

Pgpool-II 4.7 新機能解説

2026/3/24
株式会社SRA OSS
越野 太基

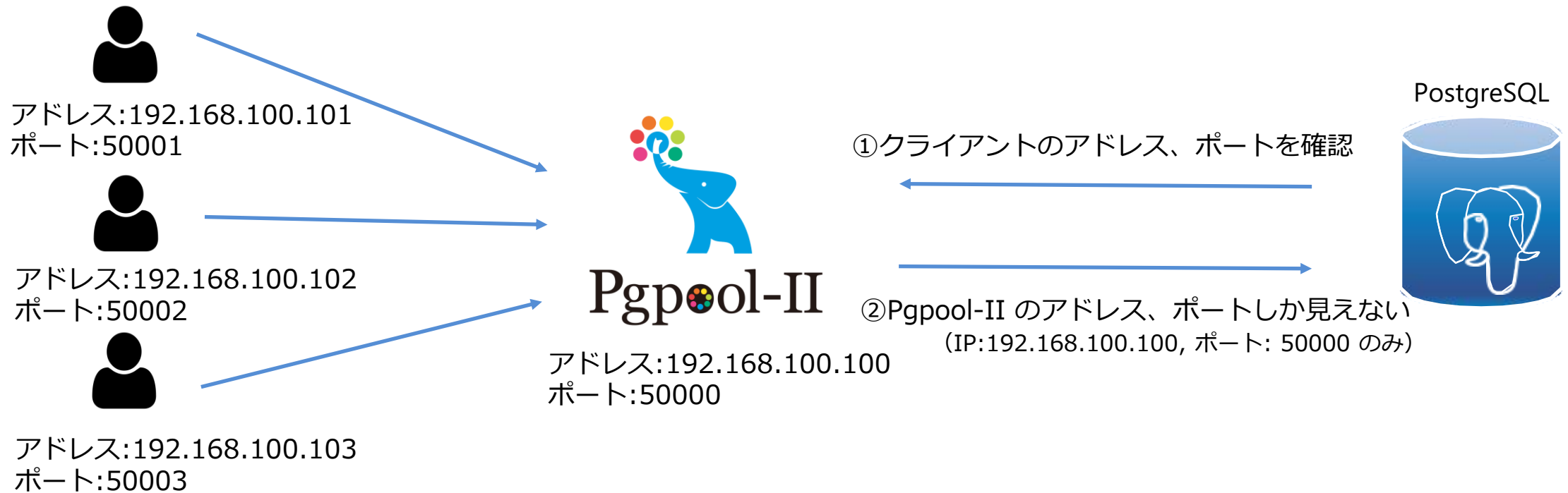
- 名前： 越野 太基
koshino@sraoss.co.jp
- 所属：株式会社SRA OSS
技術部 データベース技術グループ
- 職務：
 - PostgreSQL 技術サポート
 - PostgreSQL トレーニング講師
 - PostgreSQL クラスタ管理ツール Pgpool-II 開発者



- クライアント情報取得の機能強化
- ライフチェックログの改善
- Watchdog と heartbeat receiver のセキュリティ強化
- オンラインリカバリの機能強化
- フロントエンド / バックエンドプロトコルバージョン 3.2 サポートの追加
- PostgreSQL 18 の raw parser の移植



- Pgpool-II 経由で PostgreSQL に接続する
 - PostgreSQL からはクライアントが Pgpool-II に見える
 - PostgreSQL からは Pgpool-II のアドレス, ポートしか確認できなかった
 - 障害発生時にどのクライアントの接続の障害か特定できなかった

