SRA OSS, INC.

pgpool-II 新バージョン 3.2 登場！

～多機能ミドルウェア pgpool-II の活用で PostgreSQL 利用の幅を広げる～

07 / 25 / 2012

SRA OSS, Inc. 日本支社
技術開発部　安齋 希美

Agenda

1. On Memory Query Cache
2. Watchdog

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

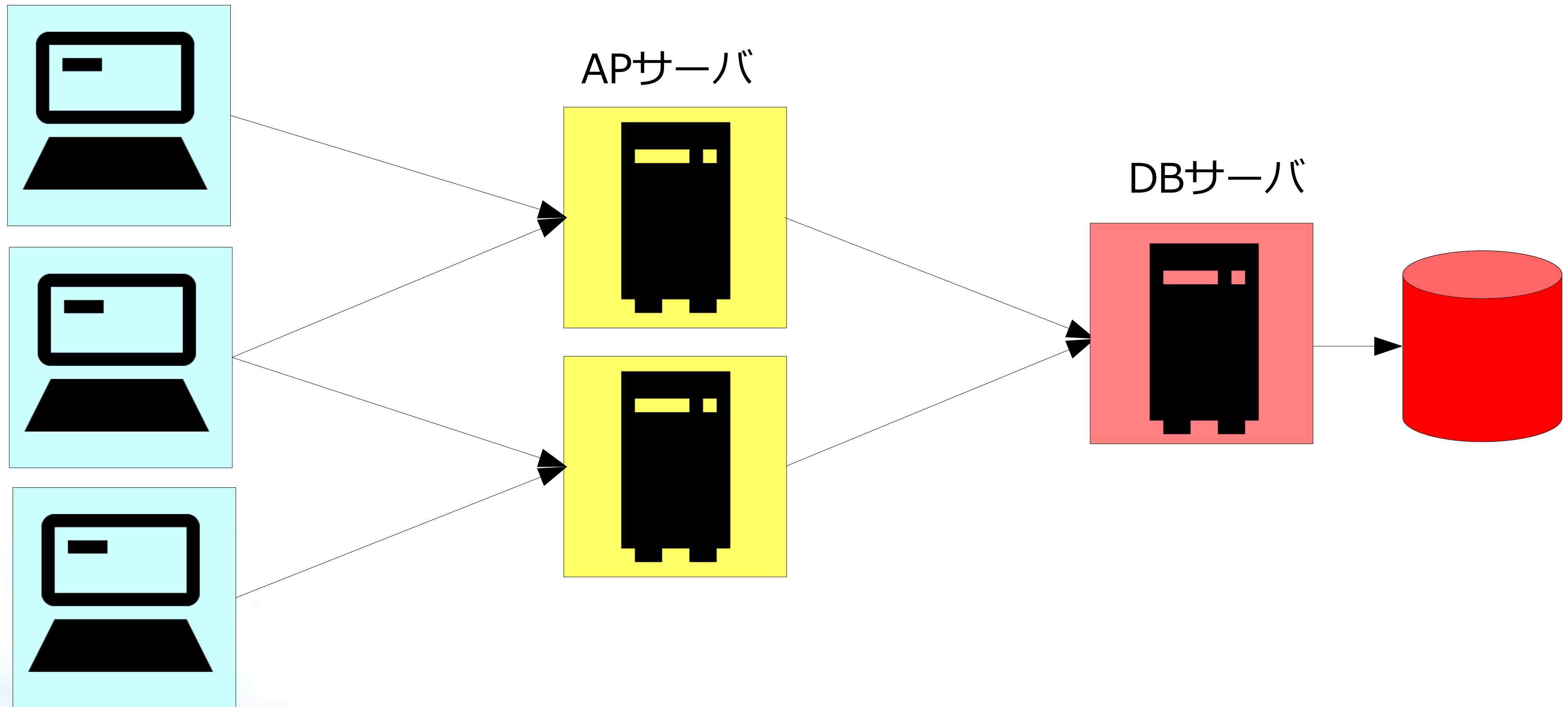
1. On Memory Query Cache

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

そもそも
