PostgreSQLクラスタ環境の管理機能を大幅に強化！

－ PostgreSQL のストリーミングレプリケーション対応が大幅に強化された、Pgpool-II 4.0 のご紹介 －

Pgpool-II Day
- Pgpool-II 4.0 Anniversary -
2018年9月25日

SRA OSS, Inc. 日本支社
彭 博（ペン ボ）
pengbo@sraoss.co.jp

自己紹介

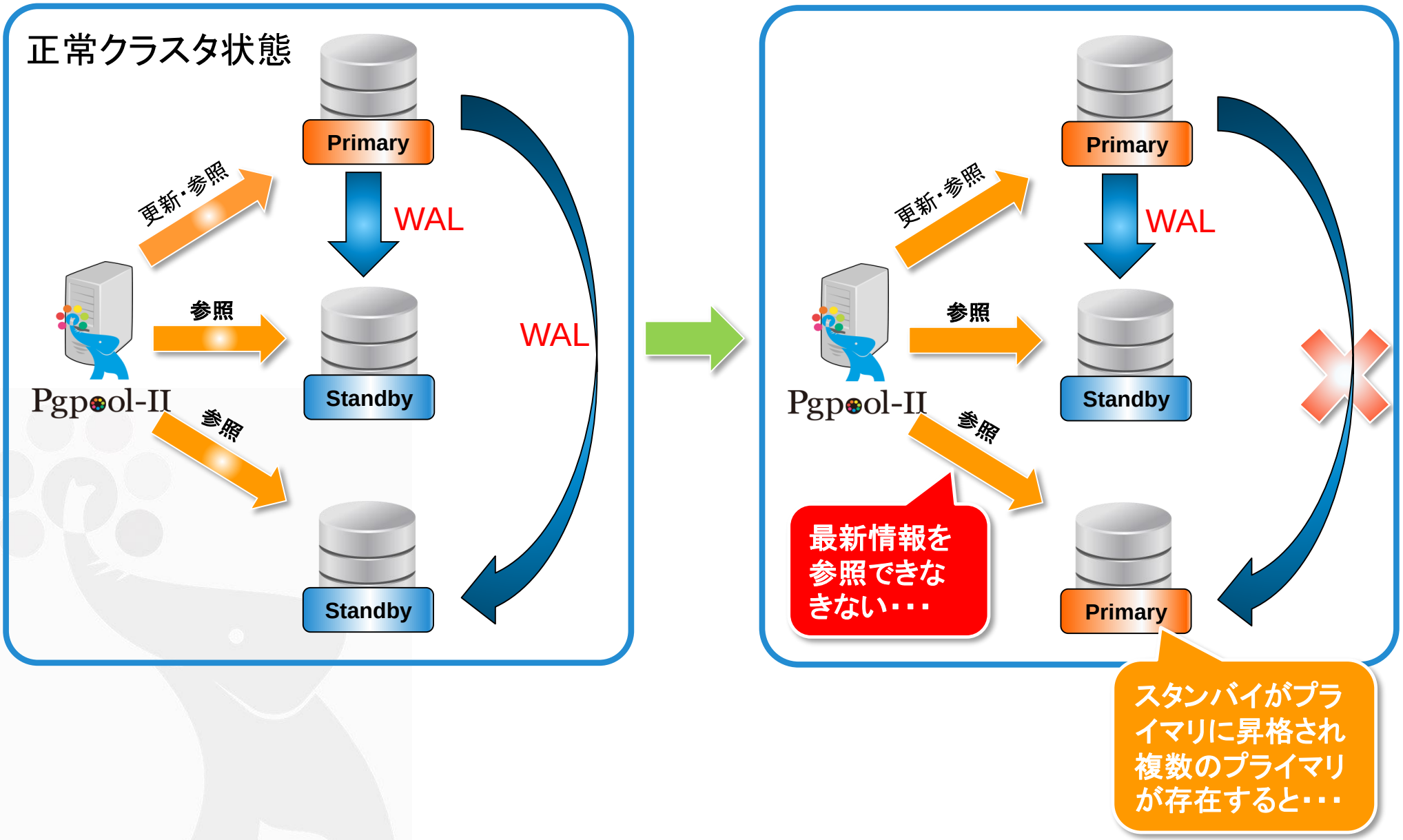
- 彭 博（ペン ボ）
  - 中国出身
  - 2014 年に SRA OSS, Inc. 日本支社に入社
  - 基盤技術グループ
- オープンソースソフトウェアの技術サポート
  - Zabbix、Pacemaker など
- Pgpool-II 開発者
  - 2016 年から
  - 最初は中国語のドキュメント作成
  - ML の対応、不具合修正及び新機能追加

4.0 の新機能紹介

- 不正なプライマリサーバの検出
- 負荷分散の高度化
- SHOW POOL NODES コマンドの機能強化
- PostgreSQL 11 パーサの取り込み
- クライアントメッセージの改善

