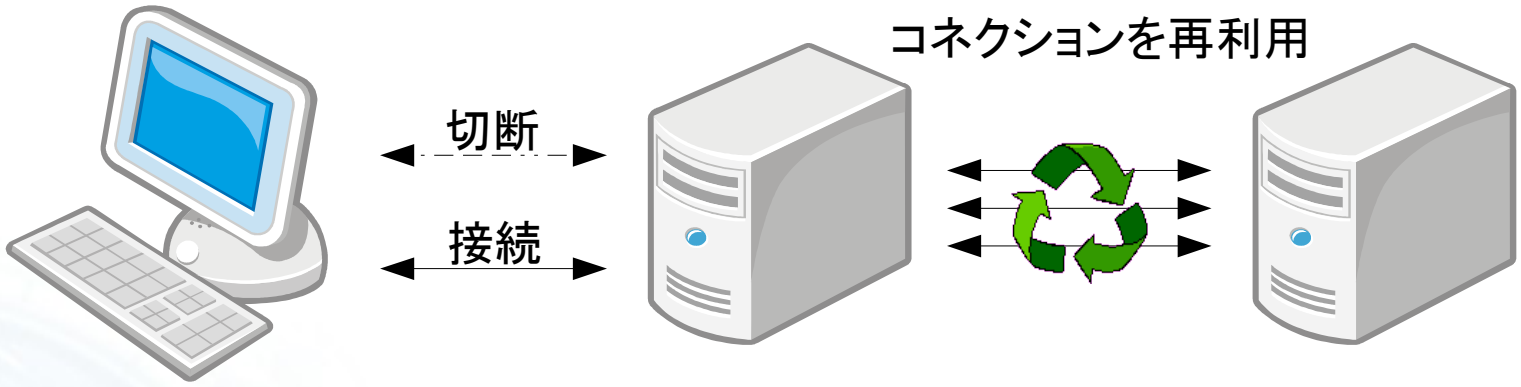
PostgreSQL 9.0とpgpool-II 3.0で構築する高可用性/負荷分散レプリケーション構成

SRA OSS, Inc. 日本支社
技術開発部／pgpool-II開発者
北川 俊広

pgpool-IIとは

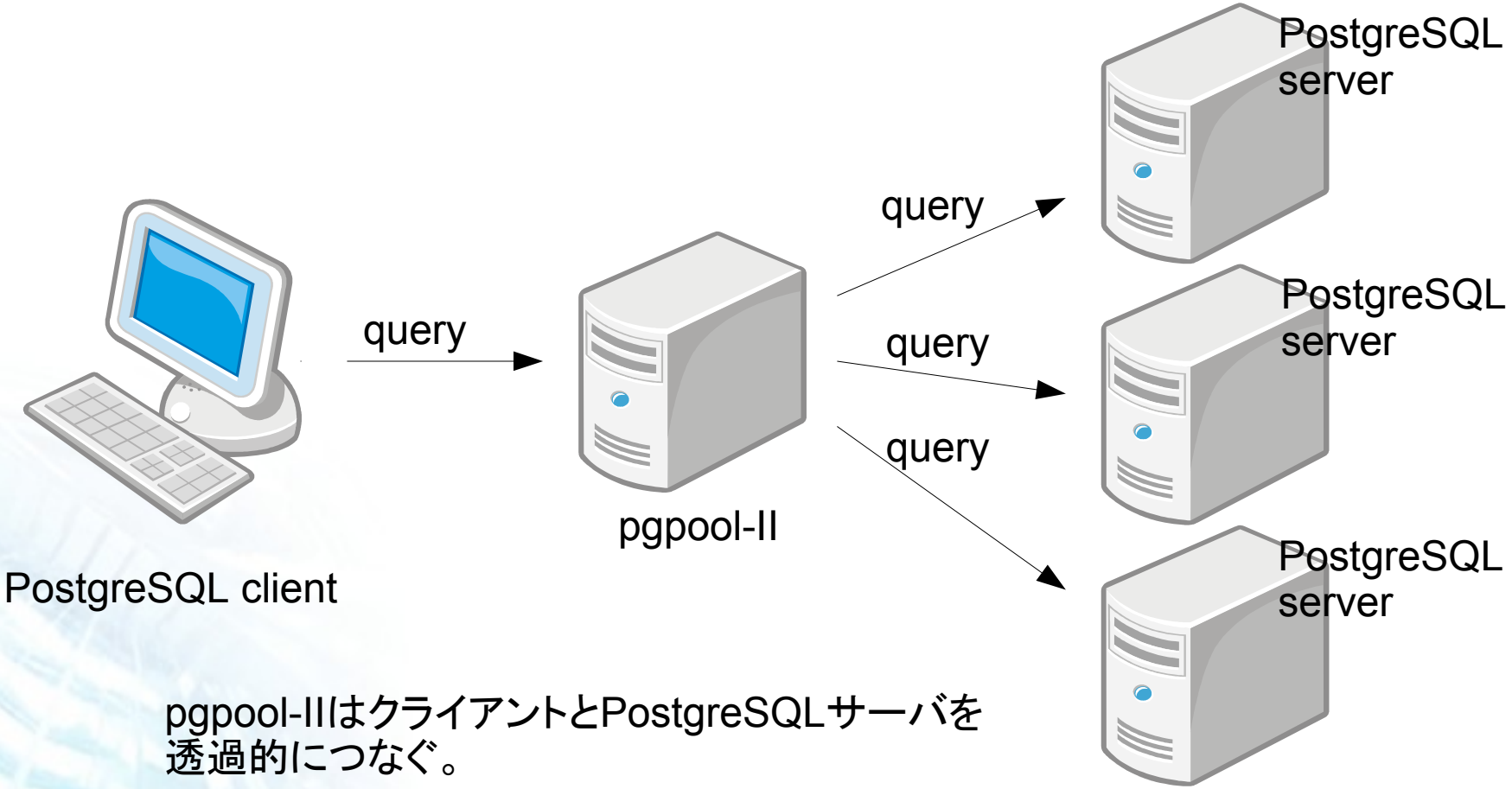
- アプリケーションとPostgreSQLの間に入って、便利な機能を提供するソフトウェア
- オープンソースソフトウェア（BSDライセンス）
  - pgpool Global Development Group にて開発
- 多彩な機能
  - 同期レプリケーション、ロードバランス、コネクションプーリング、自動フェイルオーバー、パラレルクエリ
  - 他のレプリケーションツールとの連携
    - Slony-I, Warm standby, Streaming replication
- 設定が容易、GUI管理ツールも用意

