【2022年最新版】PostgreSQL専用のミドルウェア Pgpool-II 4.4の新機能紹介

OSC2023 Online/Osaka
2023/01/28

SRA OSS LLC
彭 博 (ペン ボ)

自己紹介

ペン ボ

- 名前：彭 博 (Bo Peng)
  pengbo@sraoss.co.jp
- 所属：SRA OSS LLC
  基盤技術グループ
- 職務：
  - OSS技術サポート、ミドルウェア構築
  - PostgreSQLクラスタ管理ツールPgpool-II開発者

本日の流れ

- Pgpool-IIとは
- Pgpool-IIの機能紹介
  - 自動フェイルオーバ
  - Watchdog
  - 負荷分散
- Pgpool-II 4.4新機能紹介

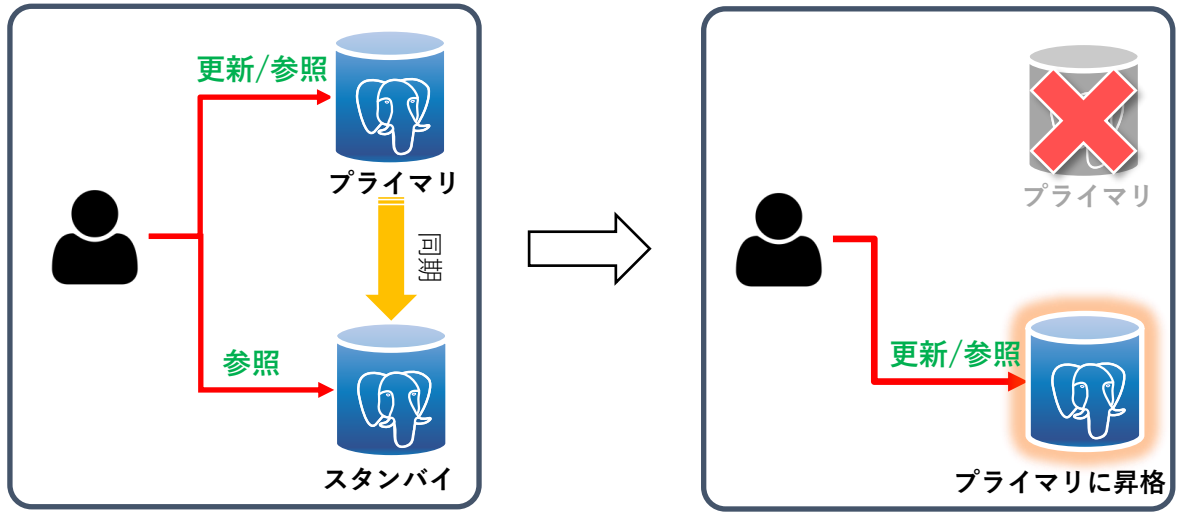
なぜクラスタは必要なの？

高可用性

- データの冗長化
- 稼働系に障害が発生した際に、待機系に切り替えてサービスを継続させる
- ダウンタイムを最小限に抑える

負荷分散

- レプリケーションされている複数台のPostgreSQLクラスタで、参照クエリをいずれかのPostgreSQLに振り分ける
- プライマリの負荷を下げる
- 性能向上



PostgreSQLクラスタの課題

自動フェイルオーバ/フェイルバックの仕組みがない

- 障害が発生した際に手動での切り替えが必要
  - 障害の検知
  - 待機系の昇格
  - 残りの待機系の同期元の変更
  - クライアントの接続先情報の変更
- 障害が発生したノードの復旧
  - 障害が発生したノードを復旧し、クラスタに再度組み込む

更新クエリと参照クエリを振り分ける機能を提供していない

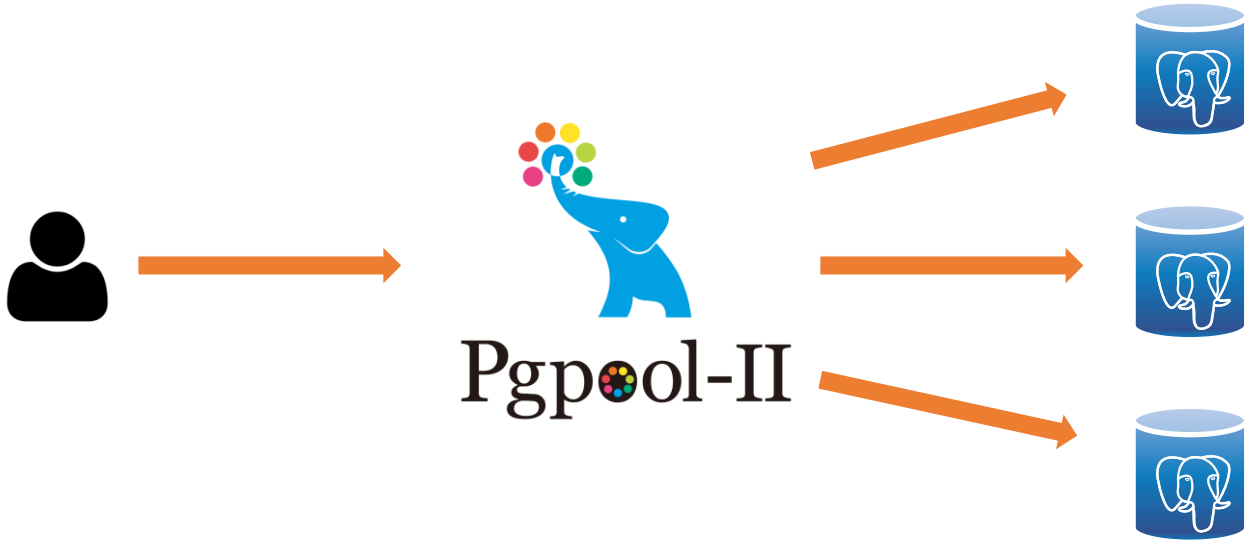
- プライマリは更新クエリ/参照クエリを処理でき、スタンバイは参照クエリのみ処理できる
- アプリケーション側でクエリを振り分ける処理を実装する必要がある