不正なプライマリサーバの検出

不正なプライマリサーバの検出①



不正なプライマリサーバの検出②

こういった状態をどう検出するか？

新しいパラメータ:
detach_false_primary

- 不正なプライマリを検出し切り離す
- 正しいプライマリ: すべてのスタンバイに接続している

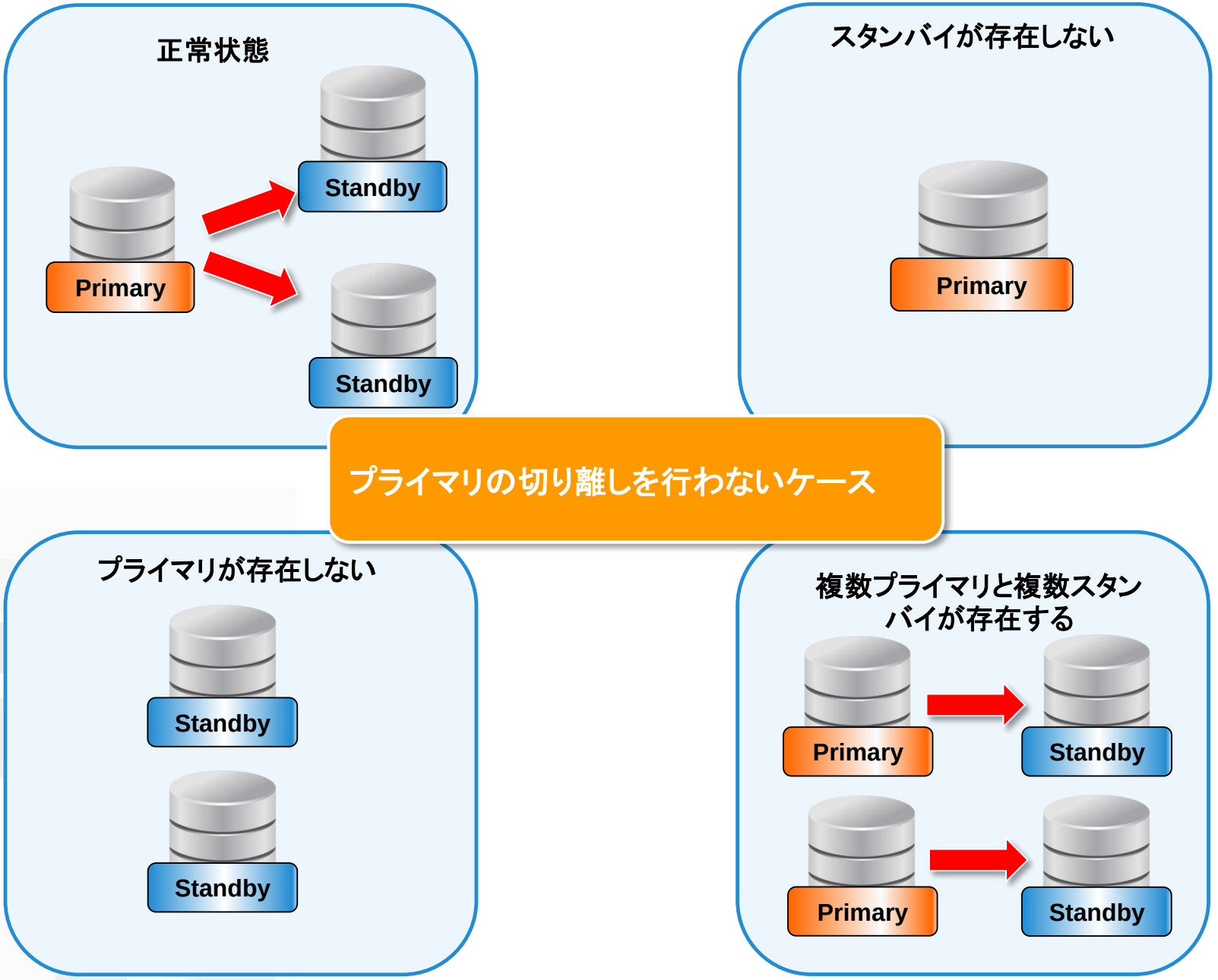
  ″不正″ なプライマリ: 上記以外
- pg_stat_wal_receiver を使用し、プライマリとスタンバイノードの間の接続性を検証
- PostgreSQL 9.6 以降で利用可能

```
postgres=# show pool_nodes;
 node_id | hostname | port  | status | lb_weight |  role   | last_status_change
---------+----------+-------+--------+-----------+---------+---------------------
 0       | /tmp     | 11002 | up     | 0.333333  | primary | 2018-09-08 23:36:24
 1       | /tmp     | 11003 | up     | 0.333333  | standby | 2018-09-08 23:36:24
 2       | /tmp     | 11004 | down   | 0.333333  | standby | 2018-09-08 23:37:05
```