コネクションプーリング



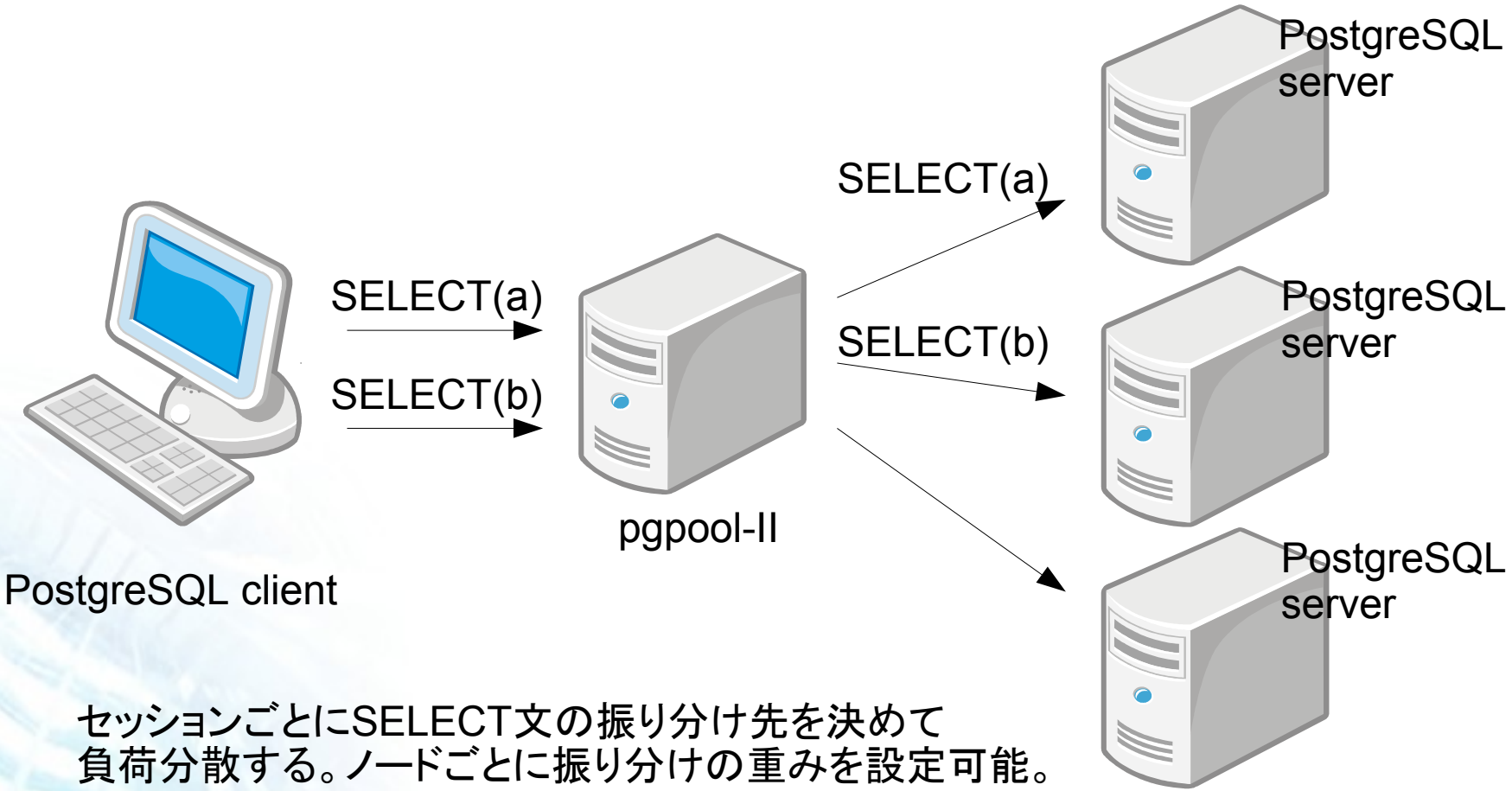
すでに確立しているコネクションを再利用することにより、PostgreSQLが接続時に行っている、認証、子プロセスの生成、データアクセスのための前処理などを省いて効率化できる。

