

PostgreSQL 9.1とpgpool-II 3.1で構築する 高可用性/負荷分散レプリケーション構成

SRA OSS, Inc. 日本支社
pgpool-II開発者
北川 俊広

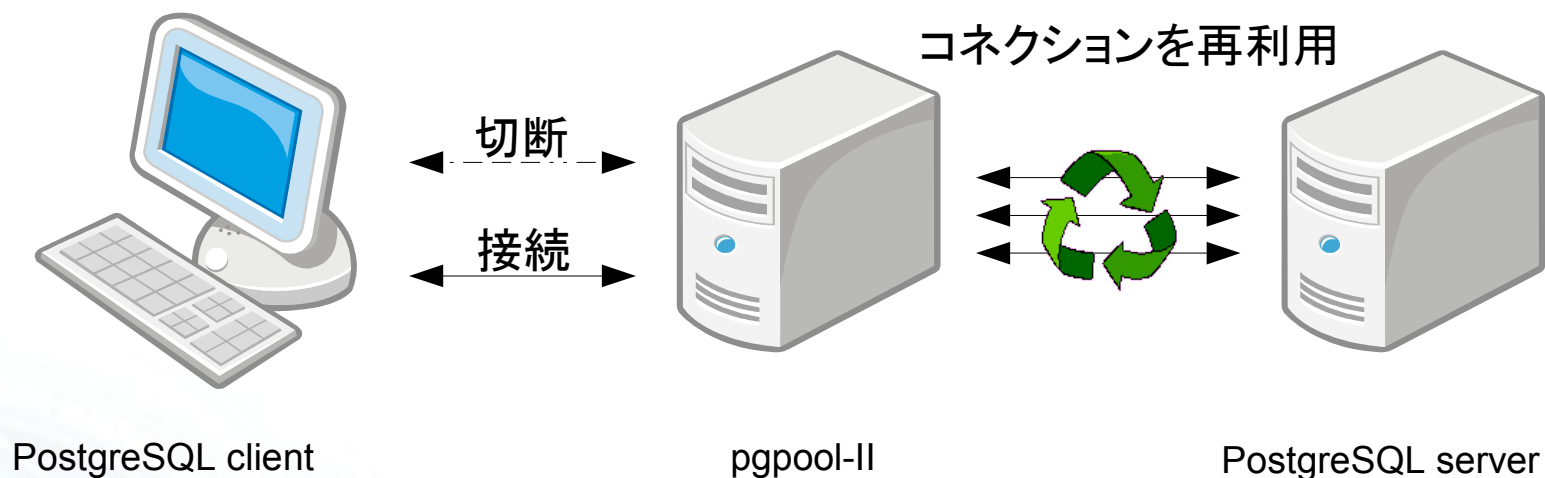
pgpool-IIとは

- アプリケーションとPostgreSQLの間に入って、
便利な機能を提供するソフトウェア
- オープンソースソフトウェア (BSDライセンス)
 - pgpool Global Development Group にて開発
- 多彩な機能
 - 同期レプリケーション、ロードバランス、コネクションプーリング、自動フェイルオーバーなど
 - 他のレプリケーションソフトウェアとの連携
 - Streaming replication, Warm standby, Slony-I
- 設定が容易、GUI管理ツールも用意