PostgreSQLクラスタの課題を解決するために

Pgpool-IIを利用することでこれらの課題を解決できる

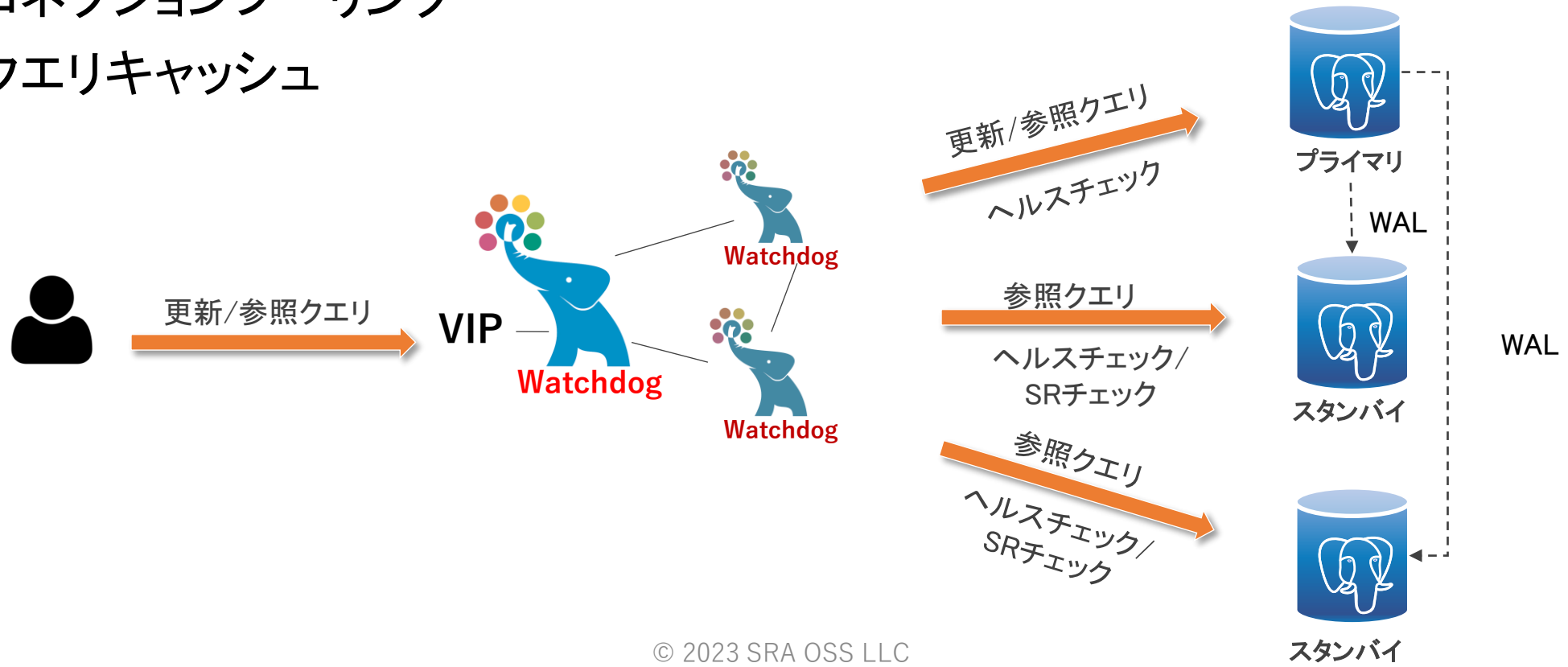
Pgpoolとは

- クライアントとPostgreSQLの間で動作するミドルウェア
- ユーザは複数PostgreSQLサーバを意識せず、1台のように見える



Pgpool-IIの主な機能

- 負荷分散
- 自動フェイルオーバ
- Watchdog
- コネクションプーリング
- クエリキャッシュ



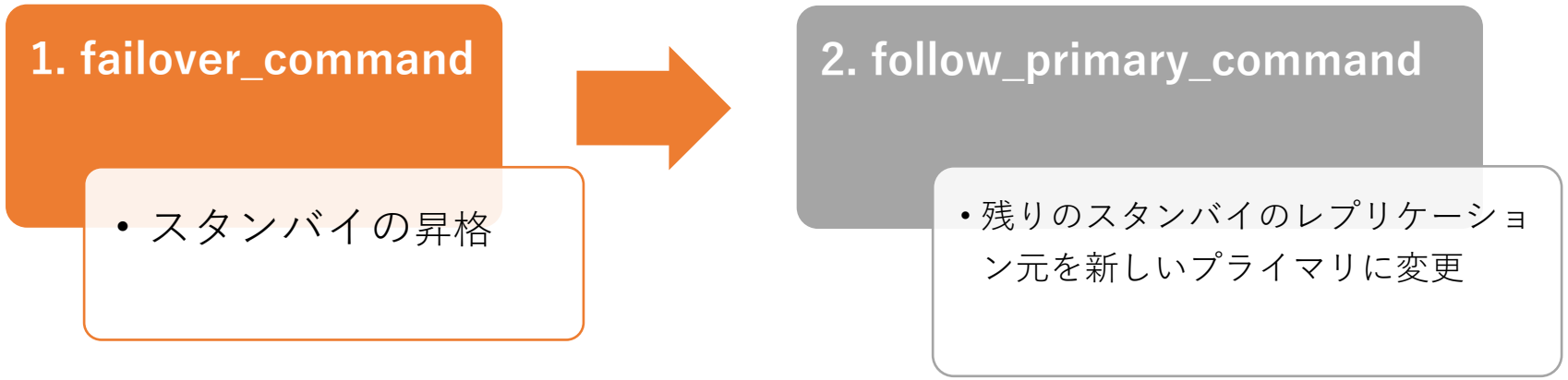
自動フェイルオーバ

フェイルオーバの契機

- ヘルスチェックでダウンと判定された場合
  - ヘルスチェック：各PostgreSQLノードの状態を監視するプロセス
  - 実行間隔、最大リトライ回数などを設定可能
- PostgreSQLへの接続時、および接続後にネットワーク通信エラーが発生した場合
  - failover_on_backend_error = onの場合