同期レプリケーション



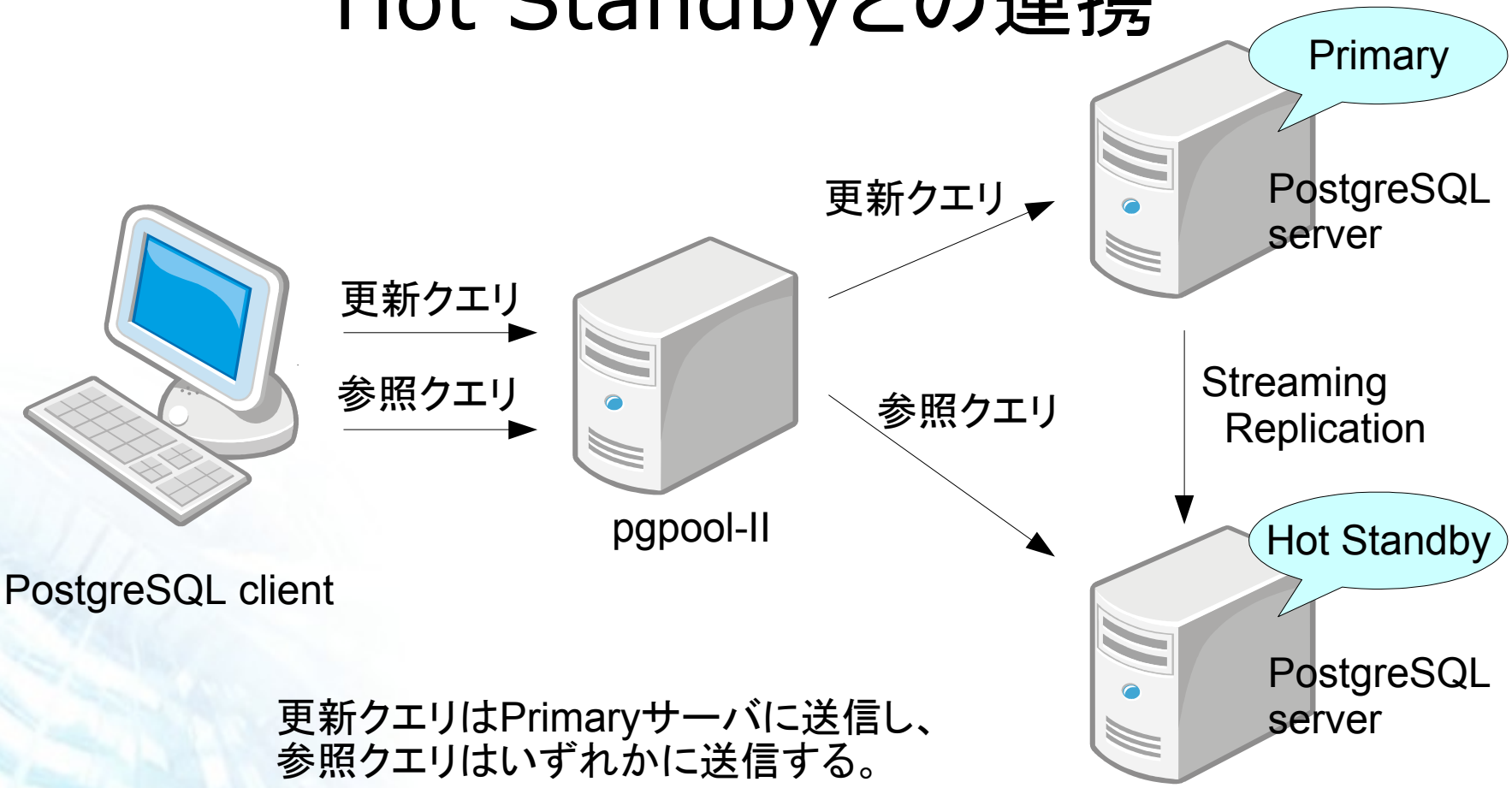
pgpool-IIはクライアントとPostgreSQLサーバを
透過的につなぐ。

ロードバランス



セッションごとにSELECT文の振り分け先を決めて
負荷分散する。ノードごとに振り分けの重みを設定可能。

Streaming Replication/ Hot Standbyとの連携



更新クエリはPrimaryサーバに送信し、
参照クエリはいずれかに送信する。

pgpool-IIの進化の歩み

2003年～2007年

- pgpool 0.1(2003/06)～3.4.1(2007/09)
  - 弊社支社長の石井が開発
    - 後にpgpool Global Development Groupへ移行
  - BSDライセンス
  - 機能
    - コネクションプーリング（開発初期はこれだけ）
    - 同期レプリケーション（2ノード）
    - ロードバランス
    - 自動フェイルオーバー
    - 他のレプリケーションツールとの連携