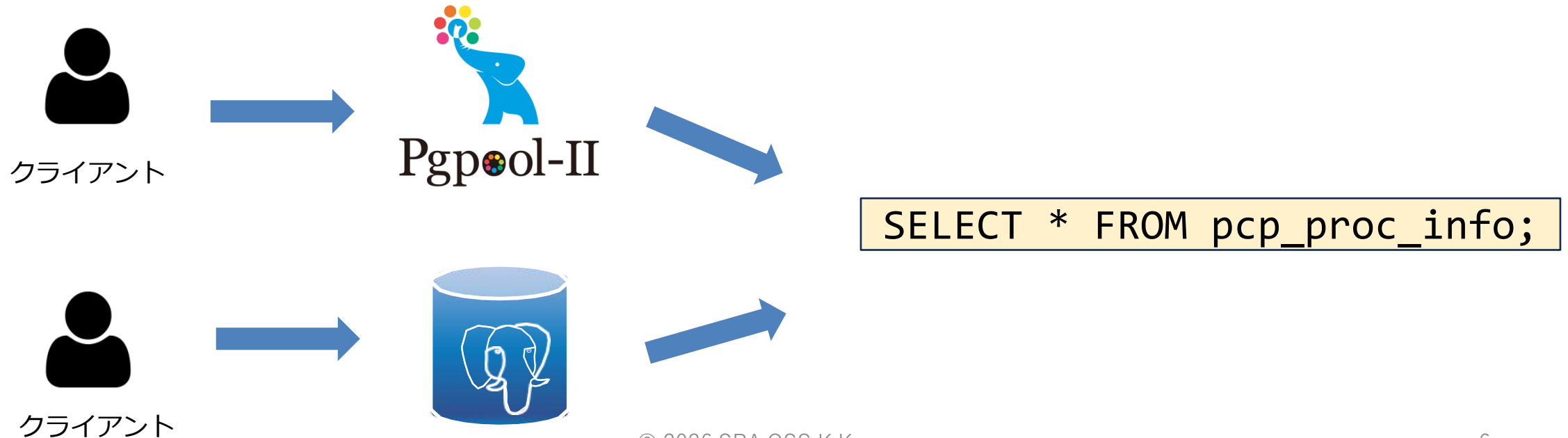


- pcp_proc_info, show pool_pools に DB 接続しているクライアントの情報を追加
 - クライアントホスト名(client_host)
 - クライアントポート名(client_port)
 - 最後に実行した SQL(statement)
- Pgpool-II 経由接続でも DB に接続しているクライアント情報が確認できるように
 - 障害発生時にどのクライアントでの障害か調査可能になった

```
$ pcp_proc_info -p 11001 --verbose
Database           : test
Username           : sraoss
...(中略)
client_host        : 192.168.100.101
client_port        : 50001
statement          : select 1;
```

SRA OSS クライアント情報取得の機能強化

- pgpool_adm エクステンション
 - pcp コマンドを PostgreSQL からでも呼び出せるようになるエクステンション
 - pcp コマンドは本来は Pgpool-II からしか呼び出せない
 - クライアントはどちらに接続しているか意識する必要がなくなる
 - SQL を利用でき WHERE 句で条件を指定して表示内容を絞りこむこともできる
 - pcp コマンドの場合は出力結果を整形するために OS のコマンドを併用する必要有



SRA OSS クライアント情報取得の機能強化

- pgpool_adm_pcp_proc_info の追加
 - pcp_proc_info 同様クライアント情報を表示できる
pgpool_adm 拡張機能
- 4.6 までは各クライアント情報を参照できなかった
- 4.7 で追加された pgpool_adm_pcp_proc_info は PostgreSQL のバックエンドプロセス ID からクライアント情報を参照可能
- WHERE 句で取得するレコードの条件指定も可能

```
test=# select * from pcp_proc_info(host => '', port => 11001, username => 'sraoss', password => 'sraoss')
      where connected = '1';
```

```
-[ RECORD 1 ]-----
```

```
database          | test
```

```
(中略)
```

```
connected         | 1
```

★ WHERE 句で指定したレコードを取得

```
client_host       | 192.168.100.101
```

```
client_port       | 50001
```

```
statement         | select 1;
```