“不正”な
プライマリ

切り離す

不正なプライマリサーバの検出③

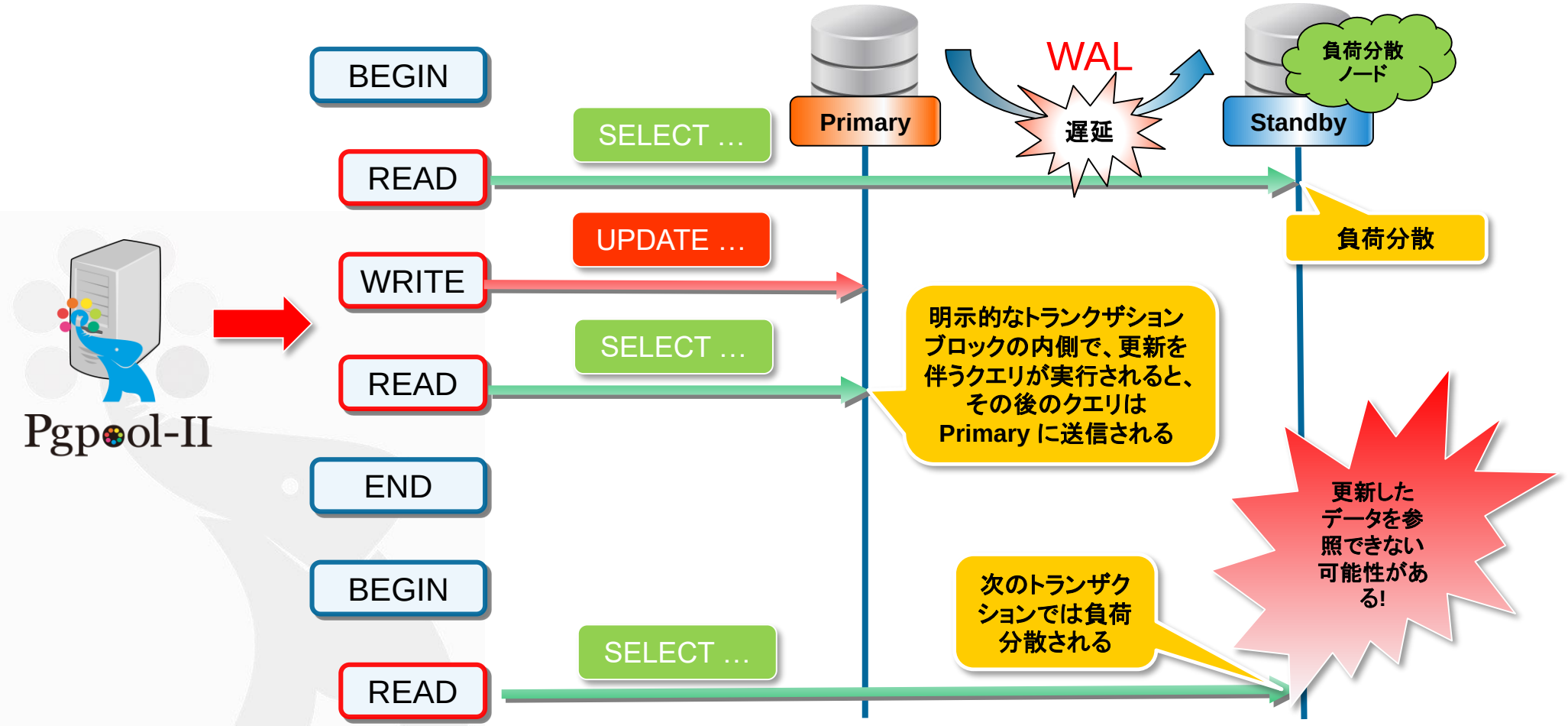


負荷分散の高度化 ①

disable_load_balance_on_write

負荷分散機能の高度化 ① disable_load_balance_on_write

- **負荷分散**
  - 参照クエリを自動的に PostgreSQL サーバに振り分けることにより、検索性能を向上させる
- **Pgpool-II 3.7 まで**
  - 明示的なトランクザションの内側で、更新を伴うクエリが実行されると、更新したデータを読み取るために、トランザクションが終了するまで負荷分散しないようになっている
  - 後続のトランザクションで参照クエリが負荷分散される



負荷分散機能の高度化 ① disable_load_balance_on_write

3.7 までの問題点

- データの一貫性を保つことができたが、負荷分散してもいい場合には性能が低下してしまう
- WAL 転送に遅延が発生した場合に更新したデータを参照できない可能性がある