Streaming Replication/Hot Standby とpgpool-IIを組み合わせる

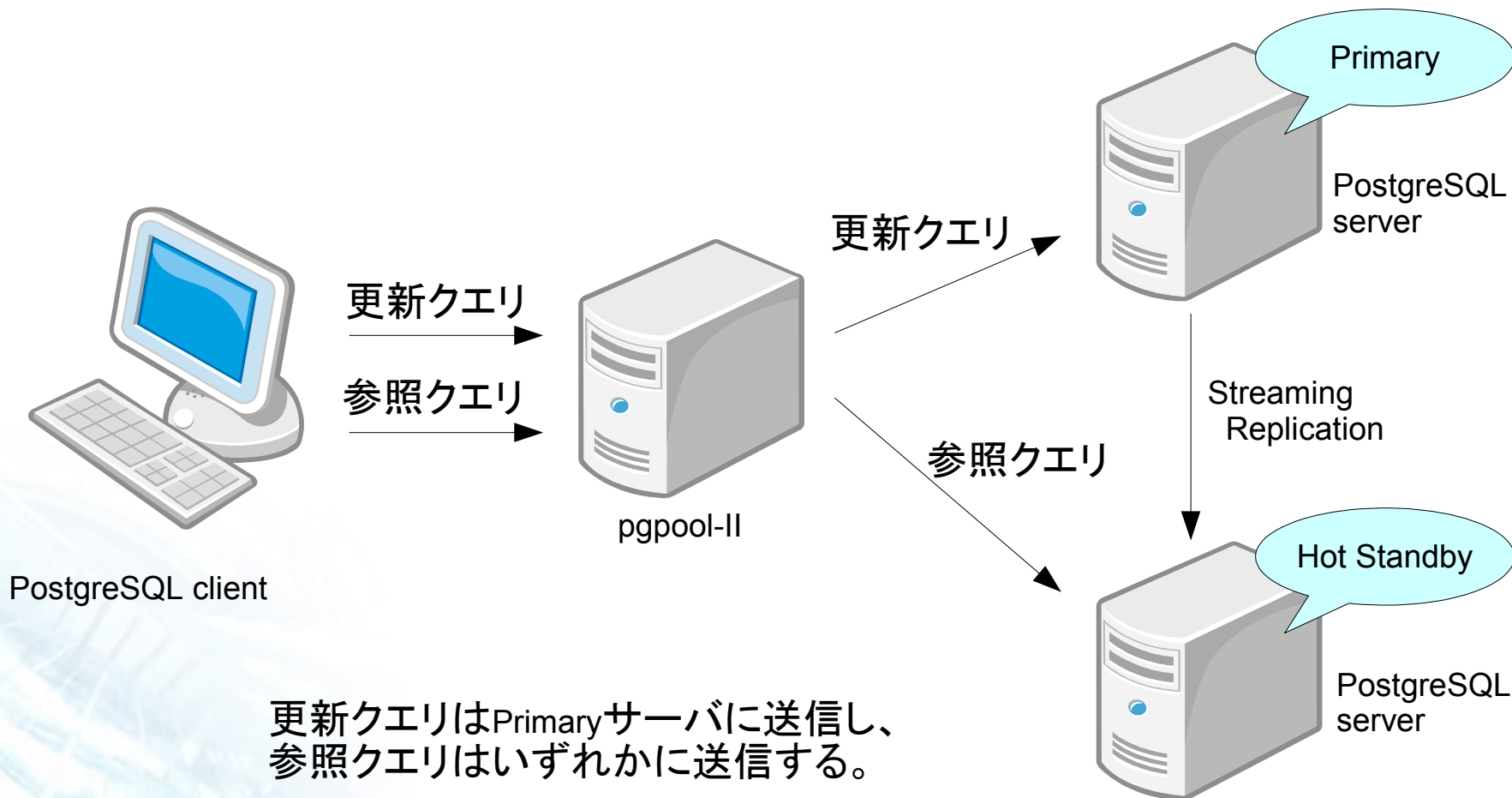
アプリケーション側を作りこまずに、
高可用性と性能向上を簡単に実現できる

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。

クエリ振り分けの自動化、ロードバランス

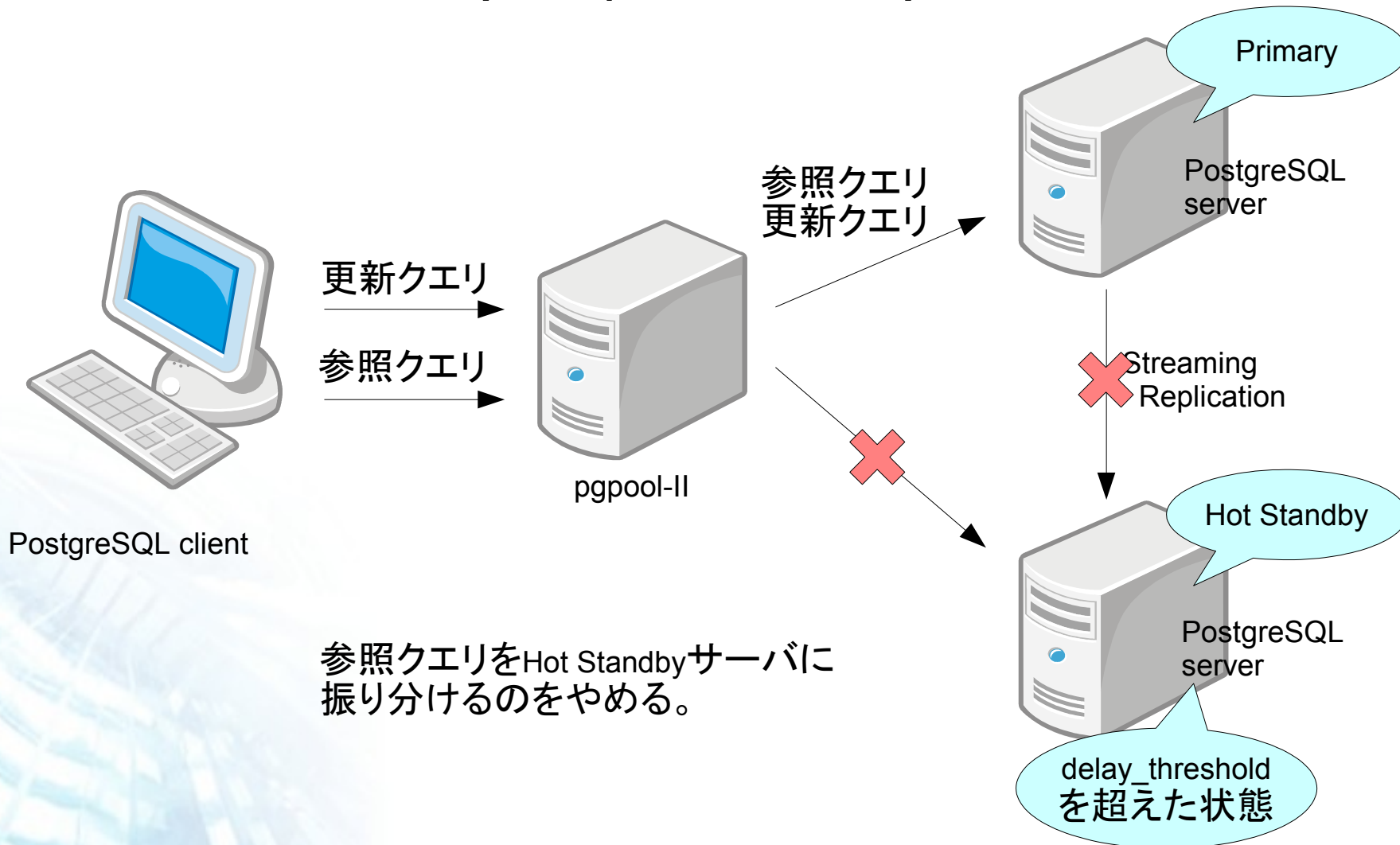


Hot Standby側の制約を考慮したクエリ振り分け ～ Hot Standby側で実行できないクエリ ～

- Data Manipulation Language (DML) - INSERT, UPDATE, DELETE, COPY FROM, TRUNCATE. Note that there are no allowed actions that result in a trigger being executed during recovery.
- Data Definition Language (DDL) - CREATE, DROP, ALTER, COMMENT. This also applies to temporary tables also because currently their definition causes writes to catalog tables.
- SELECT ... FOR SHARE | UPDATE which cause row locks to be written
- Rules on SELECT statements that generate DML commands.
- LOCK that explicitly requests a mode higher than ROW EXCLUSIVE MODE.
- LOCK in short default form, since it requests ACCESS EXCLUSIVE MODE.
- Transaction management commands that explicitly set non-read-only state:
 - BEGIN READ WRITE, START TRANSACTION READ WRITE
 - SET TRANSACTION READ WRITE, SET SESSION CHARACTERISTICS AS TRANSACTION READ WRITE
 - SET transaction_read_only = off
- Two-phase commit commands - PREPARE TRANSACTION, COMMIT PREPARED, ROLLBACK PREPARED because even read-only transactions need to write
- WAL in the prepare phase (the first phase of two phase commit).
- Sequence updates - nextval(), setval()
- LISTEN, UNLISTEN, NOTIFY



賢いロードバランス (delay_threshold)