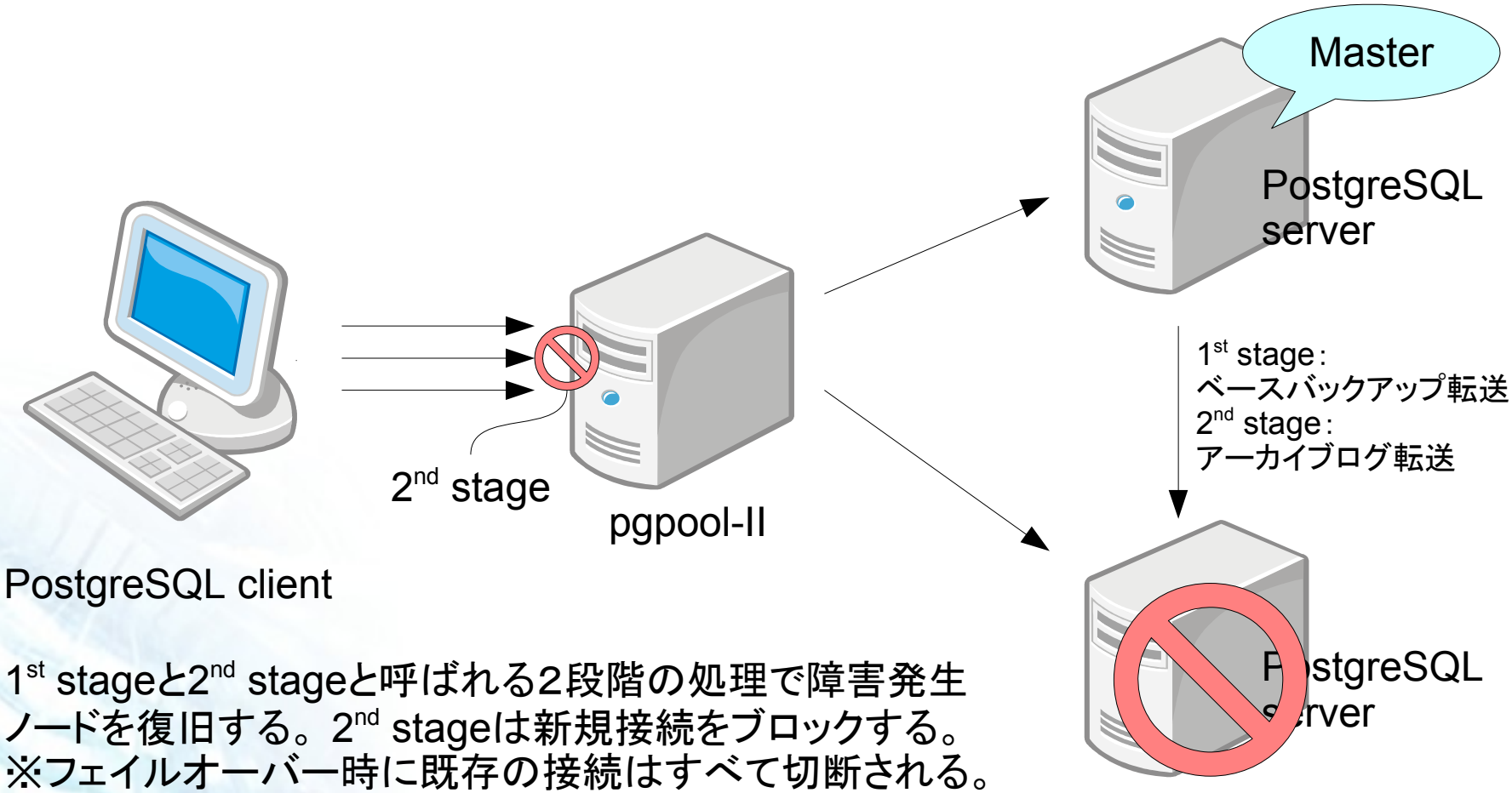
2006年～2009年

- pgpool-II 1.0（2006/09）
  パラレルクエリ、クエリキャッシュ、接続認証、128ノードまで対応
- pgpool-II 2.0(2007/11)
  オンラインリカバリ、フェイルオーバー・フェイルバック時の任意コマンド実行
- pgpool-II 2.2(2009/02)
  insert_lockを改良

2009年～2010年

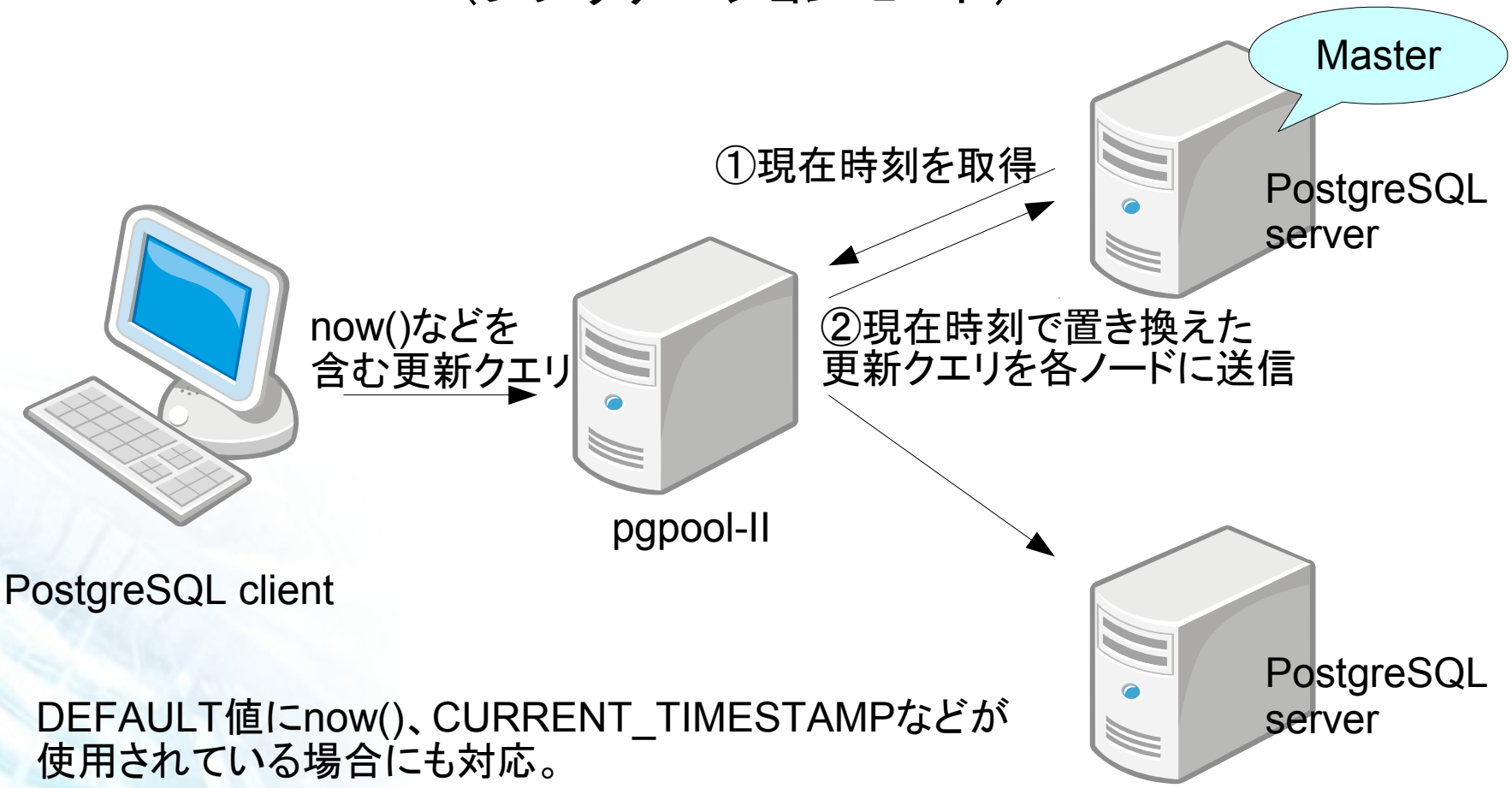
- 2.3（2009/12）
  時刻データに対応、停止時のノード状態で起動可能に
- 2.3.2(2010/02)
  SSL通信に対応、ラージオブジェクトに対応