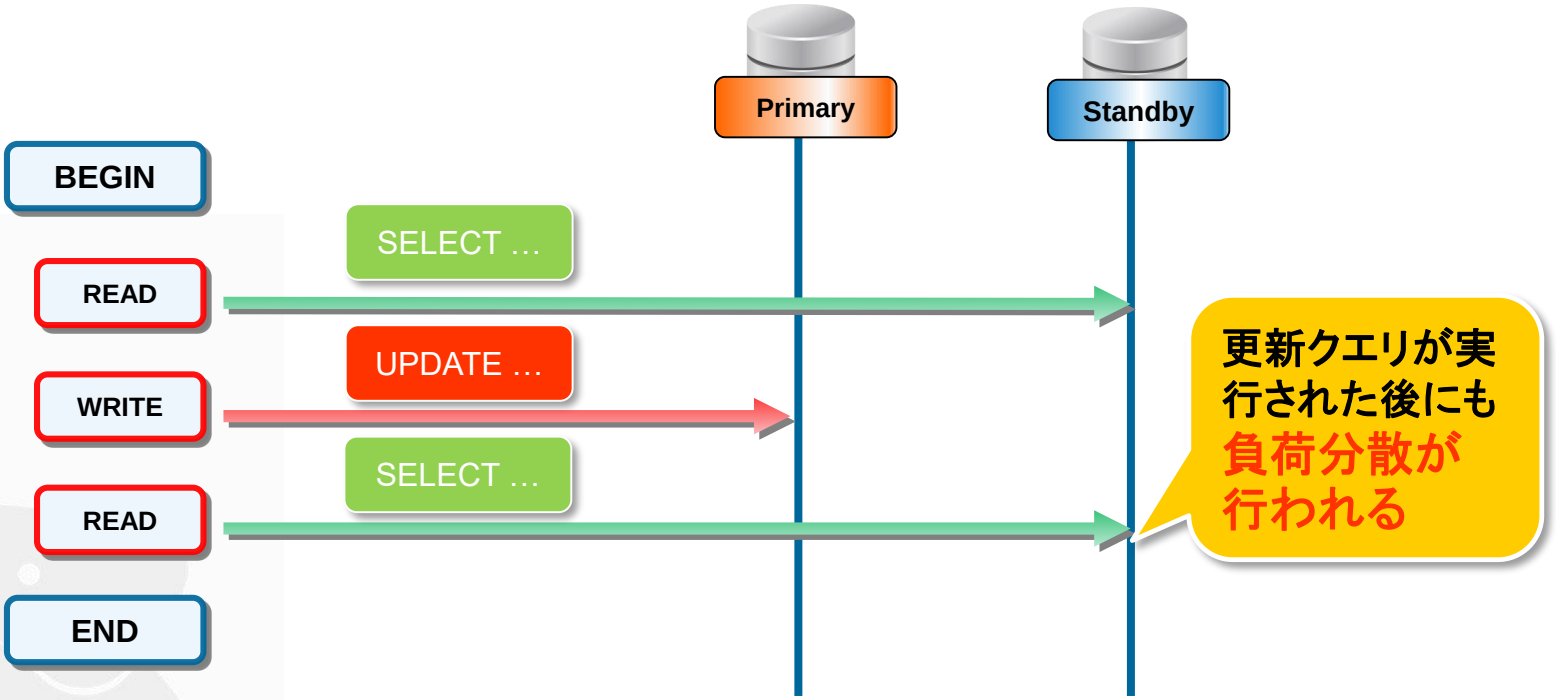
disable_load_balance_on_write

- 更新クエリが発行された時の負荷分散の振る舞いを制御できるようになる
- off/transaction/trans_transaction/always

負荷分散機能の高度化 ① disable_load_balance_on_write

disable_load_balance_on_write = off

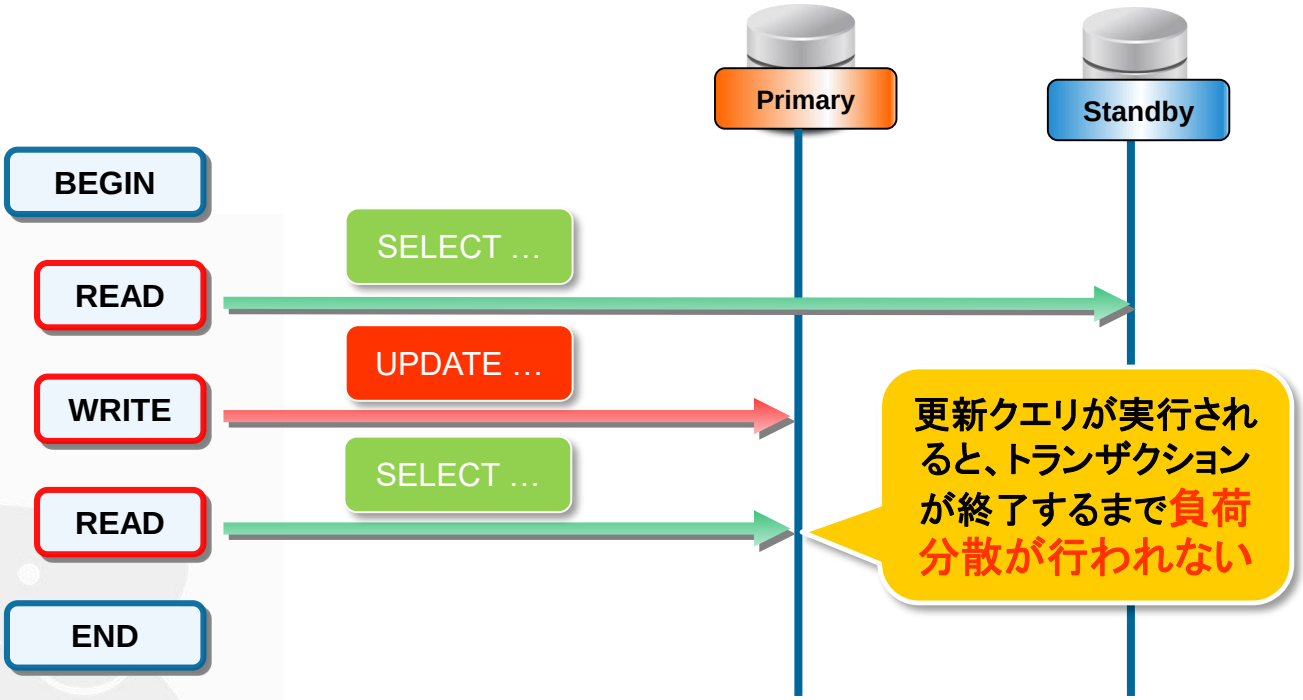