PostgreSQL client

pgpool-II

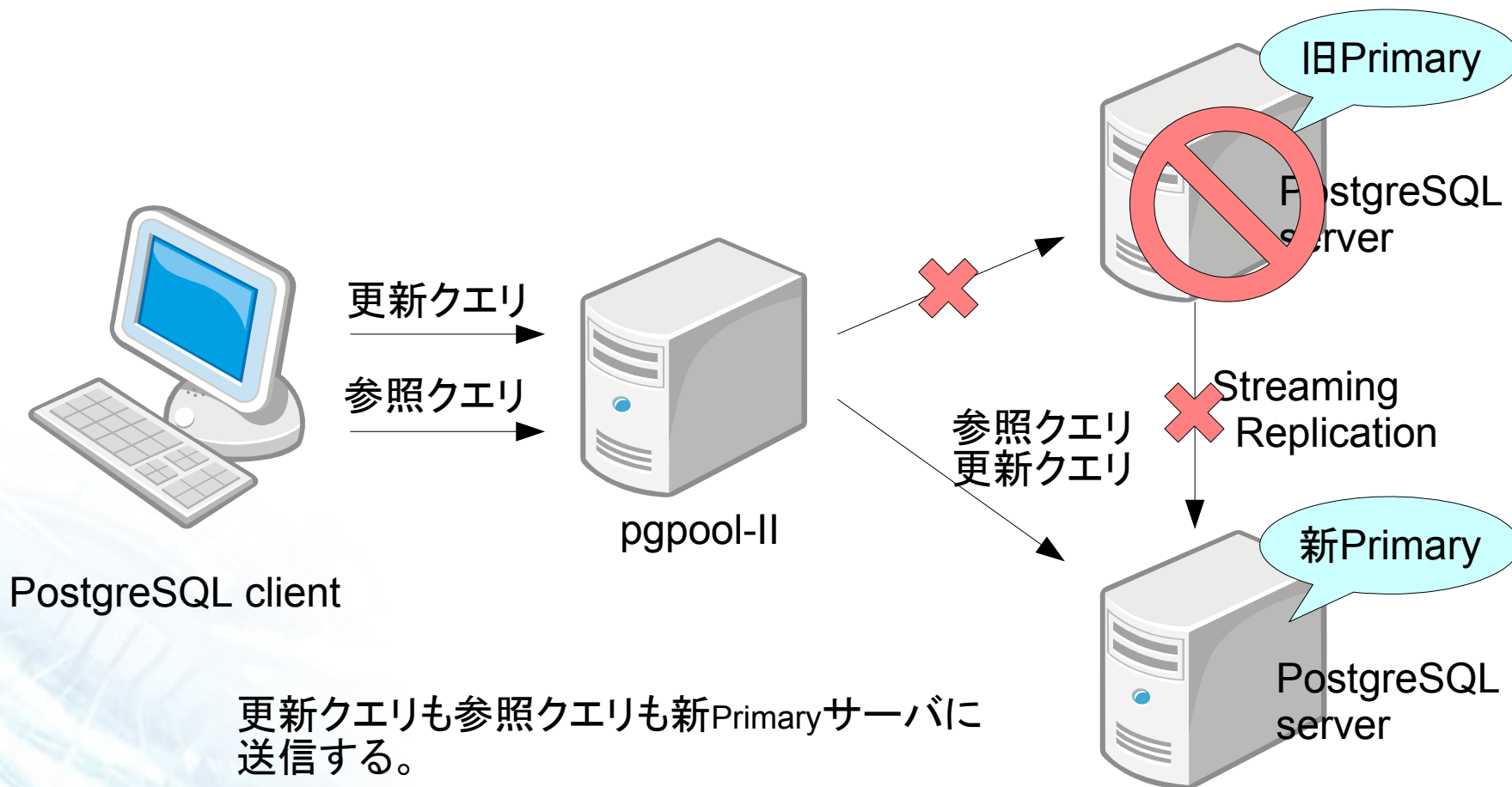
PostgreSQL server

PostgreSQL server

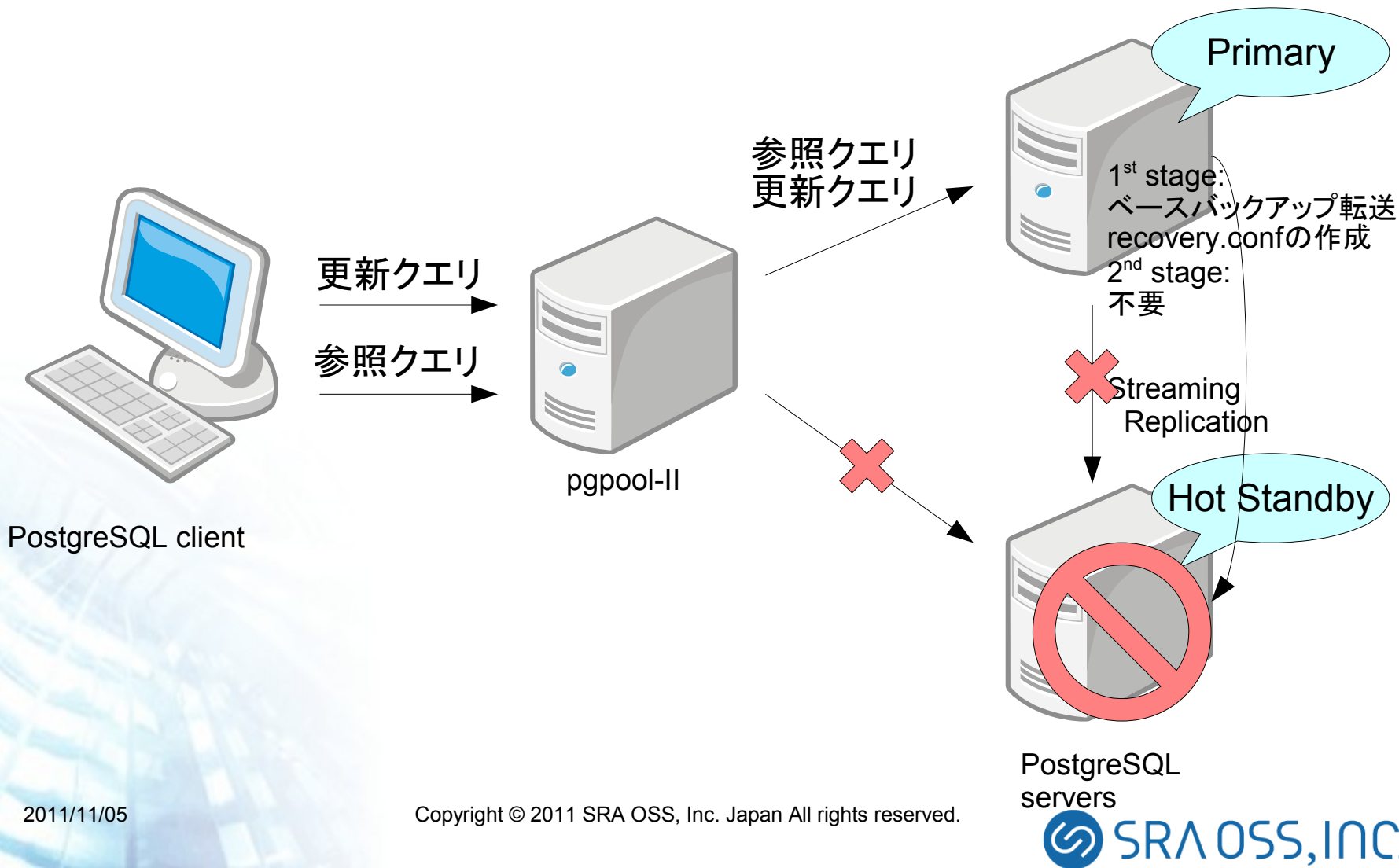
ロードバランス詳細

- 遅延を監視し、閾値を超えたノードには参照クエリを振り分けない
- SERIALIZABLEのトランザクションは参照クエリを振り分けない
- 明示的なトランザクション内の参照クエリもロードバランス可能に
- 一時テーブル、システムカタログを検索する参照クエリは、Primaryに送信
- トランザクション内で更新クエリが発行されたら、以後トランザクション内の参照クエリはPrimaryに送信

自動フェイルオーバー



オンラインリカバリを利用した Hot Standbyノードの復旧



SR/HS関連のパラメータ

- `master_slave_mode = true`
マスタースレーブモードの有効・無効を指定
- `master_slave_sub_mode = 'stream'`
組み合わせるレプリケーションソフトウェアを指定
- `delay_threshold`
スタンバイサーバにクエリを振り分けないようにする
遅延バイト数を指定