なぜキャッシュ機能がほしいのか？

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

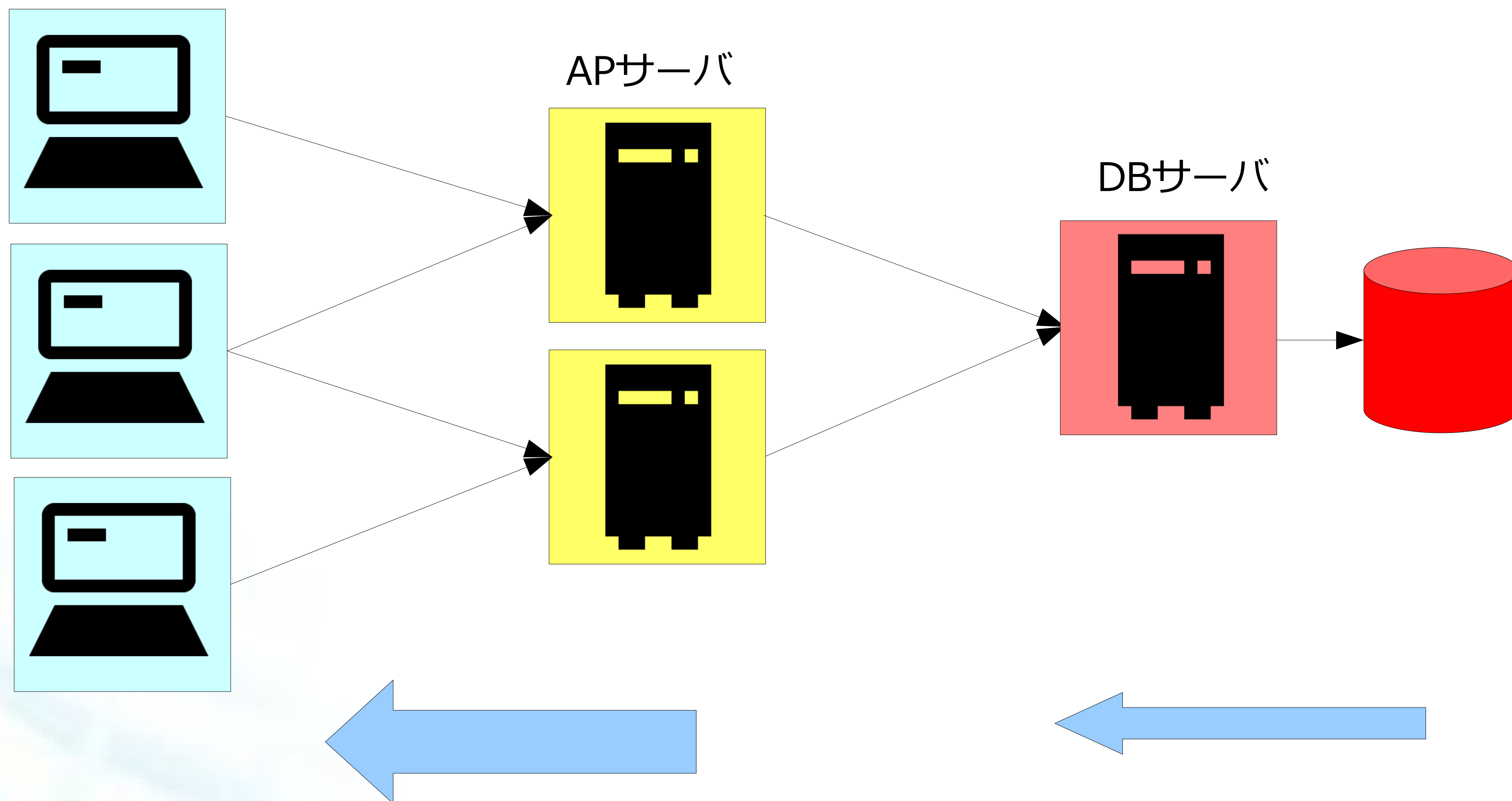
Web環境におけるレイヤー別負荷の違い



後ろのレイヤーほど負荷が高く、ボトルネックになりやすい

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

キャッシュを活用して負荷を軽減



APサーバで結果をキャッシュして返す　DBサーバで結果をキャッシュして返す

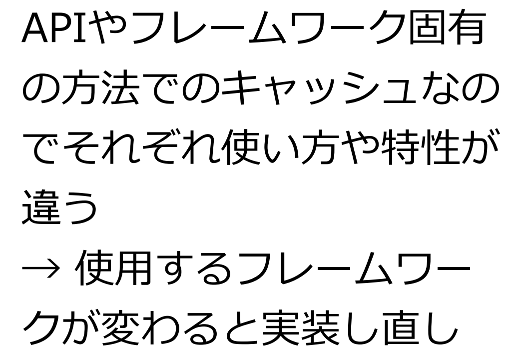
Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