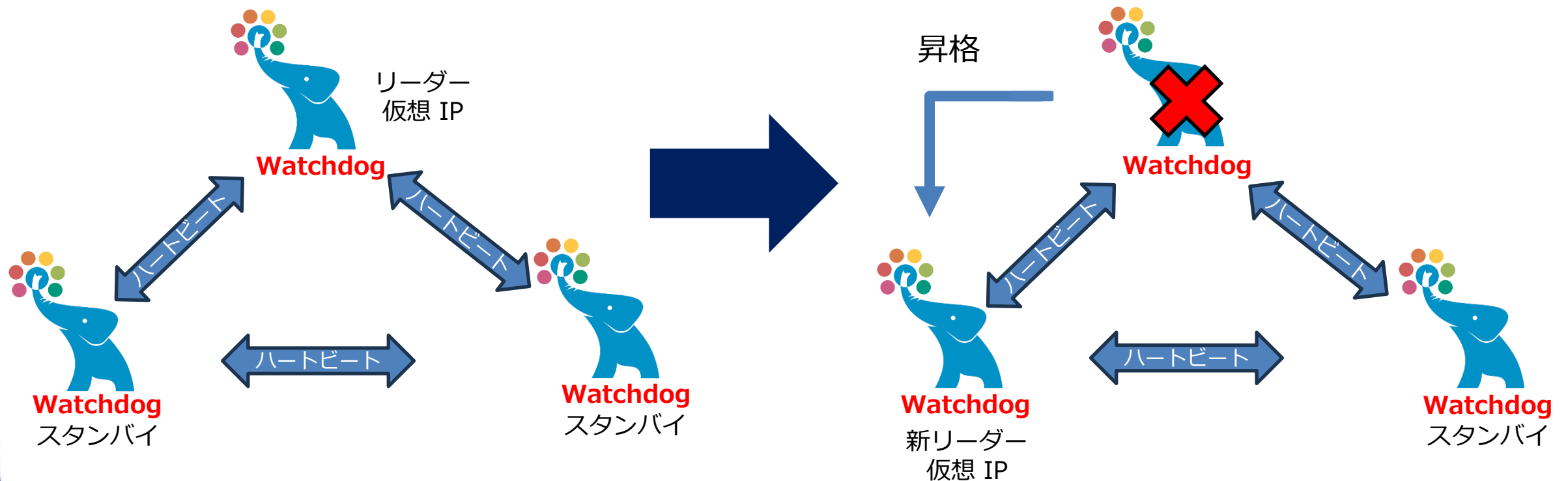
SRA OSS クライアント情報取得の機能強化

- pgpool_adm_proc_info と pg_stat_activity を結合する使用例
 - 各クライアントの詳細な接続情報を参照でき障害発生時の調査に役立つ
 - 例) pg_stat_activity の state を一緒に表示する SQL

```
test=# SELECT p.username, p.database, p.client_port, s.state, s.wait_event, s.query,  
             p.pid as pgpool_pid, s.pid as backend_pid  
             FROM pcg_proc_info(host => '', port => 11001,  
                               username => 'sraoss', password => 'sraoss') AS p  
             JOIN pg_stat_activity AS s ON s.pid::text = p.backend_pid;
```

username	database	client_host	client_port	state	query	pgpool_pid	backend_pid
sraoss	test	192.168.100.101	50001	active	select 1;	21467	21500

- Pgpool-II ノードの死活監視を行う仕組み
 - Pgpool-II 自体が単一障害点になることを防ぐ
 - heartbeat をお互いの Pgpool-II ノード間で送信し合う
 - リーダーノードがダウンしたら新しいリーダーに切り替える



- ライフチェック開始時に開始を知らせるログが出る

```
LOG: watchdog: lifecheck started
```

- 正常に動作しているかを確認するにはログを監視する必要があった
- ライフチェックがスタートしていないときにログが出ず
ライフチェックが正常でないことに気づきにくかった