- 更新クエリが来ても負荷分散が行われる



負荷分散機能の高度化 ① disable_load_balance_on_write

disable_load_balance_on_write = transaction

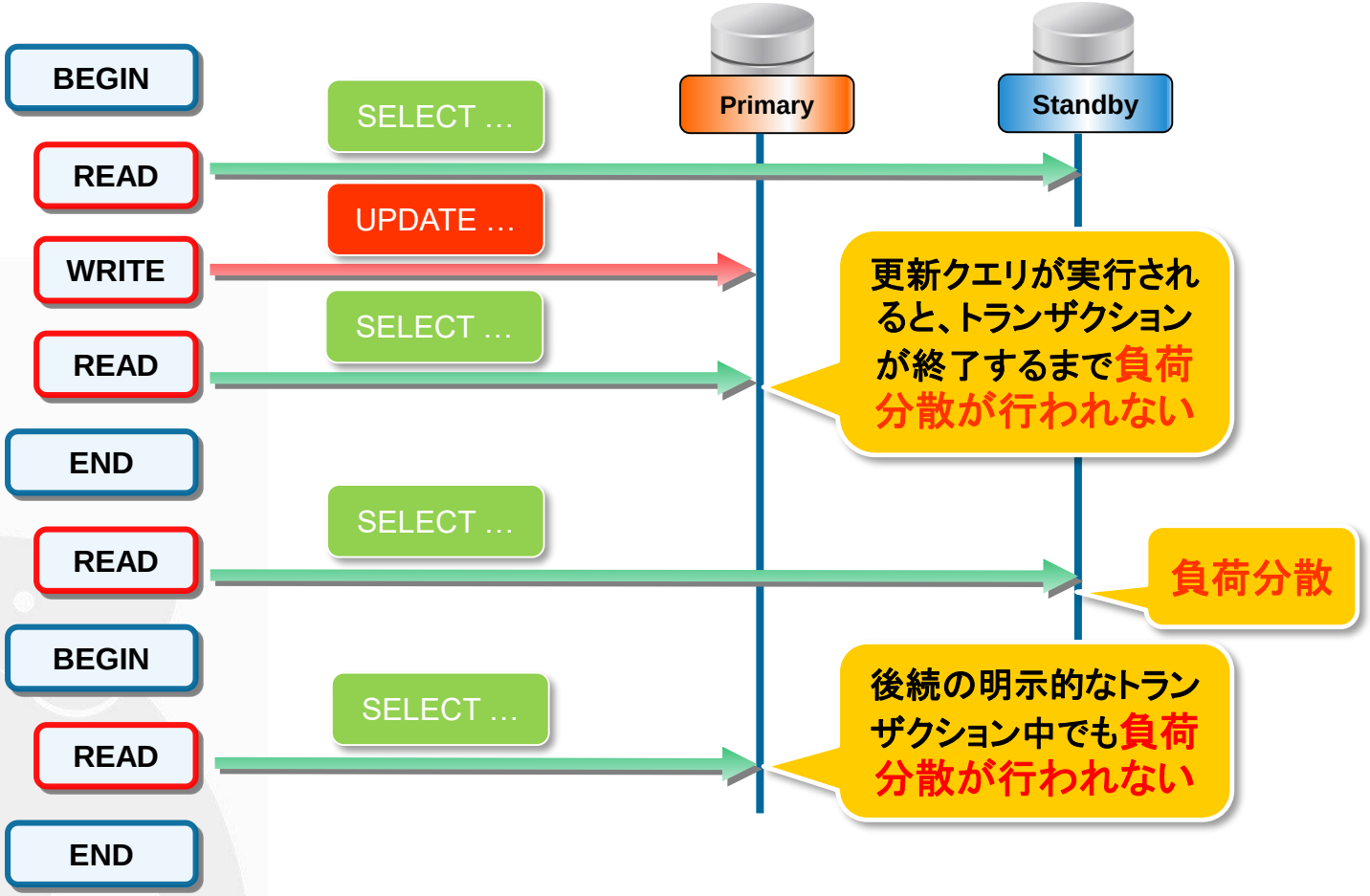
- 明示的なトランザクション中に更新クエリが来ると、トランザクションが終了するまで負荷分散が行われない
- Pgpool-II 3.7 以前のバージョンと同じ振る舞い