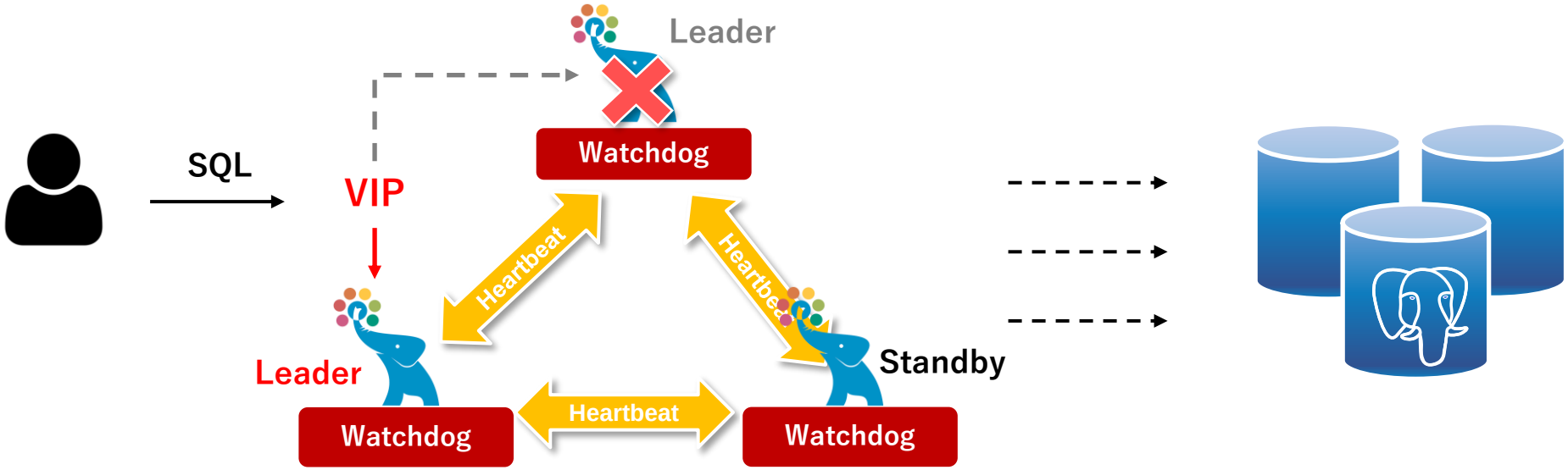
フェイルオーバ処理

- プライマリがダウンした場合
  - ダウンしたノードの状態を変更し（up->down）、以下のパラメータに設定されているスクリプトを実行



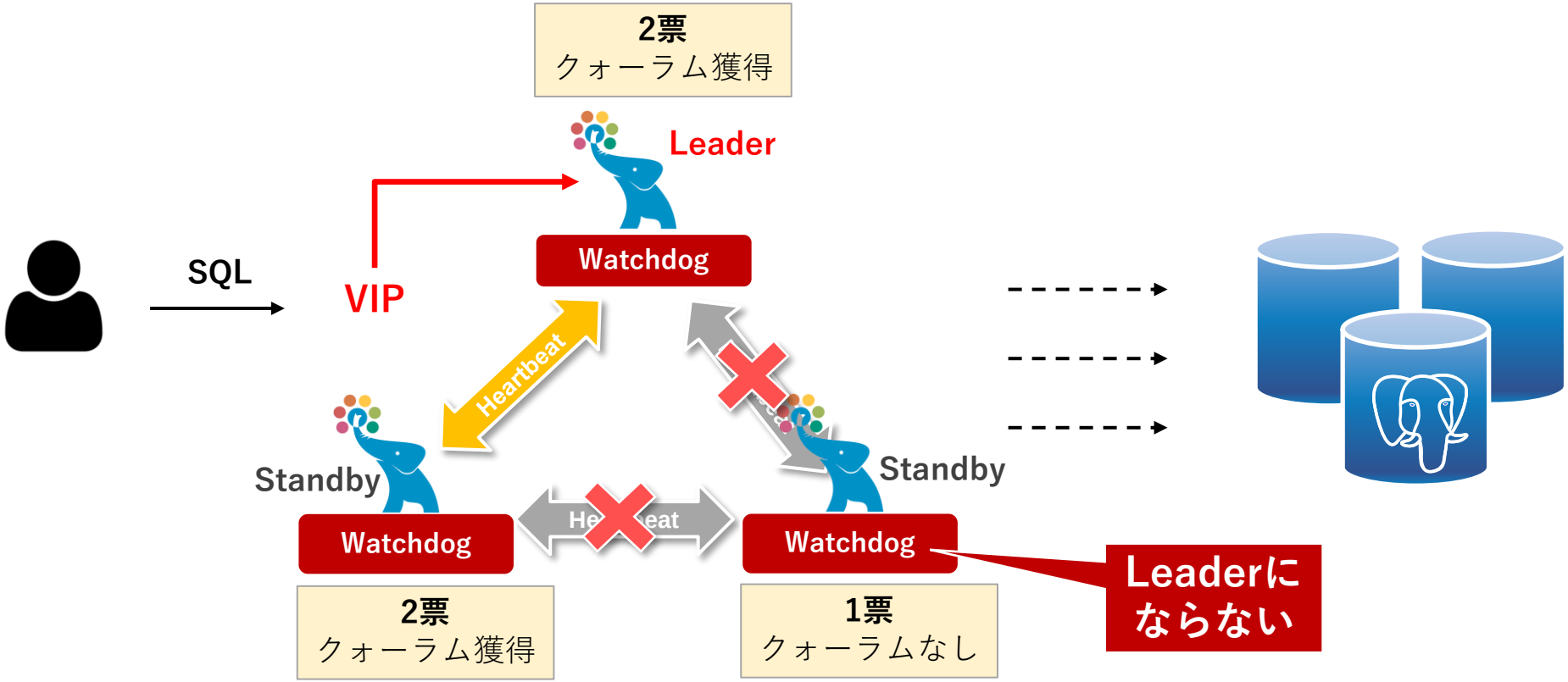
Pgpool-IIの高可用性（Watchdog）

- Pgpool-IIの単一障害点を回避
- 複数のPgpool-IIがお互いに監視することで、Pgpool-IIを冗長化するための機能
- 定期的に他のPgpool-IIノードにハートビート信号を送信
- Pgpool-IIノードの障害が検出された際に、Watchdogは投票によって新しいリーダーを決定し、切り替える
- 仮想IPの自動切り替え
- バックエンドノードのフェイルオーバの動作を制御