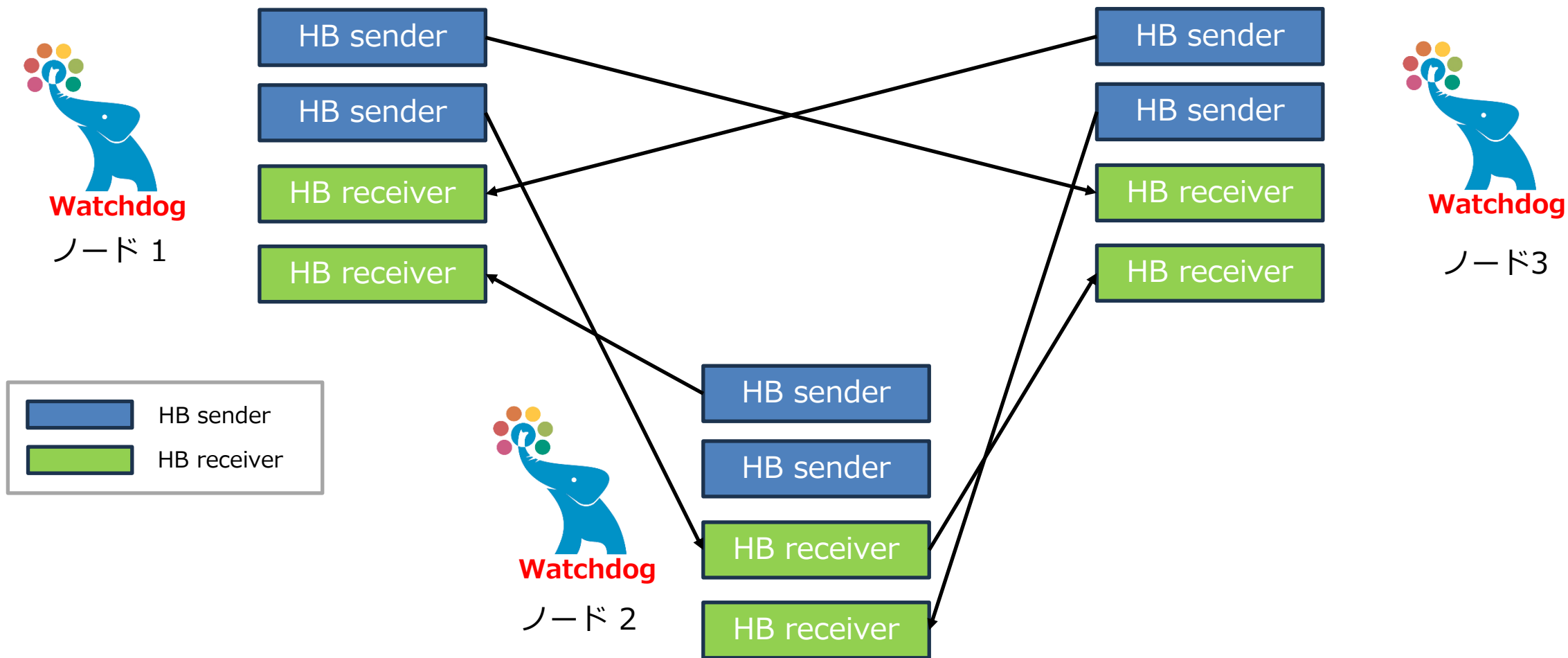
【 4.7 以降 】

- 開始までライフチェックが未開始であるログを出力
- wd_interval × 10 秒の間隔で出力

```
WARNING: watchdog: lifecheck has not started yet
```

SRA OSS watchdog の heartbeat receiver ソケット

- HB sender (heartbeat を受け取るソケット) と HB receiver (heartbeat を送るソケット) がノード数 × heartbeat_hostnameX で指定したホスト数分作成される



- 4.6 まではすべてのアドレスを LISTEN していた

tcp	0	0	0.0.0.0:9000	0.0.0.0:*	LISTEN	3583037/pgpool: watchdog
Tcp6	0	0	:::9000	:::*	LISTEN	3583037/pgpool: watchdog
Udp	0	0	0.0.0.0:9694	0.0.0.0:*		3583050/pgpool: heartbeat receiver
udp	0	0	0.0.0.0:9694	0.0.0.0:*		3583044/pgpool: heartbeat receiver
udp6	0	0	:::9694	:::*		3583044/pgpool: heartbeat receiver
udp6	0	0	:::9694	:::*		3583050/pgpool: heartbeat receiver

```
hostname0 = '192.168.100.50'  
wd_port0 = 9000  
heartbeat_hostname0 = '192.168.100.101'  
heartbeat_port0 = 9694  
  
hostname1 = '192.168.100.60'  
wd_port0 = 9000  
heartbeat_hostname1 = '192.168.100.102'  
heartbeat_port1 = 9694  
  
hostname2 = '192.168.100.70'  
wd_port0 = 9000  
heartbeat_hostname2 = '192.168.100.103'  
heartbeat_port2 = 9694
```

SRA OSS Watchdog と heartbeat receiver のセキュリティ強化

- 4.7 からは hostname, heartbeat_hostname で指定したアドレスでのみ LISTEN するようになった

tcp	0	0	192.168.100.50:9000	0.0.0.0:*	LISTEN	41636/pgpool: watchdog
udp	0	0	192.168.100.101:9694	0.0.0.0:*		41649/pgpool: heartbeat receiver
udp	0	0	192.168.100.101:9694	0.0.0.0:*		41650/pgpool: heartbeat receiver

```
hostname0 = '192.168.100.50'  
wd_port0 = 9000  
heartbeat_hostname0 = '192.168.100.101'  
heartbeat_port0 = 9694  
  
hostname1 = '192.168.100.60'  
wd_port1 = 9000  
heartbeat_hostname1 = '192.168.100.102'  
heartbeat_port1 = 9694  
  
hostname2 = '192.168.100.70'  
wd_port2 = 9000  
heartbeat_hostname2 = '192.168.100.103'  
heartbeat_port2 = 9694
```