キャッシュのさまざまな実装方法

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

キャッシュの実装例(1)

- アプリケーションサーバ、httpdサーバレベルでのキャッシュ実装
  - APC (Alternative PHP cache)
  - Apache2のmod_cache, mod_file_cache
  - RailsやHibernateなどのORマッパーでのキャッシュ
  - SquidやVarnishなどのリバースプロキシを使う
  - memcachedなどのKVS(Key Value Store)を使ったキャッシュ



Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

キャッシュの実装例(2)

- DBMSでのキャッシュ
  - MySQLの実装が有名。DB2にもある？
  - キャッシュにアクセスが高速化するが、DBMSへのアクセスがなくなるわけではないので、DBMSがボトルネックになることもある
- MySQLのクエリキャッシュ
  - オンメモリキャッシュ
  - クエリ文字列が一致したらキャッシュヒット
  - Prepared queryには対応していない
  - テーブルが更新されたら該当キャッシュは全クリアされる
  - MySQLを再起動したらキャッシュの内容はなくなる

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

APサーバとDBサーバのキャッシュの比較

	APサーバでキャッシュ	DBサーバでキャッシュ
キャッシュの効果	◎	○
DBサーバへの負荷	○	△
アプリケーション透過性	×	○
スケーラビリティ	○	×

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

pgpool-IIにもクエリキャッシュはありました…

- 旧クエリキャッシュ機能は廃止
  - キャッシュストレージがDBなので遅い
  - DBを更新しても自動ではキャッシュが更新されない
  - 拡張問い合わせに対応していない
- 新機能として実装
  - Google Summer of Codeとしてプロトタイプを実装
  - 開発グループで機能追加、改良
  - 拡張問い合わせにも対応
  - キャッシュストレージとして、共有メモリ or memcached を選択できる