負荷分散機能の高度化 ① disable_load_balance_on_write

disable_load_balance_on_write = trans_transaction

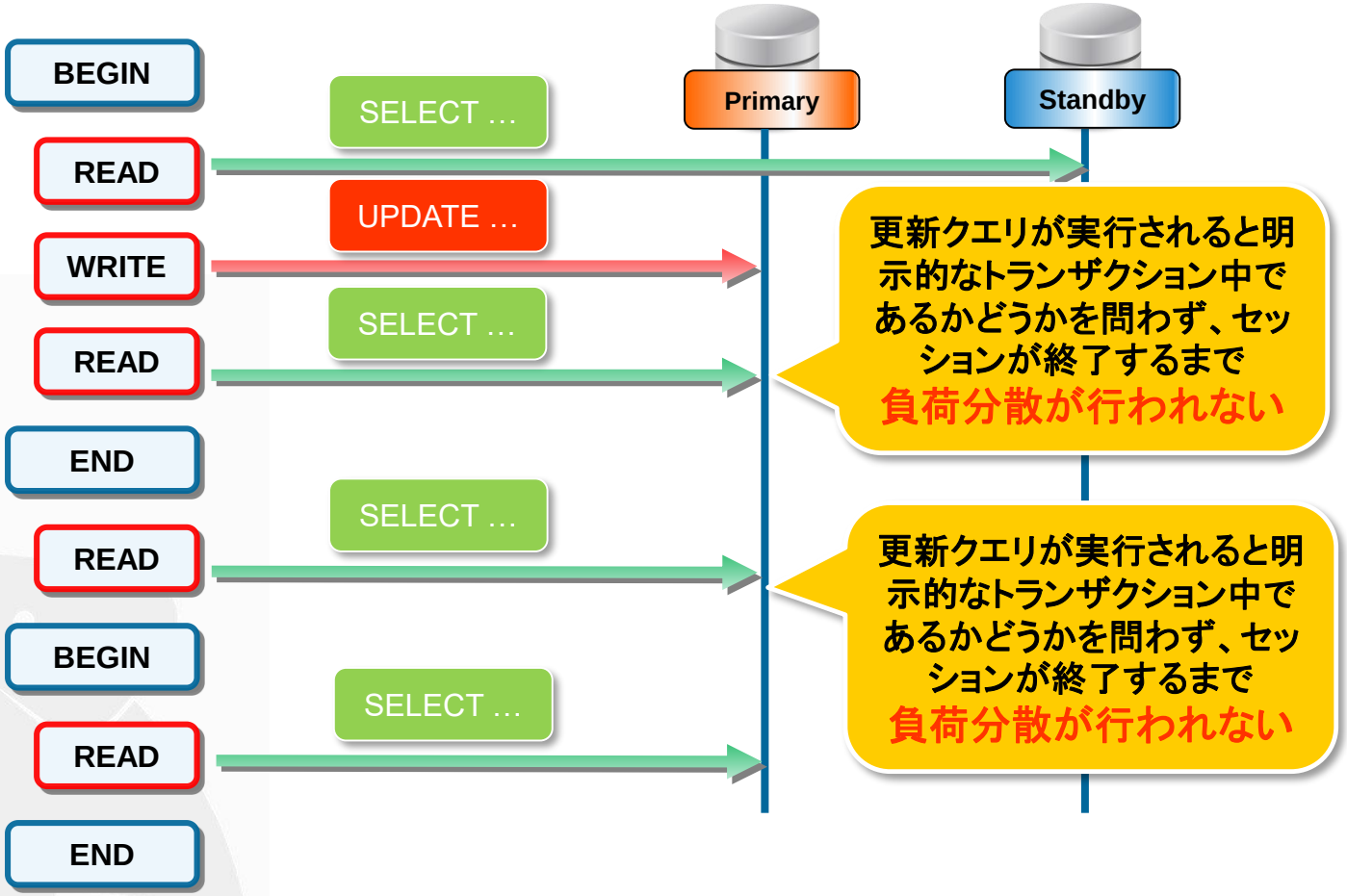
- 明示的なトランザクション中に更新クエリが来ると、トランザクションが終了するまで負荷分散が行われない
- また、セッションが終了するまで後続の明示的なトランザクション中でも負荷分散が行われない