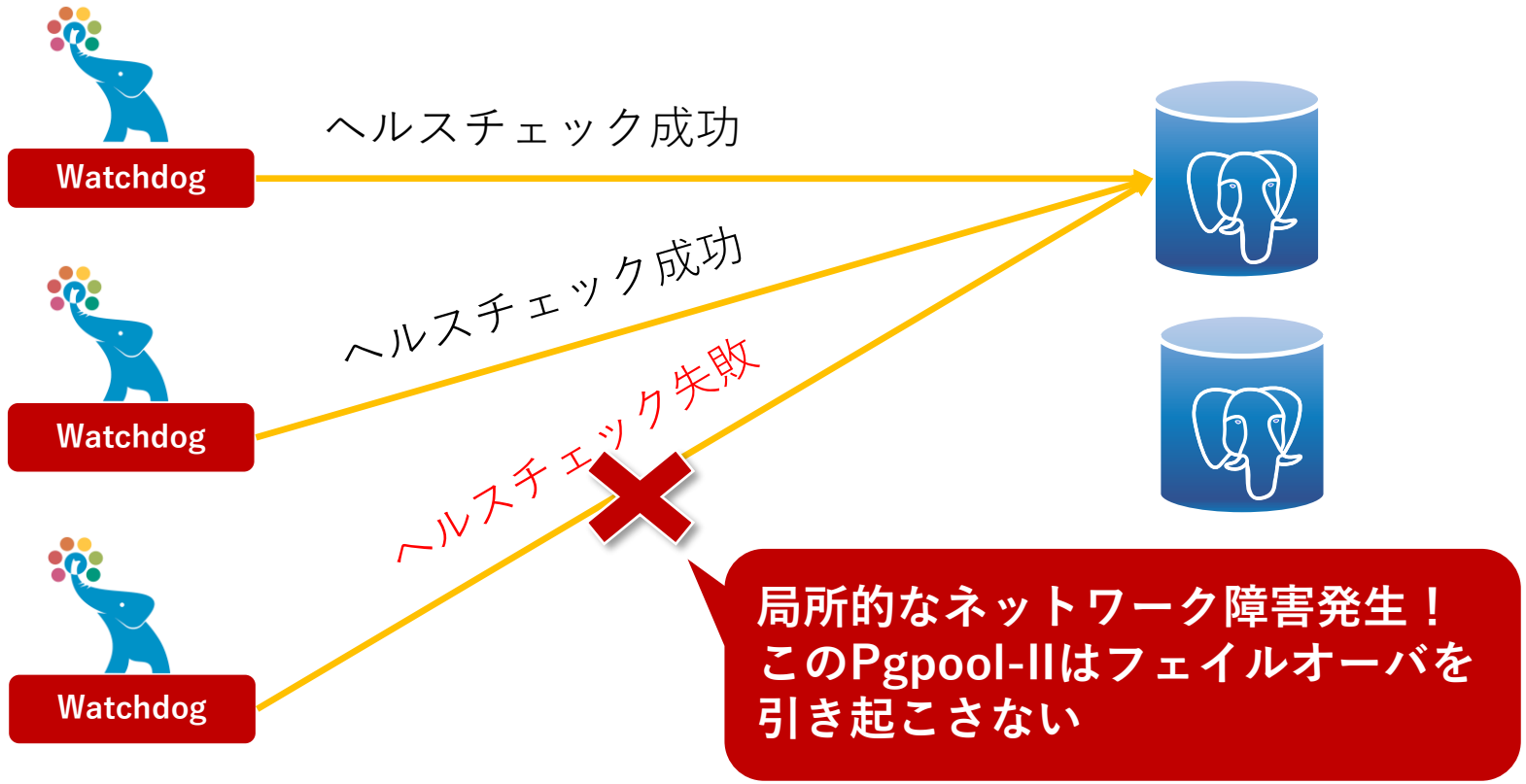
Watchdog – スプリットブレイン対策

- スプリットブレイン対策
- Pgpool-IIを奇数台(3台以上)用意することによって、多数決でどれが本当にダウンしているのかを正しく判断できる



Watchdog –フェイルオーバの制御

- ヘルスチェックでも多数決の原理を利用
- 局所的なネットワーク障害による障害誤検知を防止



負荷分散

負荷分散

- SQLパーサを搭載しており、クエリを解析できる
- 更新クエリをプライマリに送り、参照クエリを複数のPostgreSQLノード間で分散

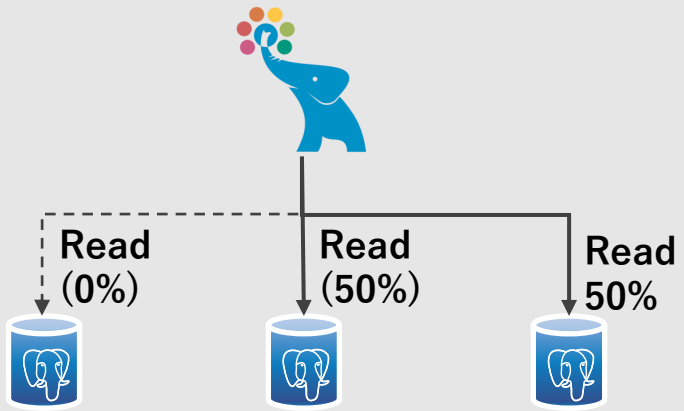
負荷分散モード

- セッションレベル (デフォルト)
- ステートメントレベル

負荷分散の比率が設定可能

例えば、プライマリを更新処理専用にし、すべての参照クエリをスタンバイに振り分けたい

```
backend_weight0 = 0
backend_weight1 = 1
backend_weight2 = 1
```