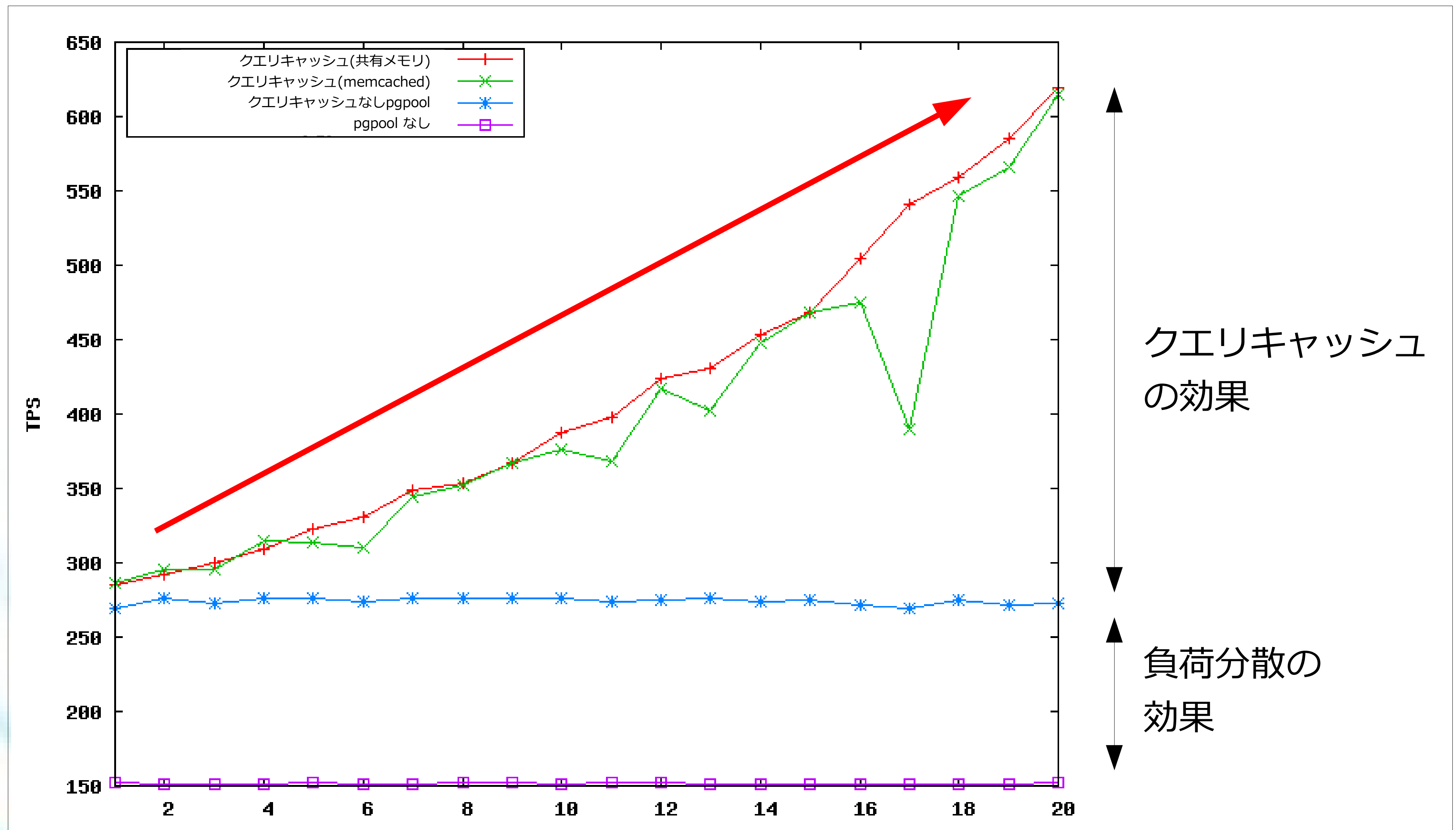
Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS,INC.

新しいクエリキャッシュ！

on memory query cache

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

クエリキャッシュの効果



Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

SRA OSS,INC.

どういうときならキャッシュヒットとみなす？

- **クエリ文字列 + データベース名 + ユーザ名**

  これをmd5ハッシュしたものをキーにして一致を判断し、ハッシュ値が一致したらキャッシュがヒットしたものと見なす

- **なるべく小さく！**

  直接SQL文を比較しないのは、長大なSQL文を保存する必要性を避けるため

- **セキュリティ！**

  ユーザ名がキーに含まれているのは、参照権限がないテーブルのデータをキャッシュを経由して他のユーザが参照できないようにするため

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

キャッシュの更新/無効化

- キャッシュ対象のテーブルが一部でも更新されたら、そのテーブルを参照している**キャッシュをすべて自動的に削除**する
  - 更新クエリ：INSERT/UPDATE/DELETE/TRUNCATE
  - そのために、キャッシュを登録する際に参照しているテーブルのOIDを調べ、**OIDマップ** というファイルに登録しておく（検索インデックスのイメージ）
  - 更新クエリが実行されたらOIDマップを調べて、関連するキャッシュを削除する
- データベースやテーブルが削除された場合も同様

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

クエリキャッシュのしくみ

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

クエリ実行前の処理： SimpleQuery()

1. 実行しようとしているクエリは SELECT & 更新のあったトランザクションでない？
   - md5ハッシュを生成し、キャッシュがないか見てみる
   - あったらキャッシュにあるクエリの結果を返す
2. 実行しようとしているクエリはDROP DATABASE？

   → 実行前に対象データベースの OID を取得しておく
3. 実行しようとしているクエリはSELECT？
   - ホワイトリスト等のチェック後、キャッシュOKフラグをたてる
   - SELECT 内のテーブルの OID を取得する

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

クエリ実行後の処理： ReadyForQuery()

