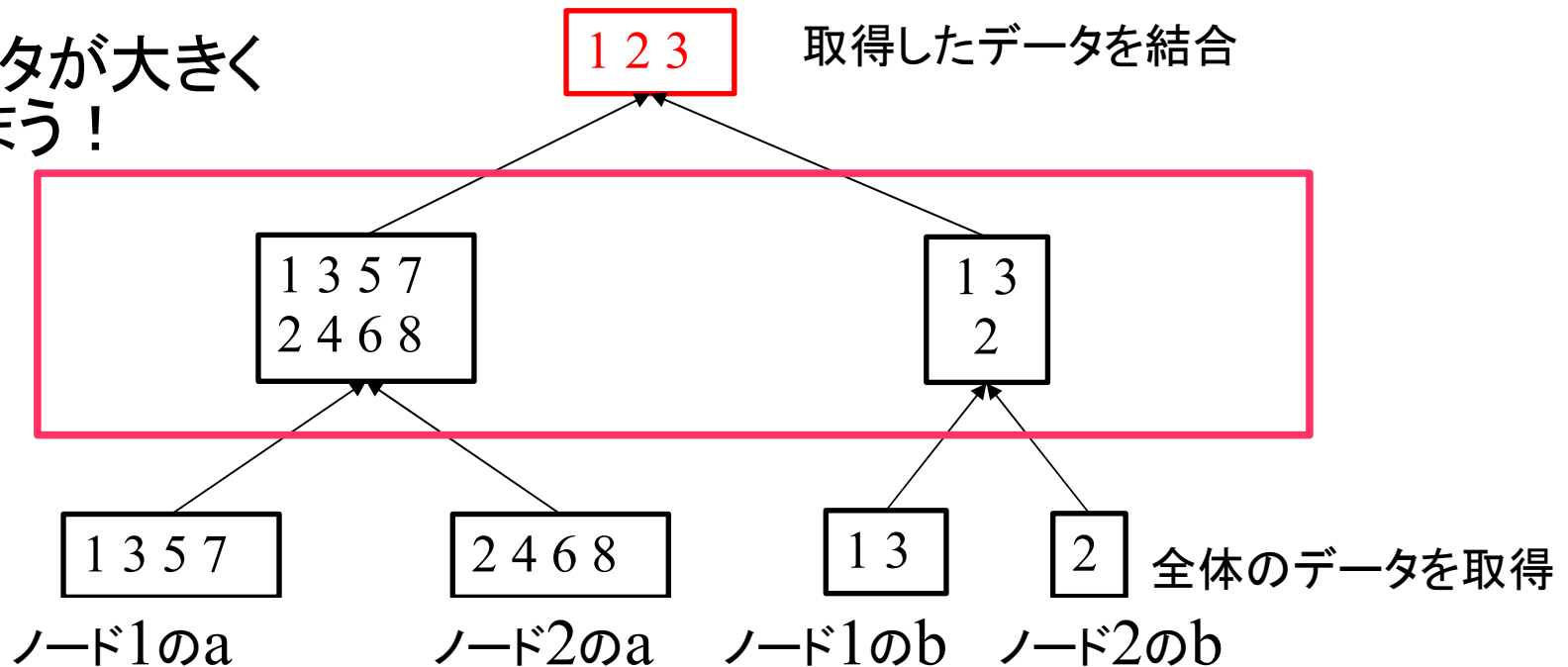
◆ 1.xの挙動

- ◆ すべてのテーブルを分割する必要がある
- ◆ 中間データが大きくなるのでJOINが遅い

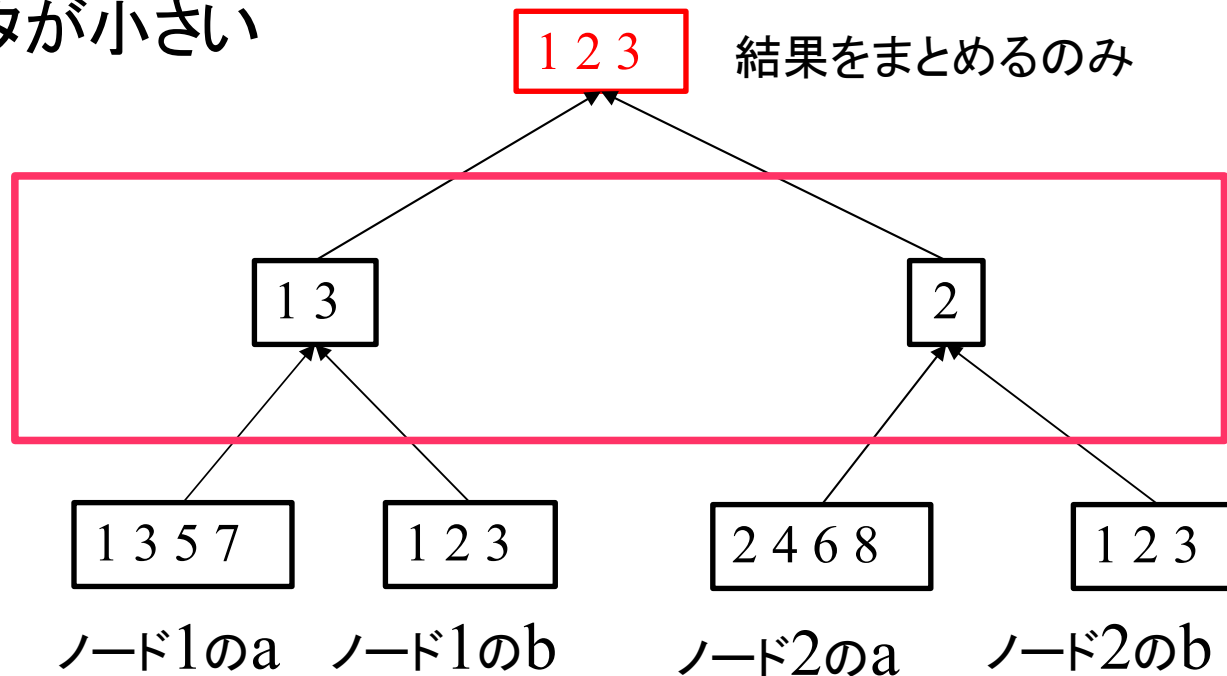
◆ 2.0

- ◆ レプリケーションテーブルとパーティショニングテーブルを混在させることが可能
 - ◆ クエリをより並列に動かすことが可能

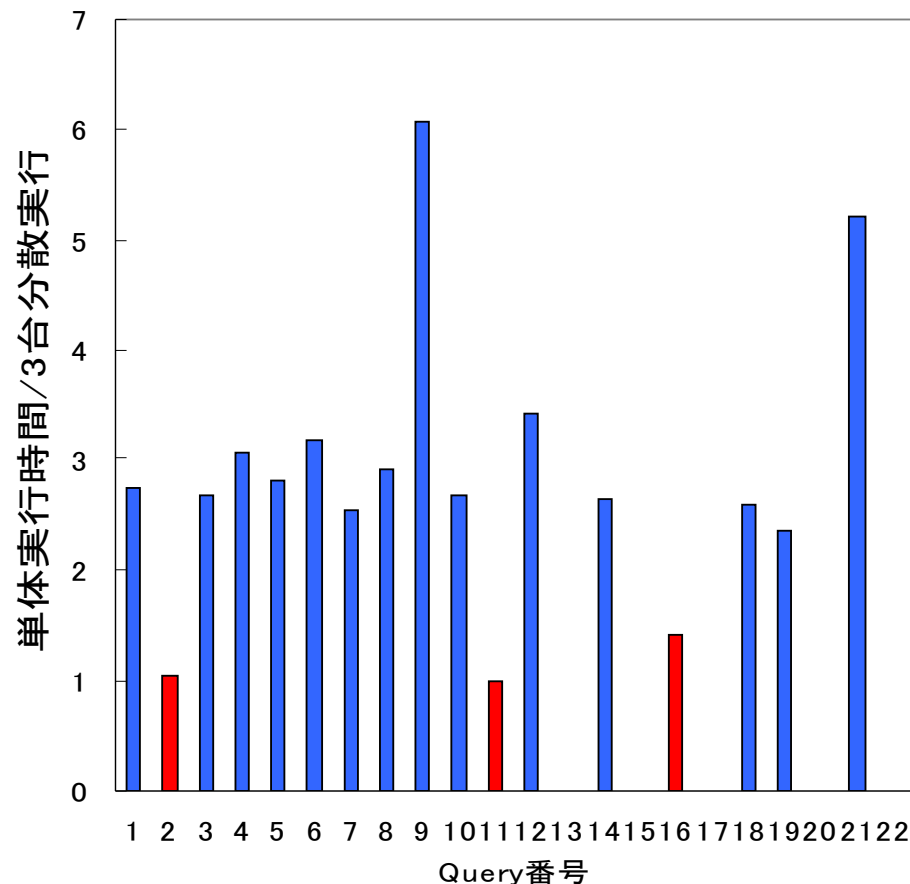
中間データが大きくなってしまう！



中間データが小さい
ので高速



PostgreSQLと3台並列処理の実行時間の比較



◆ line_itemとordersをテーブル分割

■ PostgreSQL 3 ノード(3分割)

- ◆ 青線: パーティションテーブルとレプリケーションテーブルの結合
- ◆ 赤線: レプリケーションテーブル同士の結合
- ◆ パラレルクエリの弱いクエリは未測定 (13,15,17,20,22)
- ◆ scale factor=10