オンラインリカバリとは



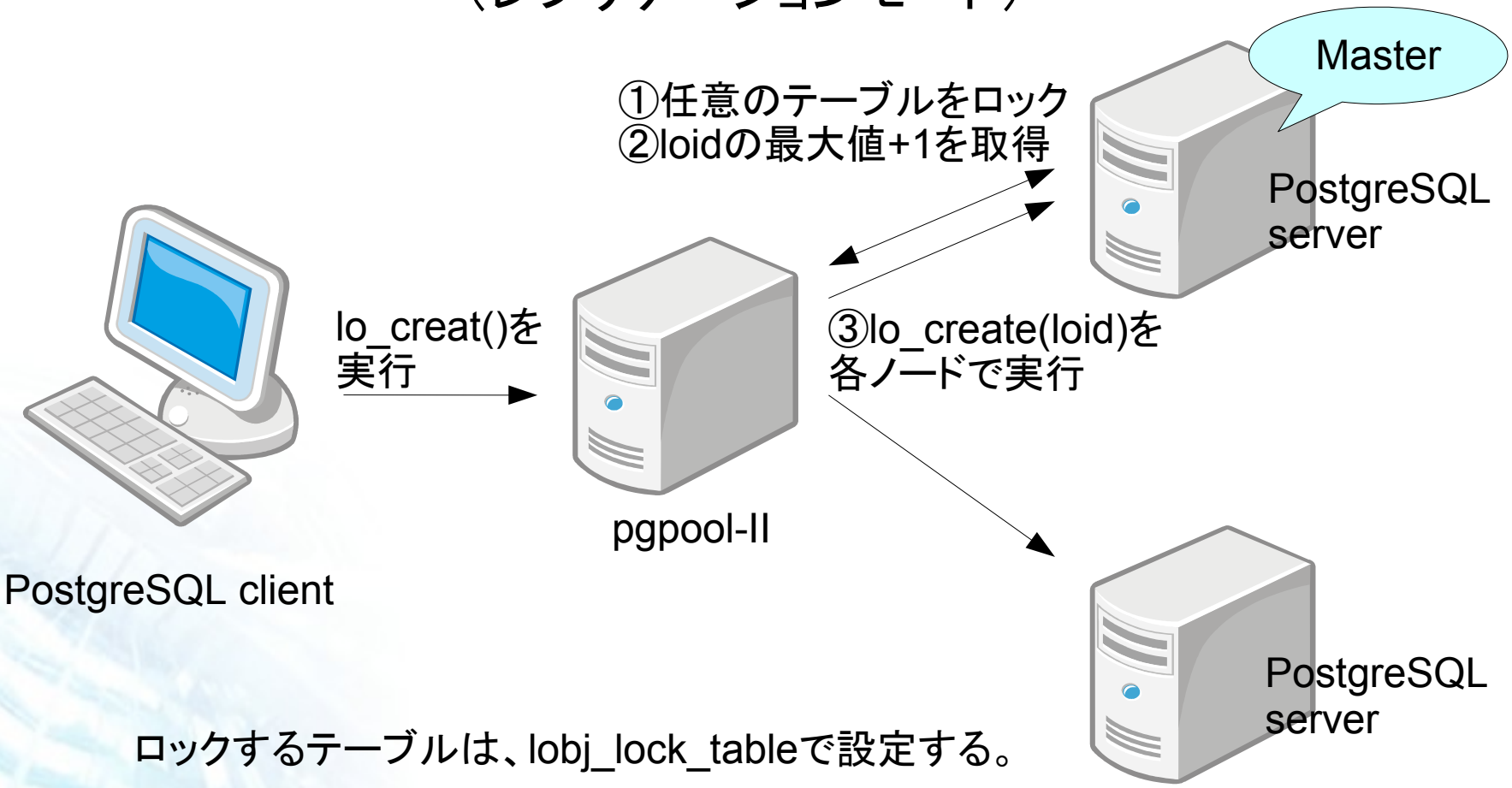
1st stageと2nd stageと呼ばれる2段階の処理で障害発生ノードを復旧する。2nd stageは新規接続をブロックする。
※フェイルオーバー時に既存の接続はすべて切断される。

時刻データへの対応
（レプリケーションモード）



DEFAULT値にnow()、CURRENT_TIMESTAMPなどが使用されている場合にも対応。

ラージオブジェクトへの対応
（レプリケーションモード）



ロックするテーブルは、lobj_lock_tableで設定する。

2010年9月
pgpool-II 3.0 リリース

Streaming Replicationに対応

- master_slave_sub_modeを追加
- Hot Stanby側の制約を考慮したクエリ振り分け
- 賢いロードバランス
  - 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（delay_threthold、log_standby_delay）
  - 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