SRA OSS Watchdog と heartbeat receiver のセキュリティ強化

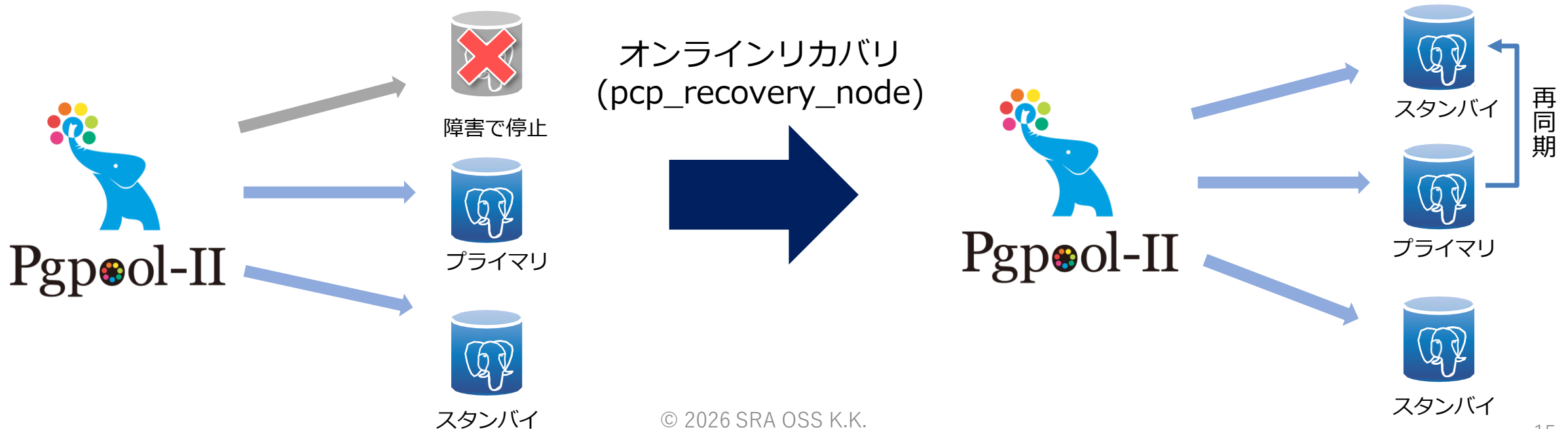
- 4.7.1 からは Watchdog と heartbeat それぞれで LISTEN するアドレス、ポートを指定可能に
 - watchdog の LISTEN ソケットを指定するパラメータ:
wd_listen_address, wd_listen_port
 - heartbeat receiver の LISTEN ソケットを指定するパラメータ:
wd_heartbeat_listen_addresses, wd_heartbeat_listen_port

tcp	0	0	192.168.100.100:50000	0.0.0.0:*	LISTEN	41636/pgpool: watchdog
udp	0	0	192.168.100.110:51000	0.0.0.0:*		41649/pgpool: heartbeat receiver
udp	0	0	192.168.100.110:51000	0.0.0.0:*		41650/pgpool: heartbeat receiver

```
hostname0 = '192.168.100.50'  
wd_port0 = 9000  
heartbeat_hostname0 = '192.168.100.101'  
heartbeat_port0 = 9694  
  
wd_listen_address = 192.168.100.100  
wd_listen_port = 50000  
wd_heartbeat_listen_addresses = 102.168.100.110  
wd_heartbeat_listen_port = 51000
```

SRA OSS オンラインリカバリの機能強化

- オンラインリカバリ
 - サービスを止めることなくデータベースノードを同期・復帰させる
 - 1. スタンバイノードで障害が発生
 - 2. `pcp_recovery_node` をノード番号を指定して実行
 - 3. スタンバイ側でプライマリのフルバックアップを取得
 - 4. スタンバイとして再起動



- オンラインリカバリでは PostgreSQL 上のデータベースを用いる
- 4.6 まではリカバリ用に用いるデータベースは template1 のみしか使用できなかった



【 4.7 以降 】

- 4.7 からは recovery_database でリカバリ用のデータベースを指定可能に
- デフォルトは postgres

```
#pgpool.conf の recovery_database を指定  
recovery_database = 'postgres'
```

SRA OSS .pcppass ファイルからの読み取りに失敗時の挙動改善

- .pcppass ファイル
 - pcp コマンドを実行する際の実行ユーザーのパスワードを保存しておくファイル
- 4.6 までは .pcppass ファイルが未作成またはパスワードファイルの権限設定が不適切の場合エラーで終了していた

```
FATAL: authentication failed for user "sraoss"  
DETAIL: username and/or password does not match
```