より柔軟な負荷分散設定

特定のクエリを負荷分散させたくない

primary_routing_query_pattern_listに指定されたSQLパターンを含むクエリは負荷分散されない

文字列で指定

```
primary_routing_query_pattern_list = '.*my_table*.'
```

先頭にコメントが付与されたクエリは負荷分散されない

コメントを付与

```
/*NO LOAD BALANCE*/SELECT * FROM t1
```

write_function_listに指定された関数は負荷分散されない

関数で指定

```
write_function_list = 'f1,f2'
```

その他の負荷分散の設定

database_redirect_preference_list

- データベース名によってクエリの振り分け先を決定

app_name_redirect_preference_list

- アプリケーション名によってクエリの振り分け先を決定

disable_load_balance_on_write

- 更新クエリが発行された後の負荷分散の振る舞いを設定

Pgpool-II 4.4新機能紹介

Pgpool-II 4.4の新機能ハイライト

性能向上

- アイドル状態のプロセスの数を動的に調整できるように
- クエリキャッシュの排他制御仕組みの変更

セキュリティ向上

- UNIXドメインソケット通信の強化
- 複数のIPアドレスをLISTEN可能に

設定がより柔軟に

- レプリケーション遅延を時間で指定
- 上位ネットワーク接続確認コマンドのカスタマイズ

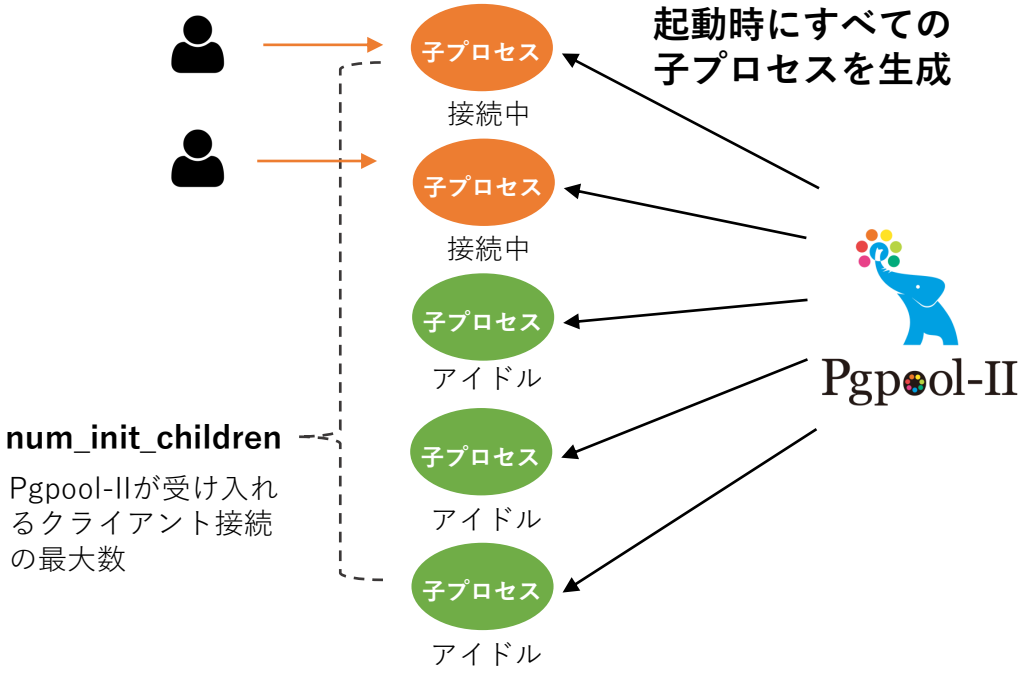
PostgreSQL 15の対応

- PostgreSQL 15パーサの移植

アイドルプロセスの管理

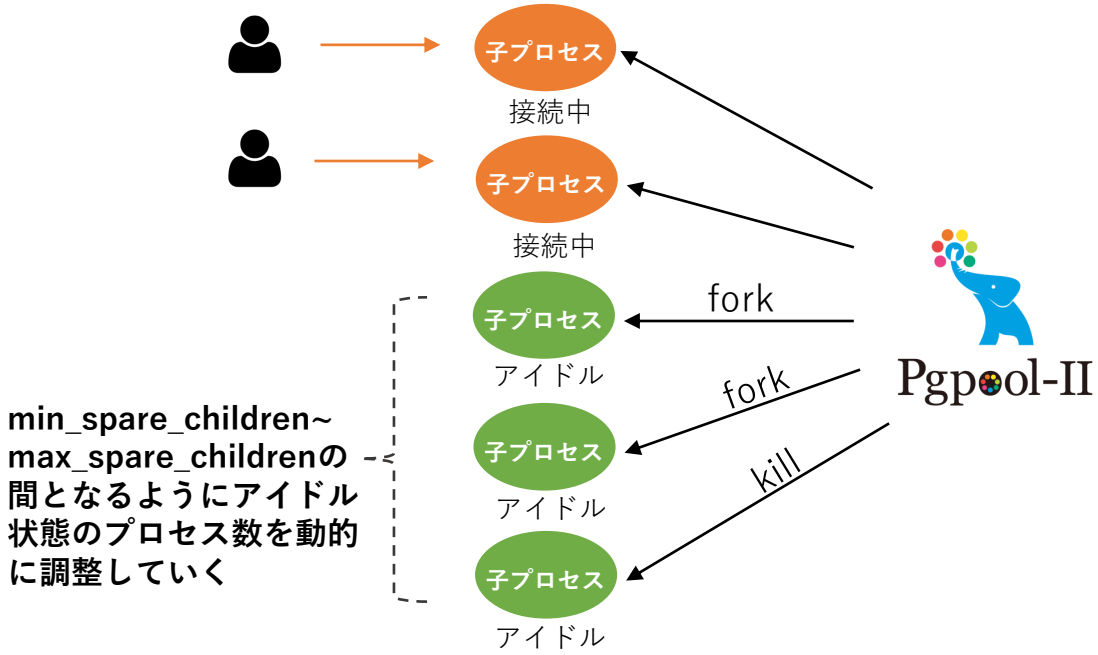
静的管理モード(デフォルト)

`process_management_mode = static`

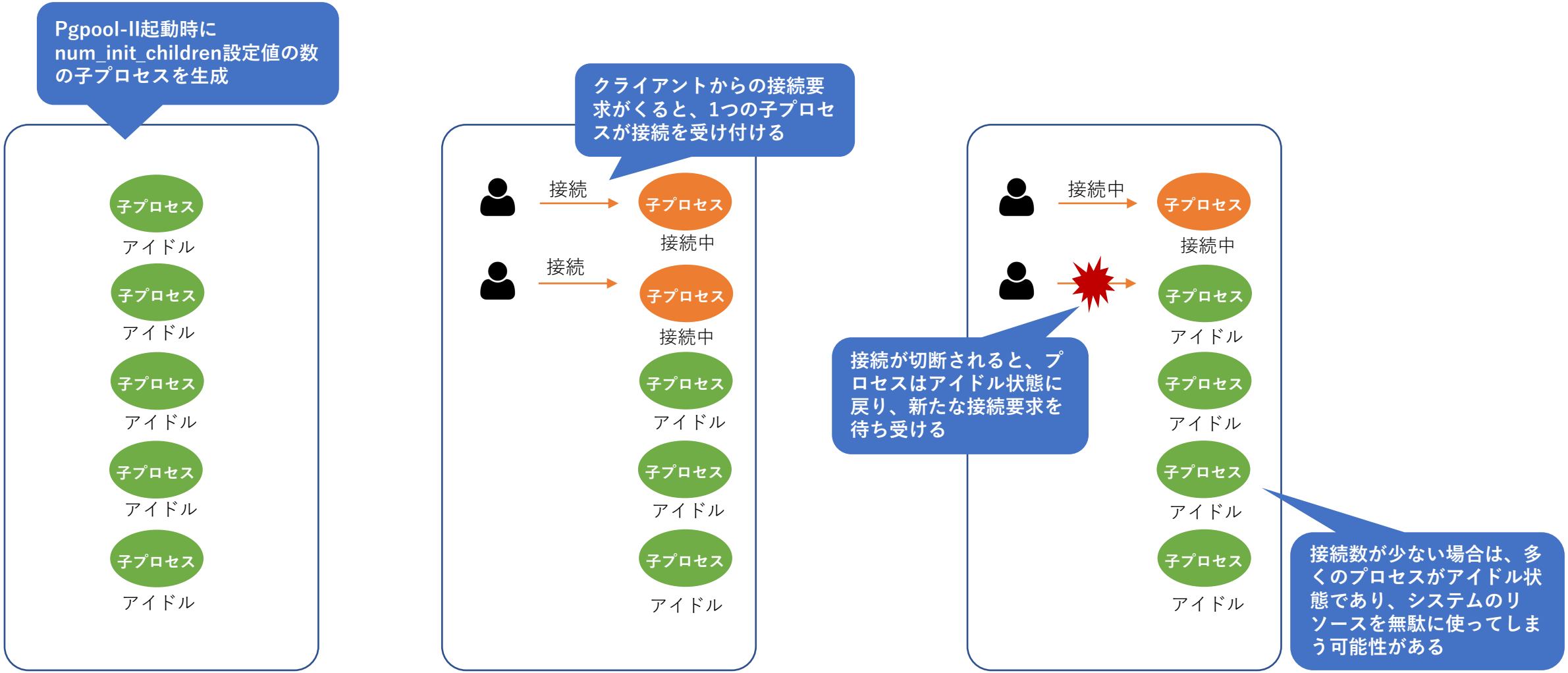


動的管理モード(4.4以降)