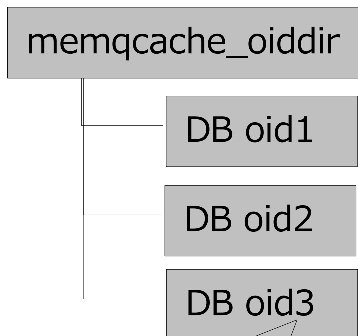
1. キャッシュが **memqcache_maxcache** より小さい？
2. キャッシュ可フラグがたっている？ → キャッシュを登録する
   - md5ハッシュからキャッシュIDを検索し、すでに存在しないかチェックする
   - 共有メモリ or memcached にキャッシュを登録する
   - **memqcache_oiddir/DB の OID/テーブルのOID** のファイルにキャッシュ ID を登録する
3. DROP DATABASE だったら **memqcache_oiddir/DB の OID** ディレクトリを削除する

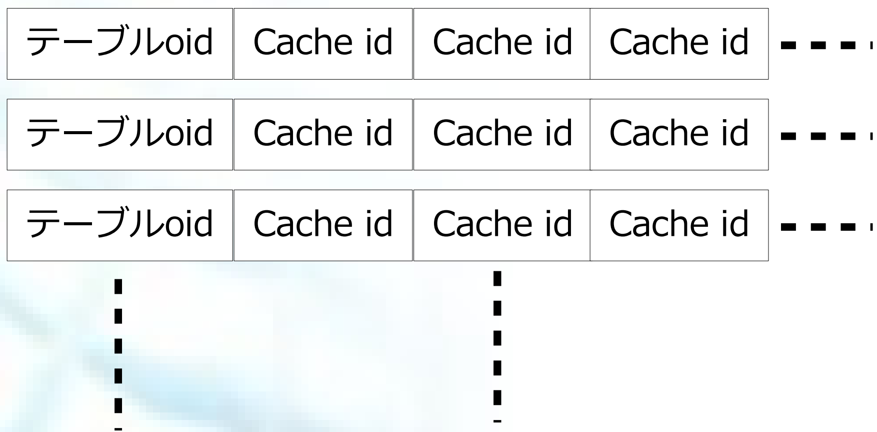
Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

SRA OSS,INC.