- ◆ インデックス含めてトータルで30GB

**最大で6倍のパフォーマンス改善
(クエリ9番)**

- ◆ failover_command
 - ◆ ノードを切り離れた時に指定したコマンドを実行
 - ◆ Slony-I の連携やアラートメールを送ることが可能
- ◆ failback_command
 - ◆ ノードを戻した時に指定したコマンドを実行
- ◆ pgpool-IIから必要な情報を渡すことも可能(特殊文字を置換)
 - ◆ %d: 対象ノード番号
 - ◆ %h: 対象ノードのホスト名
 - ◆ %p: 対象ノードのポート番号
 - ◆ %D: 対象ノードのデータベースクラスタパス

◆ client_idle_limit

- ◆ クライアントからクエリが届くまでの最大待ち時間を指定可能
- ◆ 秒単位で指定
- ◆ TCPのタイムアウト待ち防止などに有効

- ◆ pgpool-Iは今後pgpool-IIに移行してもらう
- ◆ pgpool-II 1.xは今後メンテを終了予定
- ◆ pgpool-II 2.xへ一本化
- ◆ pgpool-HAはpgpool-IIでも利用可能

- ◆ pgpool-II 2.0の特長
 - ◆ レプリケーションの信頼性向上
 - ◆ 更新処理の高速化(4ノードなら従来の2倍！)
 - ◆ オンラインリカバリ・ノード追加
 - ◆ 部分レプリケーションによるパラレルクエリの高速化
- ◆ SRA OSSでは, pgpool-IIの商用サポートを提供する予定です！

◆ pgpool-II 開発サイト

- <http://pgfoundry.org/projects/pgpool/>

◆ pgpool Wiki

- <http://pgpool.sraoss.jp/>

◆ pgpoolメーリングリスト

- <http://www.sraoss.jp/mailman/listinfo/pgpool-general-jp>

◆ マイコミジャーナルでのpgpool-IIの解説記事

- <http://journal.mycom.co.jp/articles/2007/11/08/pgpool/index.html>

pgpool-II デモシステムの構成