`process_management_mode = dynamic`



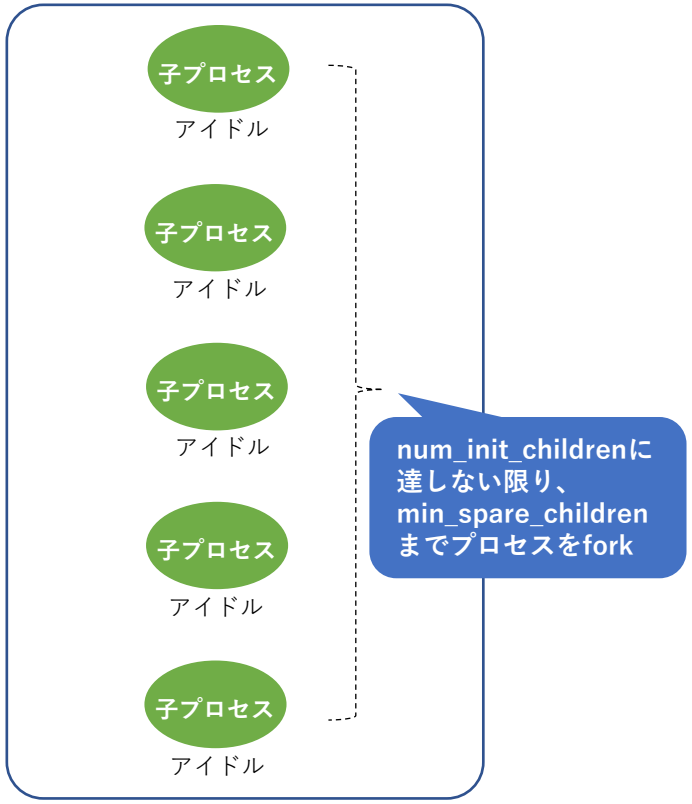
静的プロセス管理モード

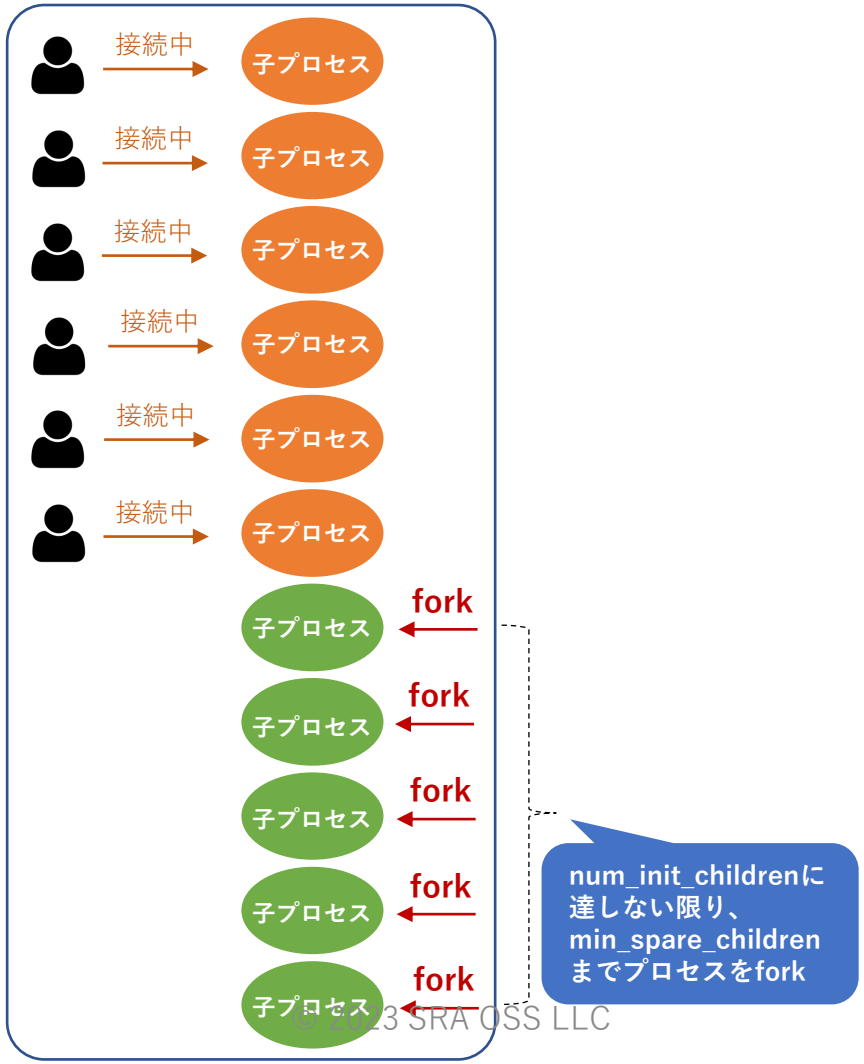


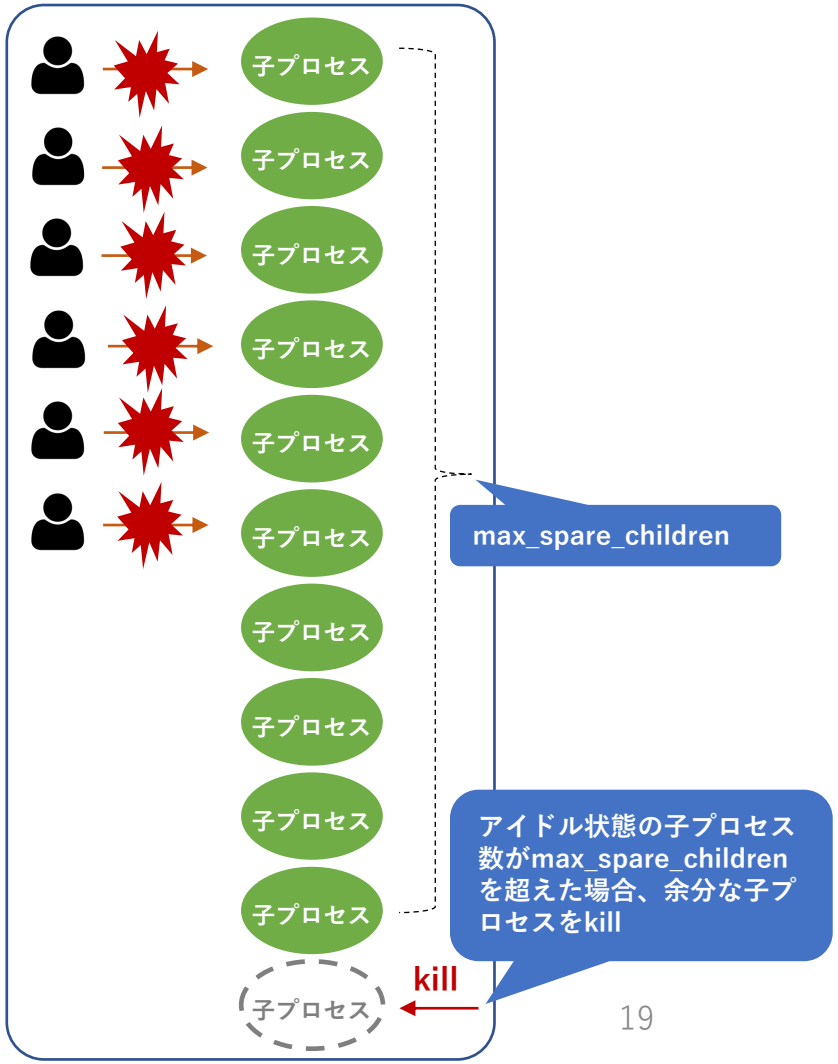
動的プロセス管理モード（4.4以降）

例）
```
min_spare_children = 5
max_spare_children = 10
```

min_spare_children~max_spare_childrenの間となるように
アイドル状態のプロセス数を動的に調整していく







余分なアイドルプロセスの調整

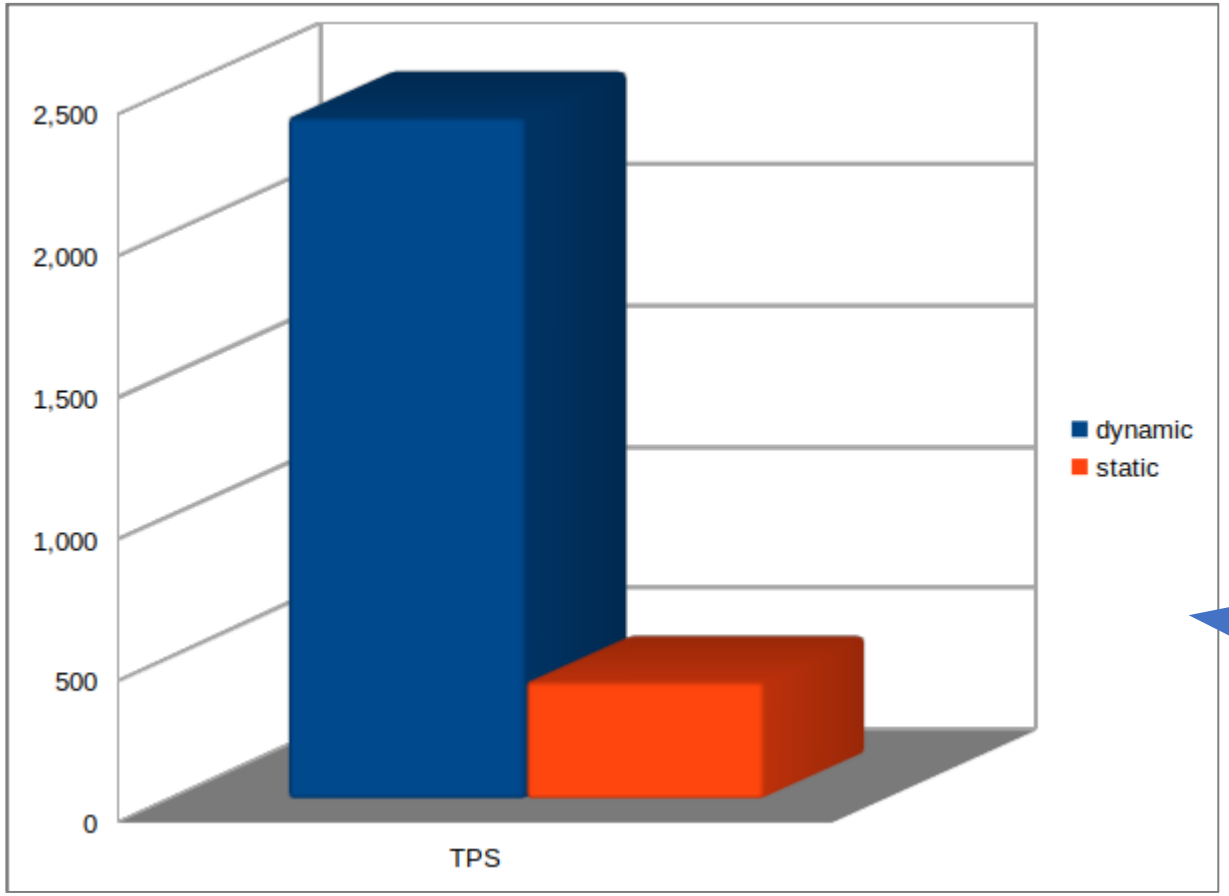