Hot Stanby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

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)
```

追加された主なパラメータ

- white_function_list、black_function_list
  - 更新を伴う関数呼び出しを行うSELECT文を制御する
  - 以前は、/*REPLICATION*/、/*NO LOAD BALANCE*/といったコメントで一文ずつ対応
- failover_if_affected_tuples_mismatch
  - 更新結果の行数が異なった場合にフェイルオーバする
  - 以前は、トランザクションをROLLBACKするのみ
- debug_level
  - デバッグメッセージの出力を制御する
  - 設定ファイルのリロードでon/off可能

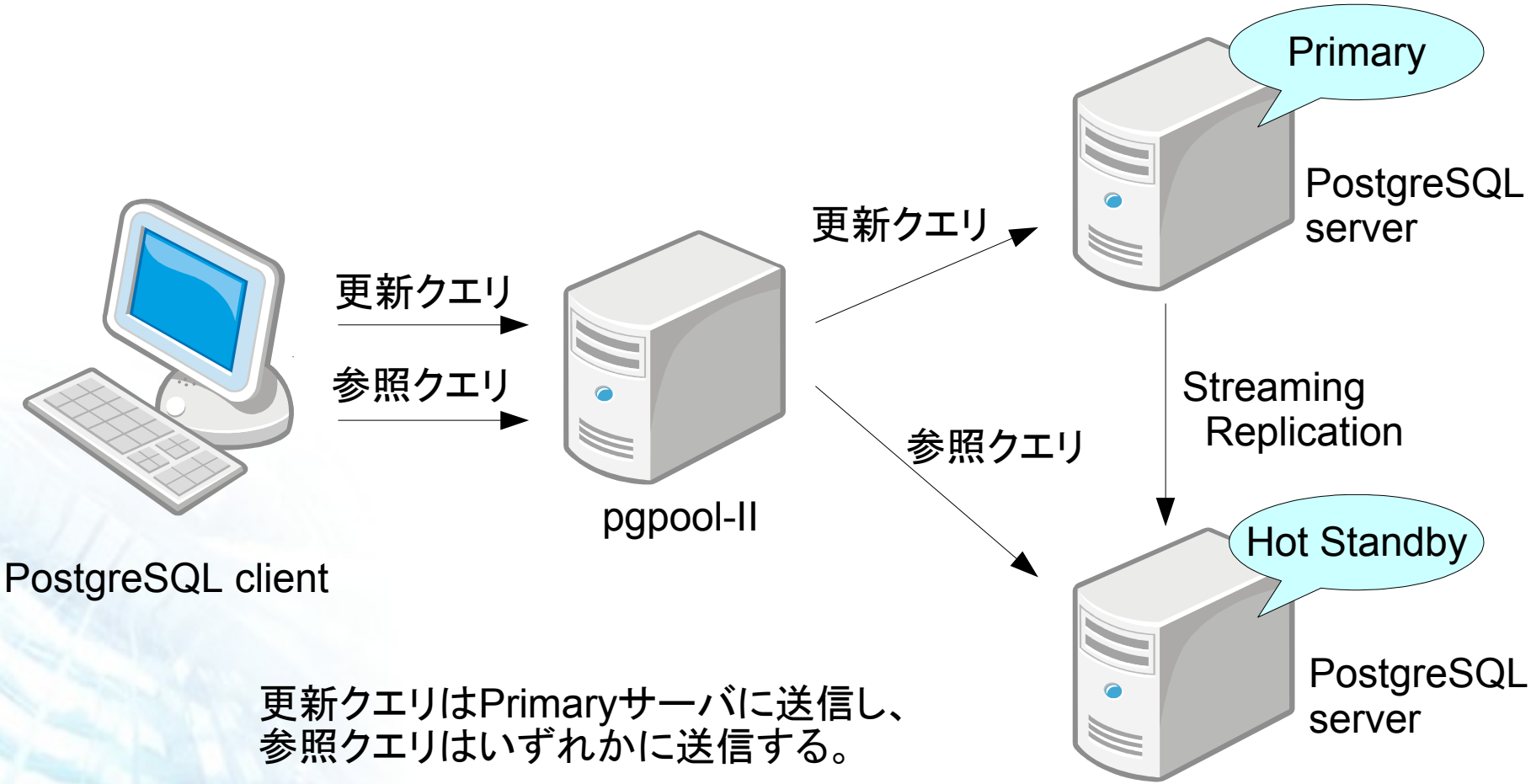
その他

- md5認証をサポート（レプリケーションモード、マスタースレーブモード）
- pgpool-IIの状態を表示するSHOWコマンドを追加