- 失敗後は .pcppass を作成するか -W オプションを付与して pcp コマンドを再実行・パスワードを入力する必要があった

```
pcp_node_info -p 9898 -W  
Password:
```

- 4.7 からは読み取り失敗後
 - W オプション付与時同様コマンドプロンプトにフォールバック

```
pcp_node_info -p 9898  
Password:
```

- -d, --debug オプションなしでもデバッグ情報も表示するように
 - 4.6 以前は -d, --debug オプションなしでは失敗の原因が出力されなかった
 - 読み取り失敗の原因を調査しやすくなった
 - 以下は .pcppass の読み取りに権限問題で失敗したときの例

```
WARNING: password file "~/.pcppass" has group or world access;  
permissions should be u=rw (0600) or less
```

- PostgreSQL 18
frontend/backend プロトコルのバージョン 3.0 → 3.2 へ
- BackendKeyData とクエリキャンセルメッセージの秘密鍵が可変長に



【 4.7 以降 】

- Pgpool-II も 3.2 に対応
- 3.0 バージョンでの接続も引き続き可能

```
# max_protocol_version オプションで 3.0 バージョンで接続  
psql "host=localhost dbname=test user=koshino max_protocol_version=3.0 port=11000"  
  
# 環境変数 PGMAXPROTOCOLVERSION に 3.2 を指定して接続  
PGMAXPROTOCOLVERSION="3.2" psql -p 11000 test
```

- 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 4.7 では **PostgreSQL 18** のパーサを取り込んでいる

PostgreSQL 18 パーサのおもな変更点

- 生成列を仮想列として使用可能
- DML クエリの RETURNING に OLD/NEW のサポートを追加
- 各種制約の追加(WITHOUT OVERLAPS, NOT ENFORCED)

SRA OSS logdir を work_dir に変更

- 4.6 までは pgpool の 管理ファイルを logdir 以下に格納
 - pgpool_status
 - memq_lock_file
- ログ関連のファイルを格納するディレクトリと誤解されやすかった



【 4.7 以降 】

- logdir -> work_dir に名称変更
- logdir も継続して使用可能
 - logdir を使用した場合は以下 WARNING が出て logdir で指定したディレクトリが使用される

```
WARNING: logdir is changed to work_dir
DETAIL:  if logdir is specified, the value will be set to work_dir
```

- Slony モードの廃止
 - backend_clustering_mode の一つである Slony モードが廃止
 - backend_clustering_mode :レプリケーション方式を指定できるパラメータ
 - Slony モードはデータのレプリケーションを Slony-I が担当するモード
 - 最終リリースが 3 年前
 - 現在利用ユーザーがほとんどいない
 - 廃止アナウンスに異論がなかったため 4.7 から廃止となった
- 4.7.1 からPgpool-II ライブラリ libpcp の名称を libpgpoolpcp に変更
 - RHEL の libpcp ライブラリとのコンフリクト回避のため libpgpoolpcp に名称変更

• 4.7 では以下の機能強化が行われた

- クライアント情報取得の機能強化
- ライフチェックログの改善
- Watchdog と heartbeat receiver のセキュリティ強化
- オンラインリカバリの機能強化
- フロントエンド / バックエンドプロトコルバージョン 3.2 サポートの追加
- .pcppass ファイルからの読み取りに失敗時の挙動改善
- PostgreSQL 18 の raw parser の移植
- logdir を work_dir に変更
- Slony-I の廃止
- Pgpool-II ライブラリの libpcp を libpgpoolpcp に名称変更

-  SRA OSS Tech Blog
 - <https://www.sraoss.co.jp/tech-blog/pgpool-ii/>
-  SRA OSS 公式 Youtube チャンネル
 - <https://youtube.com/playlist?list=PL87TvKgogUwDi6caA7HRGfPRWTBPK44dB&si=jdSJGC9wk9RKzGhD>



- Pgpool-II 4.7 リリースノート
 - <https://www.pgpool.net/docs/latest/en/html/release-4-7-0.html> (英語)
 - <https://www.pgpool.net/docs/latest/ja/html/release-4-7-0.html> (日本語)
- Pgpool-II wiki
 - https://pgpool.net/mediawiki/index.php/Main_Page
- Pgpool-II ドキュメント
 - <https://www.pgpool.net/docs/latest/en/html/> (英語)
 - <https://www.pgpool.net/docs/latest/ja/html/> (日本語)
- バグ報告
 - <https://github.com/pgpool/pgpool2/issues>
- ML
 - https://pgpool.net/mediawiki/index.php/Mailing_lists

ご清聴ありがとうございました。