負荷分散機能の高度化 ① disable_load_balance_on_write

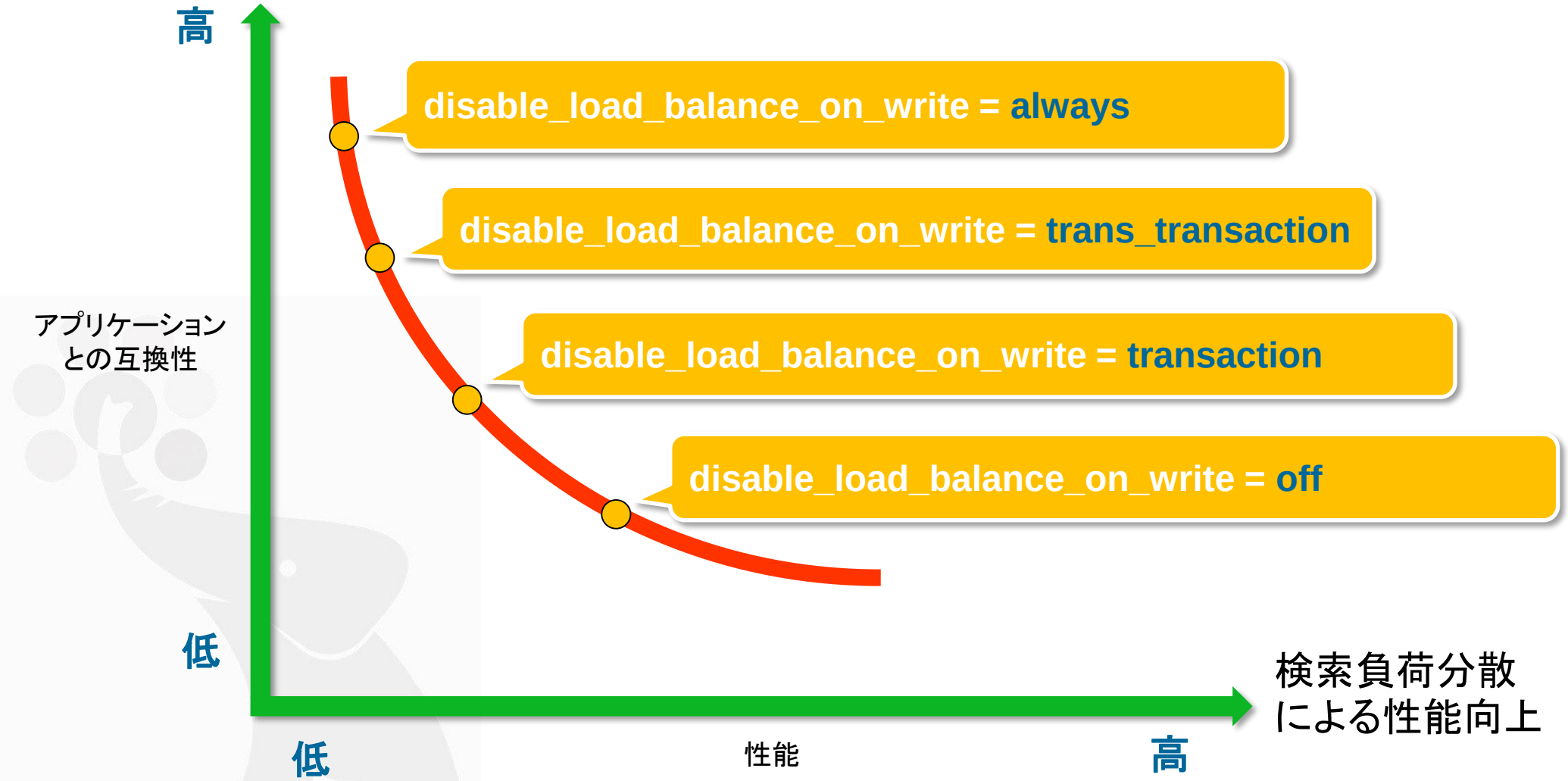
disable_load_balance_on_write = always

- 更新クエリが来ると、明示的なトランザクション中であるかどうかを問わず、セッションが終了するまで負荷分散が行われない



負荷分散機能の高度化 ① disable_load_balance_on_write

クラスタを考慮していない
レガシーアプリケーション
との互換性

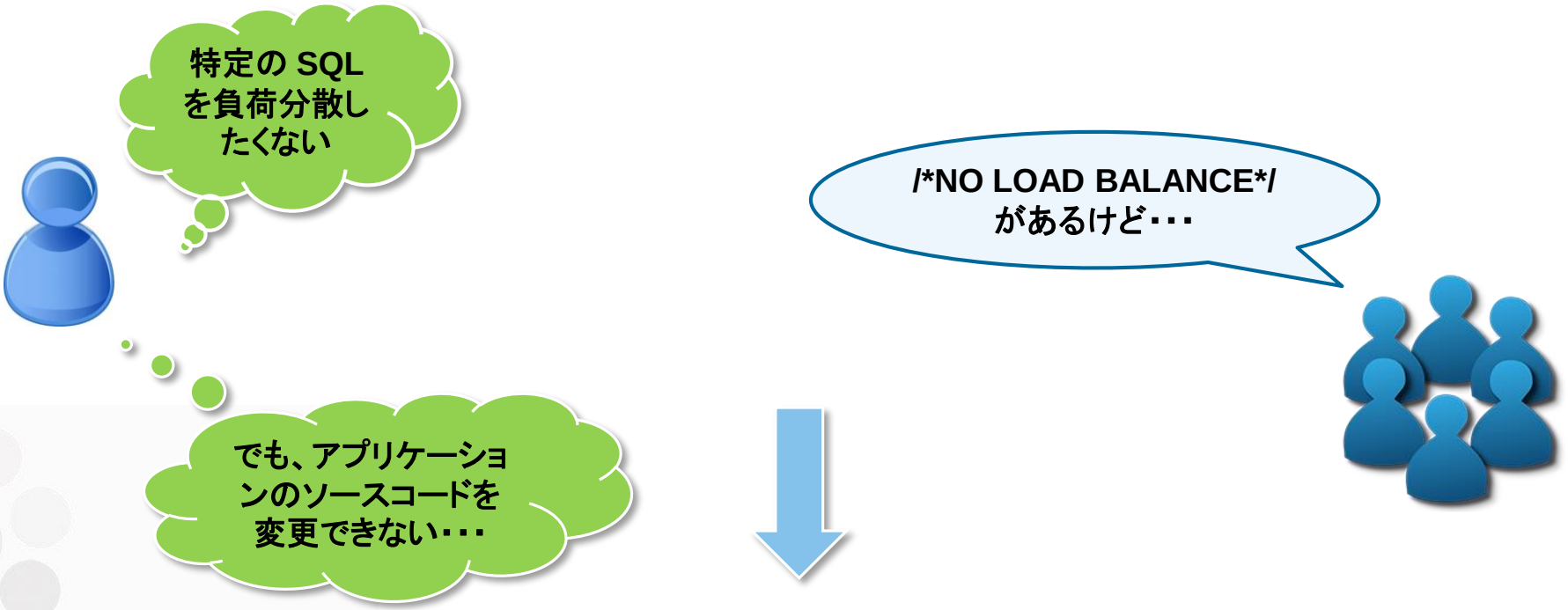


負荷分散の高度化 ②

black_query_pattern_list

負荷分散機能の高度化 ② black_query_pattern_list

ユーザからの要望

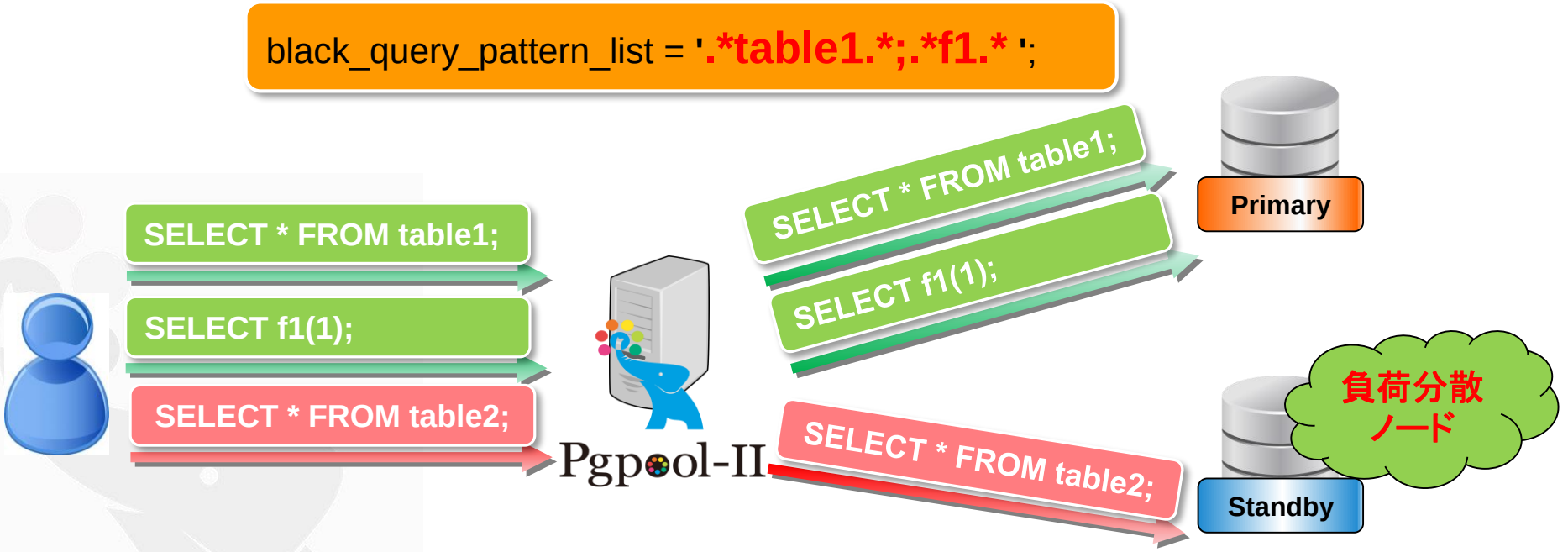


新しいパラメータ:

black_query_pattern_list

負荷分散機能の高度化 ② black_query_pattern_list

- black_query_pattern_list
  - 指定した SQL パターンに一致するクエリを Primary のみに送信
  - セミコロン区切りで複数SQLパターンを指定可能
  - 正規表現が使用可能
  - 正規表現の特殊文字は「¥」でエスケープが必要



注意: black_query_pattern_list と white_function_list の両方にマッチした場合、white_function_list の設定が無視され、プライマリノードに送信します。