- オンラインリカバリ時に、既存の接続を強制切断することが可能に
- C関数を追加
  - pgpool_regclass(cstring)
    異なるスキーマに同じ名前のテーブルが存在し、SQL文でスキーマ名を省略している場合に不具合が生じることがある。pgpool_regclass(cstring)はそれに対処する関数。
  - pgpool_switch_xlog(text)
    pg_switch_xlog()は、アーカイブログの生成完了を待たずに終了するため、オンラインリカバリ時にやや不安が残る。pgpool_switch_xlog(text)は、アーカイブログの生成を待ってから終了する関数。
- 実装の見直しとリファクタリング

Streaming Replication/Hot Standby とpgpool-IIを組み合わせるメリット

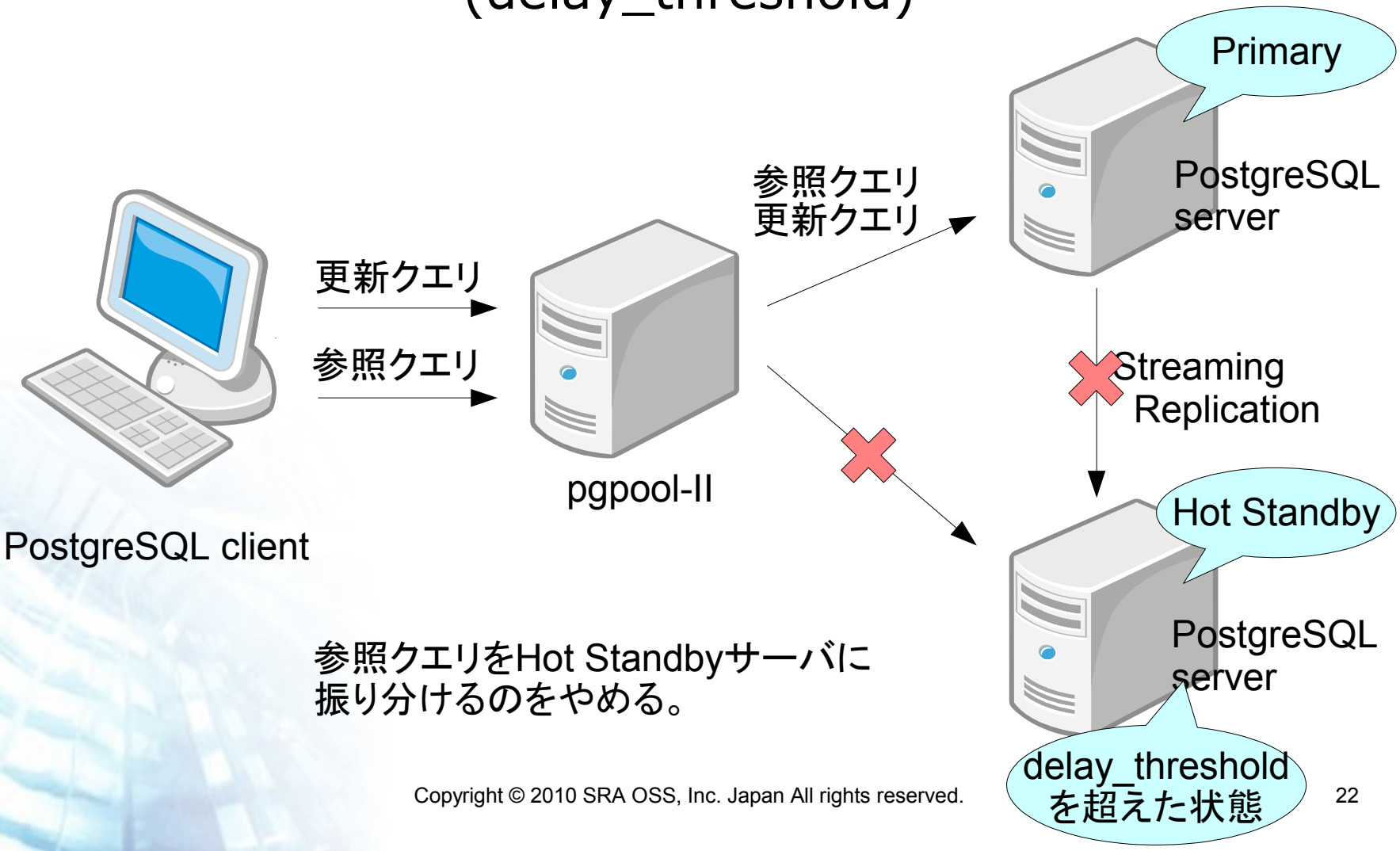
アプリケーション側を作りこまずに、
高可用性と負荷分散を簡単に実現できる

クエリ振り分けの自動化、ロードバランス



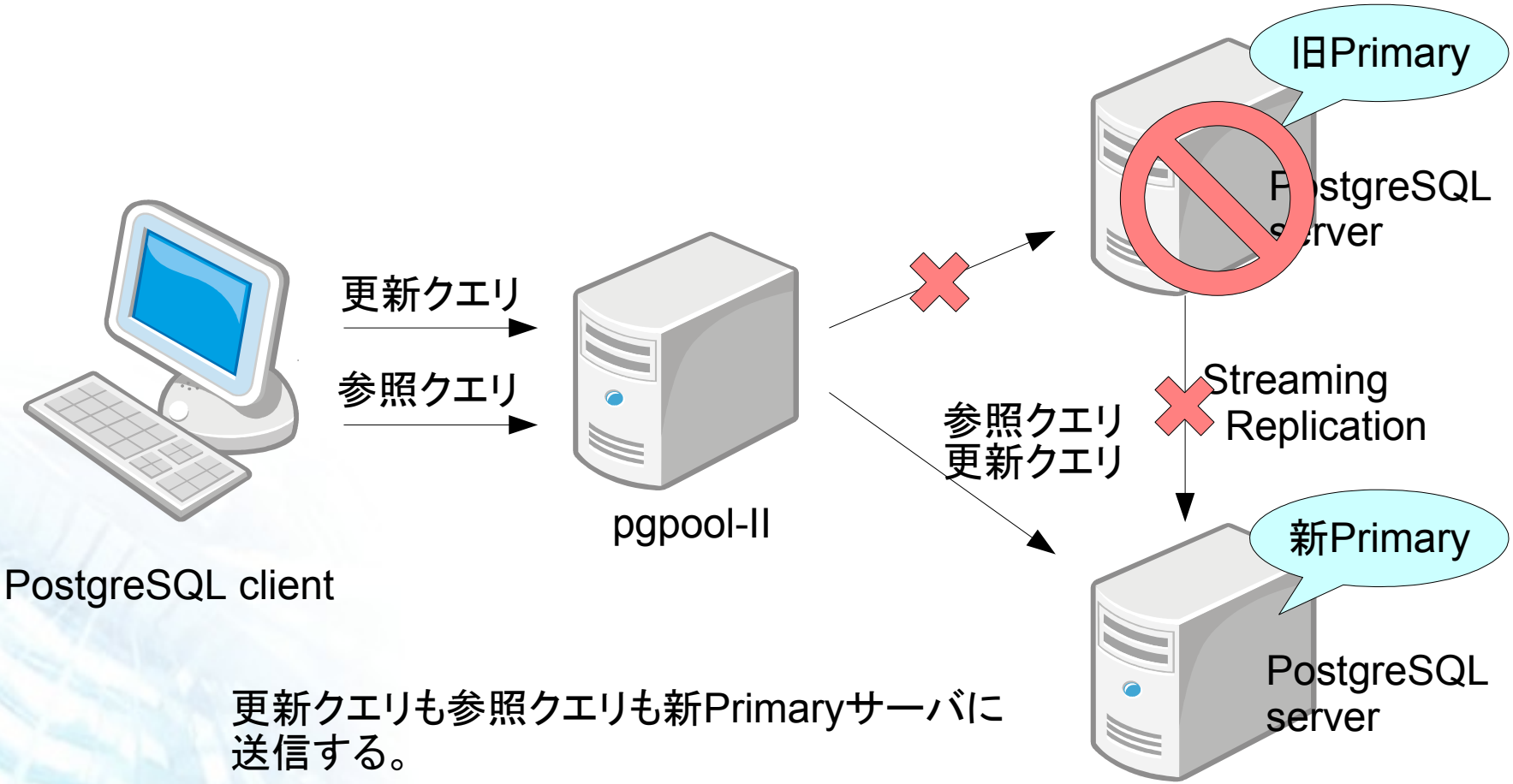