process_management_strategy

max_spare_childrenを満たすためのアイドル状態プロセスの調整戦略

指定可能な値

設定値	説明
lazy	余分なアイドル状態のプロセスが5分以上残っている場合にのみ終了される
gentle（デフォルト）	余分なアイドル状態のプロセスが2分以上残っている場合にのみ終了される
aggressive	余分なアイドル状態のプロセスの終了が積極的に実行される

ベンチマーク結果



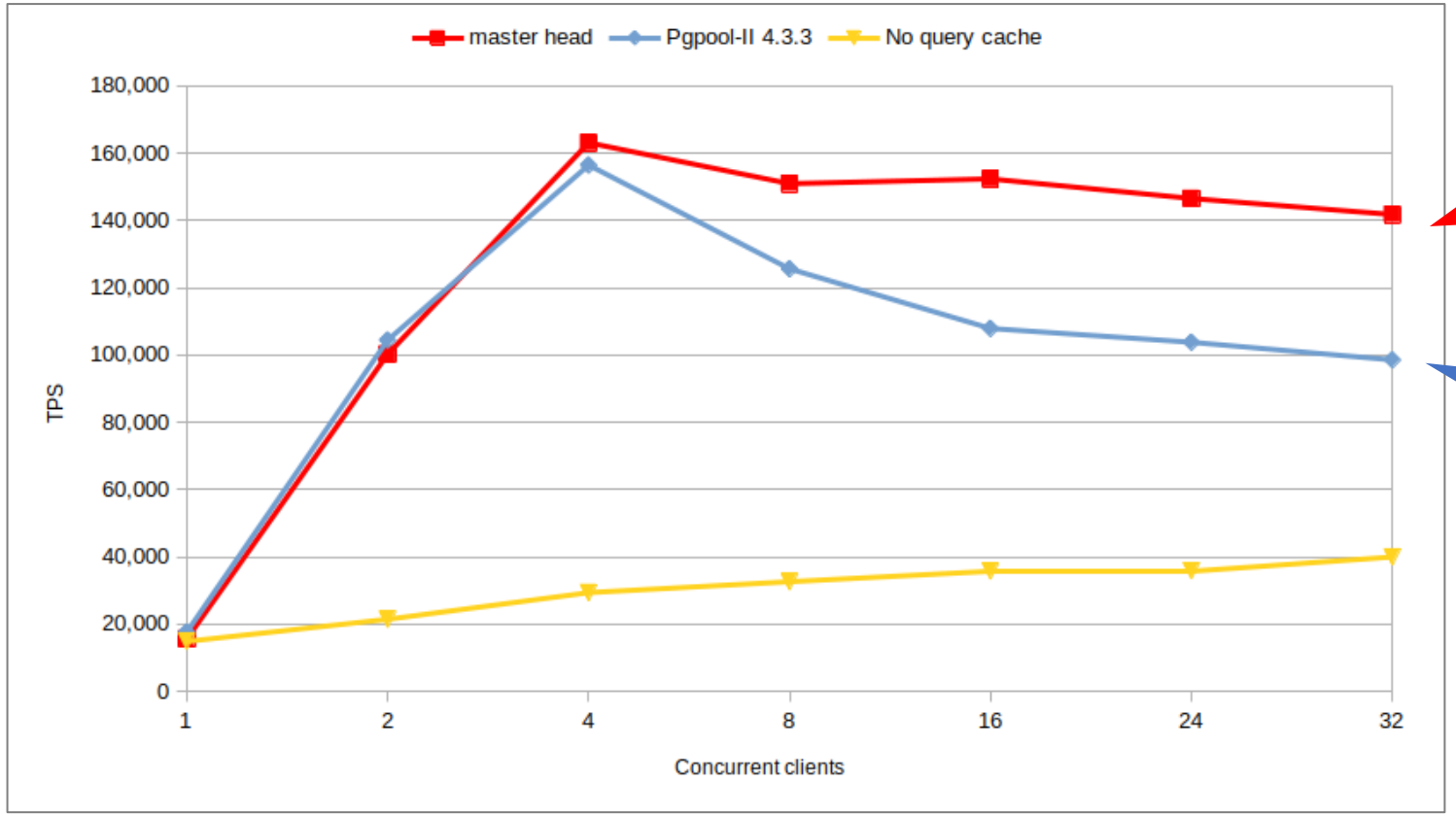

```
# 設定パラメータ
num_init_children = 900
max_connections = 1000

# 実行コマンド
pgbench -C -n -S -c 10 -T 30
```