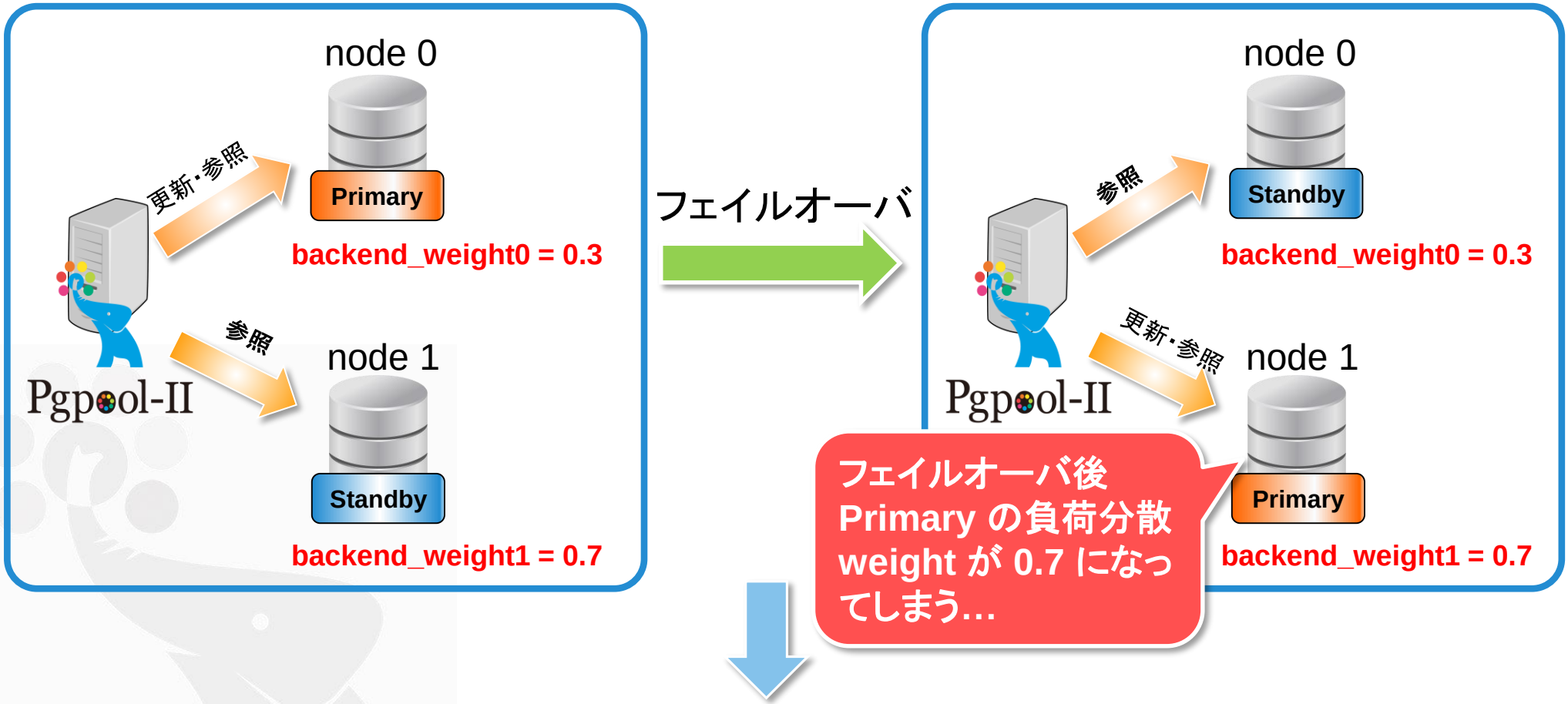
負荷分散の高度化 ③

パラメータによる負荷分散 weight の指定

負荷分散機能の高度化 ③ 負荷分散 weight の指定

30% の参照クエリを Primary に送信

- **backend_weight0 = 0.3**
- **backend_weight1 = 0.7**



常にプライマリの負荷分散 weight を一定な値に維持したい

負荷分散機能の高度化 ③ 負荷分散 weight の指定

app_name_redirect_preference_list

特定のクライアントアプリケーションの接続で参照クエリを特定のバックエンドノードに送信

database_redirect_preference_list

特定のデータベース接続で参照クエリを特定のバックエンドノードに送信

3.7 まで

負荷分散 weight を指定できない

app_name:primary

すべての参照クエリが primary に送信される

4.0 から

負荷分散 weight を指定できる

app_name:primary(0.3)

- **30% の参照クエリが primary に送信される**
- **フェイルオーバ後でも weight がかわらない**

SHOW POOL_NODES コマンドの機能強化

「last_status_change」カラムの追加

「last_status_change」カラムの追加

last_status_change:
「status」、あるいは「role」の変更時刻

- unused
- connect_wait
- up
- down
- quarantine

- primary
- standby

直近でフェイルオーバが発生したときのログを探す際に、last_status_change を手がかりにできる

```
postgres=# show pool_nodes;
 node_id | hostname | port  | status | lb_weight |  role   | last_status_change
---------+----------+-------+--------+-----------+---------+---------------------
 0       | /tmp     | 11002 | up     | 0.500000  | primary | 2018-09-10 10:36:24
 1       | /tmp     | 11003 | up     | 0.500000  | standby | 2018-09-10 10:36:24
```

PCP 及び pgpool_adm にも追加

```
$ pcp_node_info -U pengbo -p 11001 -n 0
/tmp 11002 2 0.500000 up primary 0 2018-09-10 10:37:36
```

```
test=# SELECT * FROM pcp_node_info(0,'',11001,'pengbo','pengbo');
 host | port  |      status       | weight |  role   | replication_delay | last_status_change
------+-------+-------------------+--------+---------+-------------------+---------------------
 /tmp | 11002 | Connection in use |      0 | Primary |                 0 | 2018-09-10 11:06:18
```

PostgreSQL 11 パーサの取り込み

PostgreSQL 11 パーサの取り込み

- Pgpool-II は SQL パーサを持っている
  - 複雑な SQL を正確に解析するため
  - クエリの書き換えを行うため
- メジャーリリースを行うたびに、PostgreSQL の最新パーサを移植
- Pgpool-II 4.0 は PostgreSQL 11 のパーサを導入
  - CREATE/ALTER/DROP PROCEDURE
  - CALL
  - ALTER/DROP ROUTINE
  - CREATE INDEX ... INCLUDE ...
  - { RANGE | ROWS | GROUPS } frame_start [frame_exclusion]
  - VACUUM/ANALYZE <table1>, <table2>

クライアントメッセージの改善

クライアントメッセージの改善

- クライアントメッセージ
  - クライアントから Pgpool-II へのメッセージ
- 3.7 まで
  - クライアントメッセージを記録するために、デバッグメッセージを有効にする必要があった
  - 大量のデバッグメッセージが出力されてしまう
- 新しいパラメータ log_client_messages
  - log_client_messages = on にすると、デバッグメッセージなしでクライアントメッセージのみをログ出力が可能

log_client_messages = on

Parse

Bind

Execute

Close

```
LOG:  Parse message from frontend.
DETAIL:  statement: "S2", query: "SELECT 1 FROM pgbench_accounts"
LOG:  Bind message from frontend.
DETAIL:  portal: "P1", statement: "S2"
LOG:  DB node id: 0 backend pid: 24797 statement: B message
LOG:  Execute message from frontend.
DETAIL:  portal: "P1"
LOG:  DB node id: 0 backend pid: 24797 statement: Execute: SELECT 1 …
LOG:  Close message from frontend.
DETAIL:  statement: "S2"
LOG:  DB node id: 0 backend pid: 24797 statement: C message
…
```

互換性のない変更点

- リカバリスクリプトが 5つのパラメータを受け取るようになった
  - $5: リカバリされるノード番号
  - もし 5番目のパラメータの情報を必要としないのであれば、従来 pgpool_recovery() 関数も利用できる
- パラメータ名の変更
  - **fail_over**_on_backend_error => **failover**_on_backend_error
  - 4.0 から古いパラメータ名を使用する場合、警告メッセージが表示される
- pgpool.conf に AES 暗号化パスワードを指定可能
  - health_check_password、sr_check_password、wd_lifecheck_password 及び recovery_password
  - クリアテキストパスワードの場合、「TEXTpassword」のようにパスワードの先頭に「TEXT」をつける必要がある
  - 空パスワードを指定した場合、pool_passwd からユーザパスワードの取得を試みる

ご清聴ありがとうございました。