更新クエリはPrimaryサーバに送信し、
参照クエリはいずれかに送信する。

賢いロードバランス
(delay_threshold)



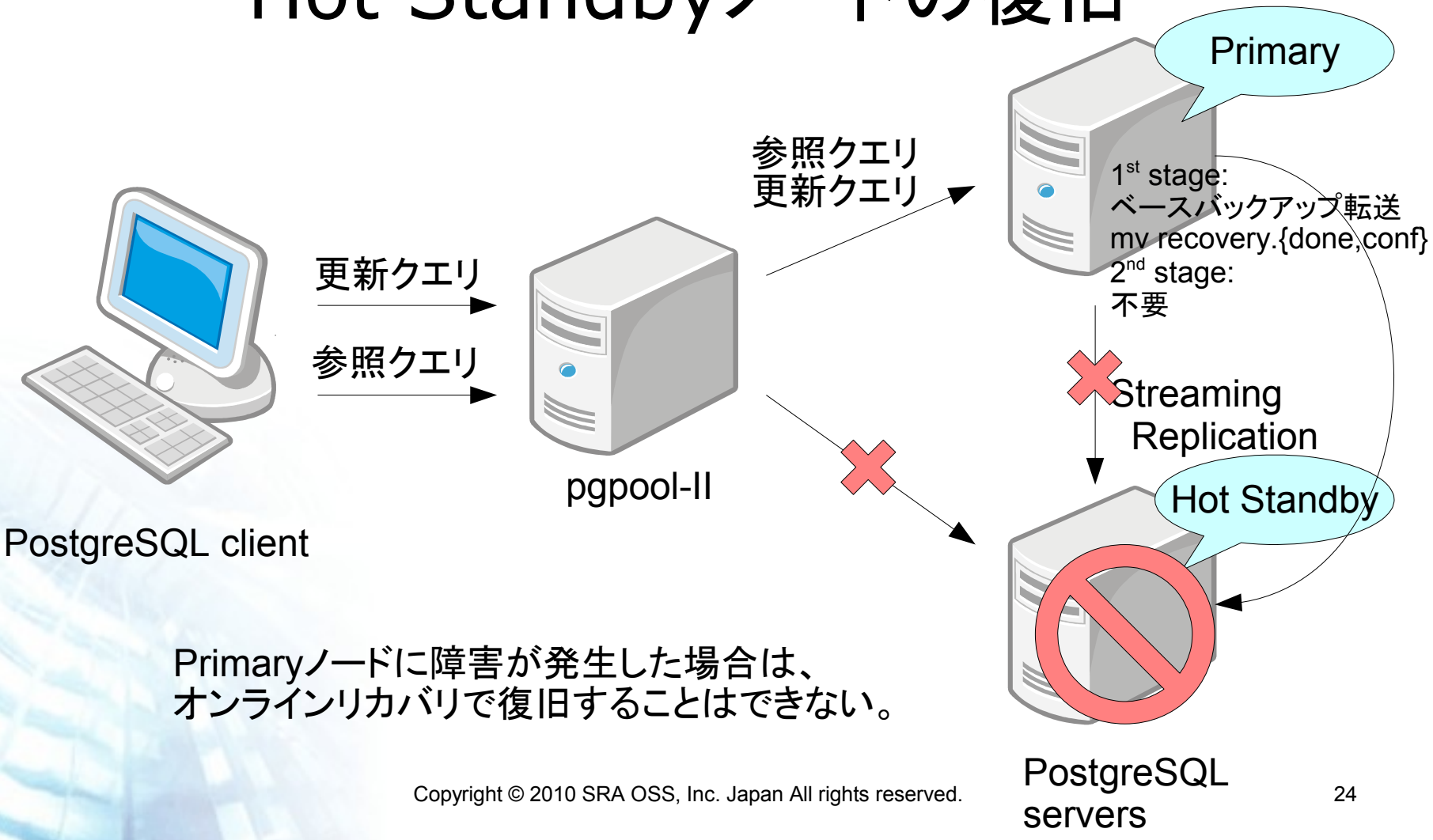
参照クエリをHot Standbyサーバに振り分けるのをやめる。

自動フェイルオーバー



更新クエリも参照クエリも新Primaryサーバに送信する。

オンラインリカバリを利用した Hot Standbyノードの復旧



Primaryノードに障害が発生した場合は、
オンラインリカバリで復旧することはできない。

参考URL

- pgpool-II開発サイト
  - http://pgfoundry.org/projects/pgpool/
- pgpool Wiki
  - http://pgpool.sraoss.jp/
- Let's PostgreSQL
  - http://lets.postgresql.jp/
- pgpoolメーリングリスト
  - http://www.sraoss.jp/mailman/listinfo/pgpool-general-jp

ご清聴ありがとうございました。