num_init_childrenの数に対して同時接続数が少ない場合に、性能が改善される

クエリキャッシュの性能向上

- クエリキャッシュにおける排他ロックを共有ロックに置き換え
- 複数の接続が同時にキャッシュからデータを読み取り可能に



4.4
わずかな低下

4.3
TPSが低下

レプリケーション遅延を時間で指定

- レプリケーション遅延が閾値を超えた場合、負荷分散対象から外す
- レプリケーション遅延閾値の指定
  - バイト単位
    - delay_threshold
    - WAL位置 (LSN) のバイト差分
  - 時間単位 (秒) (**4.4以降**)
    - delay_threshold_by_time
    - pg_stat_replication.replay_lagから取得

delay_threshold_by_time が指定されていると、レプリケーション遅延は秒単位で表示される

```
test=# show pool_nodes;
 node_id | hostname  | port  | status | pg_status | lb_weight |  role   | pg_role | select_cnt | load_balance_node |  replication_delay  |
---------+-----------+-------+--------+-----------+-----------+---------+---------+------------+-------------------+---------------------+
 0       | localhost | 11002 | up     | up        | 0.500000  | primary | primary | 23         | true              | 0                   |
 1       | localhost | 11003 | up     | up        | 0.500000  | standby | standby | 20         | false             | 1088.488892 seconds |
(2 rows)
```

上位接続確認コマンドのカスタマイズ

上流サーバへの接続確認

- 定期的に上位サーバへの疎通確認を行うことでネットワークの接続性を監視
- 疎通先
  - trusted_serversによって設定
  - 複数のサーバを指定可能
- 疎通確認コマンド

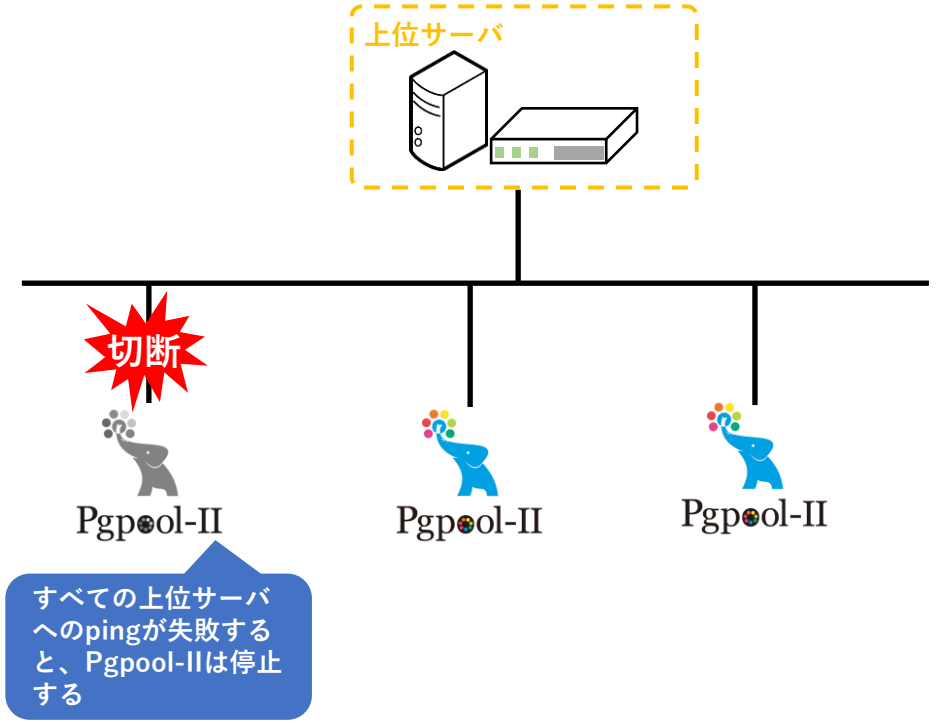
4.3以前 (ハードコーディングされた内容)

```
ping -q -c3 <疎通先サーバ>
```

4.4以降 (自由に設定可能、デフォルトではこれまで通り)

```
# 設定例
trusted_server_command = 'ping -q -c3 -W3 %h'
```

trusted_serversで指定された各サーバに置換される



UNIXドメインソケット通信の強化

- Pgpool-IIが複数のUnixドメインソケットを監視できるように
  - カンマ区切りで複数のディレクトリを指定できる

```
# 設定例

unix_socket_directories = '/tmp,/var/run/postgresql'
```

- Unixドメインソケット通信をより安全で柔軟に
  - OSユーザレベルでアクセス権限を制御
  - pg_hba.confの設定とは別のもの

```
# 例えば、postgresグループに所属するユーザしかUnixドメインソケット接続できないように設定

unix_socket_group = 'postgres'
unix_socket_permissions = 0770
```

複数のIPアドレスをLISTEN可能に

listen_addresses、pcp_listen_addressesでカンマ区切りの複数のリッスンIPアドレスを設定できるように

4.3以前

```
listen_addresses = '127.0.0.1'
OR
listen_addresses = '192.168.56.10'
OR
listen_addresses = '*'


pcp_listen_addresses = '127.0.0.1'
OR
pcp_listen_addresses = '192.168.56.10'
OR
pcp_listen_addresses = '*'
```

4.4以降

```
listen_addresses = '127.0.0.1,192.168.56.10'




pcp_listen_addresses = '127.0.0.1,192.168.56.10'
```

PostgreSQL 15パーサの移植

- Pgpool-IIは読み取りクエリを振り分けするために、PostgreSQLのSQLパーサを移植している
- Pgpool-II 4.4ではPostgreSQL 15のパーサを取り込んでいる

PostgreSQL 15パーサのおもな変更点

- MERGE文の追加
- COPY FROMにHEADER MATCHオプションの追加
- ALTER TABLEにSET ACCESS METHODアクションの追加
- など

参考情報

- Pgpool-II Wiki
  - https://pgpool.net/
- 日本語版ドキュメント
  - https://www.pgpool.net/docs/latest/ja/html/
- Pgpool-II 4.4 リリースノート
  - https://www.pgpool.net/docs/latest/ja/html/release-4-4-0.html

ご清聴ありがとうございました。