レプリケーションの遅延を確認する

- log_standby_delay
 - 'none': ログ出力しない
 - 'if_over_threshold': 閾値を超えたらログ出力する
 - 'always': 常にログ出力する

```
2010-06-28 15:51:32 LOG: pid 13223: Replication of node:1 is behind 1228800 bytes
from the primary server (node:0)
2010-06-28 15:51:42 LOG: pid 13223: Replication of node:1 is behind 3325952 bytes
from the primary server (node:0)
2010-06-28 15:51:52 LOG: pid 13223: Replication of node:1 is behind 974848 bytes
from the primary server (node:0)
2010-06-28 15:52:02 LOG: pid 13223: Replication of node:1 is behind 2990080 bytes
from the primary server (node:0)
2010-06-28 15:52:12 LOG: pid 13223: Replication of node:1 is behind 901120 bytes
from the primary server (node:0)
2010-06-28 15:52:22 LOG: pid 13223: Replication of node:1 is behind 2433024 bytes
from the primary server (node:0)
```

pgpool-II 3.0から3.1への変更点

互換性のない変更

2011/11/05

Copyright © 2011 SRA OSS, Inc. Japan All rights reserved.

insert_lockのロック方法

- insert_lockの変遷
 - ~2.3: INSERT対象のテーブルをロック
 - 3.0: シーケンステーブルの行をロック
 - 3.1: insert_lockテーブルの行をロック
 - 各DBにpgpool_catalog.insert_lockを作成
 - sql/insert_lock.sql
 - INSERT対象のテーブルのoidがinsert_lockテーブルに自動的に追加され、その行が排他制御に使われる。
 - insert_lockテーブルがない場合は2.3までと同等の動作
- --enable-table-lock, --enable-sequence-lock

backend_socket_dirの廃止

- backend_hostnameNが'/'で始まる場合は、UNIXドメインへのパスとみなす
 - libpqインタフェースに準じた仕様
 - 空文字を指定すると/tmpを利用
- backend_socket_dirも使用可能

pgpool_walrecruning()の廃止

- プライマリノード判断
 - 3.0: `!(pg_is_in_recovery() AND pgpool_walrecruning())`
 - 3.1: `!pg_is_in_recovery()`
 - プライマリノードがなかなか現れない場合は、`recovery_timeout`まで待ち、その間接続はブロックされる
 - `recovery_timeout`を超えた場合は、ノード番号の一番小さいバックエンドをプライマリとして扱う

pool_nodes

- pool_nodesにノードIDを追加

```
=# SHOW pool_nodes;
```

id	hostname	port	status	lb_weight
0	127.0.0.1	5432	2	0.5
1	192.168.1.7	5432	3	0.5

(2 lignes)

nextval(), setval()

- nextval(), setval()の扱い
 - 3.0: 常に書き込み関数として処理
 - 3.1: black/white_function_listの設定に従う

新機能

- `syslog`に対応
 - `log_destination = 'syslog'`
- `application_name`に対応
 - コネクションが再利用されても追隨
- `relcache_expire`パラメータを追加
 - システムカタログキャッシュの有効期限を設定可能に
- `follow_master_command`パラメータを追加
 - `failover_command`の後に実行するコマンドを設定可能に
- `pcp_promote_node`
 - プライマリノードを手動で変更可能に

- pcp_pool_statusコマンドを追加
 - SHOW pool_status;のコマンド版
- sr_check_period/user/passwordパラメータを追加
 - health_chek_period/userから遅延チェックを分離
- health_check_passwordパラメータを追加
 - 現時点では未実装
- pgpool_adm関数を追加
 - pcpコマンド(一部)のユーザ定義関数版
- UNLOGGEDテーブルに対応
 - UNLOGGEDテーブルは常にプライマリで処理
- 中文マニュアル、チュートリアルを追加

- backend_flagNパラメータを追加
 - フェイルオーバーしないノードの指定が可能に
 - 'ALLOW_TO_FAILOVER'または'DISALLOW_TO_FAILOVER'を指定
 - 'DISALLOW_TO_FAILOVER'の場合は、pcp_detach_nodeでもデタッチ不可

改良

- ストリーミングレプリケーション構成に関する変更
 - オンラインリカバリ時に子プロセスを再起動しないよう変更
 - 既存セッションはオンラインリカバリ中も処理を継続可能に
 - `pcp_attache_node`を既存のセッションを切断しないよう変更
 - 遅延チェックに`pg_last_xlog_replay_location()`を使用するよう変更
 - 3.0: `pg_last_xlog_receive_location()`
- `black_function_list`と`white_function_list`パラメータに正規表現が使用可能に

参考URL

- ストリーミングレプリケーションとpgpool-IIの組み合わせ、簡単設定法
 - ローカルホスト編
 - http://pgpool.projects.postgresql.org/contrib_docs/simple_sr_setting/index-ja.html
 - 複数サーバ編
 - http://pgpool.projects.postgresql.org/contrib_docs/simple_sr_setting2/index-ja.html