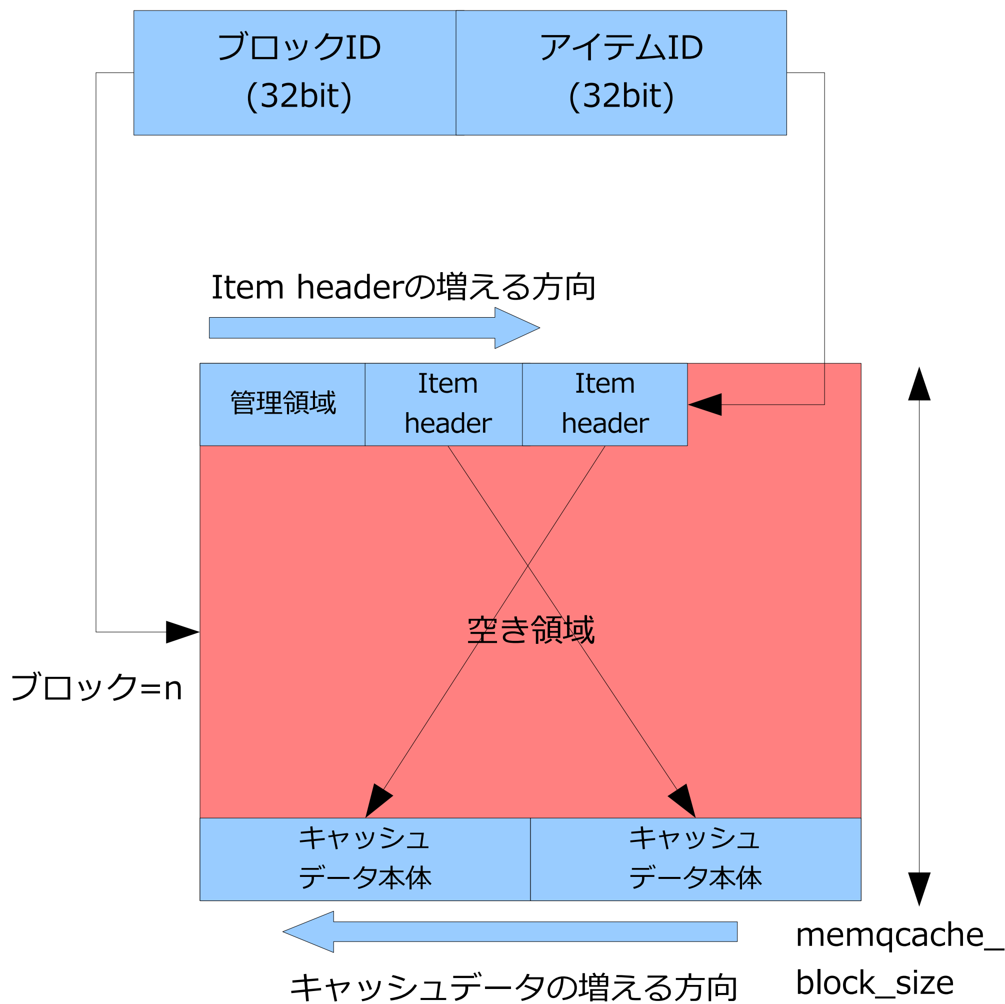
クエリキャッシュのファイル構造

OIDマップ





共有メモリ上のキャッシュ



Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

SRA OSS, INC.

ログから挙動を見てみる（SELECT初回）

```
LOG:    statement: SELECT count(*) FROM pgbench_ac
DEBUG: encode_key: `anzaiSELECT count(*) FROM pgbench_accounts;pgbench'
                   -> `9e4771a75f554a375bb3da69c4e79a2f'
DEBUG: pool_fetch_cache: search key
                         ==9e4771a75f554a375bb3da6
DEBUG: pool_fetch_cache: cache not found on shmem
```

クエリからハッシュキー生成

キャッシュにないか探す
→ なし

```
DEBUG: pool_is_allow_to_cache:check table_names[0] = pgbench_accounts
DEBUG: pattern_compare: white_memqcache_table_list (^pgbench_accounts$)
                        matched: pgbench_accounts
```

ホワイトリストをチェック → キャッシュOK

```
LOG:    DB node id: 0 backend pid: 4055
        statement: SELECT count(*) FROM pgbench_accounts;
DEBUG: ReadyForQuery: transaction state: I
```

クエリ実行

```
DEBUG: pool_add_item_shmem_cache:
       new item inserted. blockid: 0 itemid:0
DEBUG: pool_add_item_shmem_cache: block: 0 item: 0
DEBUG: pool_commit_cache: blockid: 0 itemid: 0
```

共有メモリにキャッシュ登録

```
DEBUG: pool_add_table_oid_map: dboid 366540
```

OIDマップに cache ID 登録

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

SRA OSS,INC.

ログから挙動を見てみる（SELECT 2回め～）

```
LOG:    statement: SELECT count(*) FROM pgbench_ac
DEBUG: encode_key: `anzaiSELECT count(*) FROM pgbench_accounts;pgbench'
                   -> `9e4771a75f554a375bb3da69c4e79a2f'
LOG:    query result fetched from cache.
        statement: SELECT count(*) FROM pgbench_accounts;
DEBUG: pool_fetch_from_memory_cache:
       a query result found in the query cache,
       SELECT count(*) FROM pgbench_accounts;
```

クエリからハッシュキー生成

キャッシュにないか探す
→ あり
→ 結果をキャッシュから返す

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

SRA OSS, INC.

キャッシュされないSELECT (1)

- current_timestamp など、Immutableでない（実行するたびに結果が異なる）関数への呼び出しを含むSELECT
- SELECT INTO, SELECT FOR UPDATE, SELECT FOR SHARE
- SELECTの結果データが大きいもの（memqcache_maxcache）
- 成功しなかったSELECT
- ロールバックされたトランザクション内のSELECT結果
- UnloggedテーブルをふくんでいるSELECT

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

キャッシュされないSELECT (2)

- check_temp_table が on のとき、一時テーブルはキャッシュしない
- ブラックリスト（black_memqcache_table_list）に指定された名前のテーブル

  あるいはホワイトリスト（white_memqcache_table_list）に指定されていないテーブル

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

注意事項

- キャッシュヒット率が低い場合はクエリキャッシュを使わない方が速い

（使うからには50～70%はほしい）

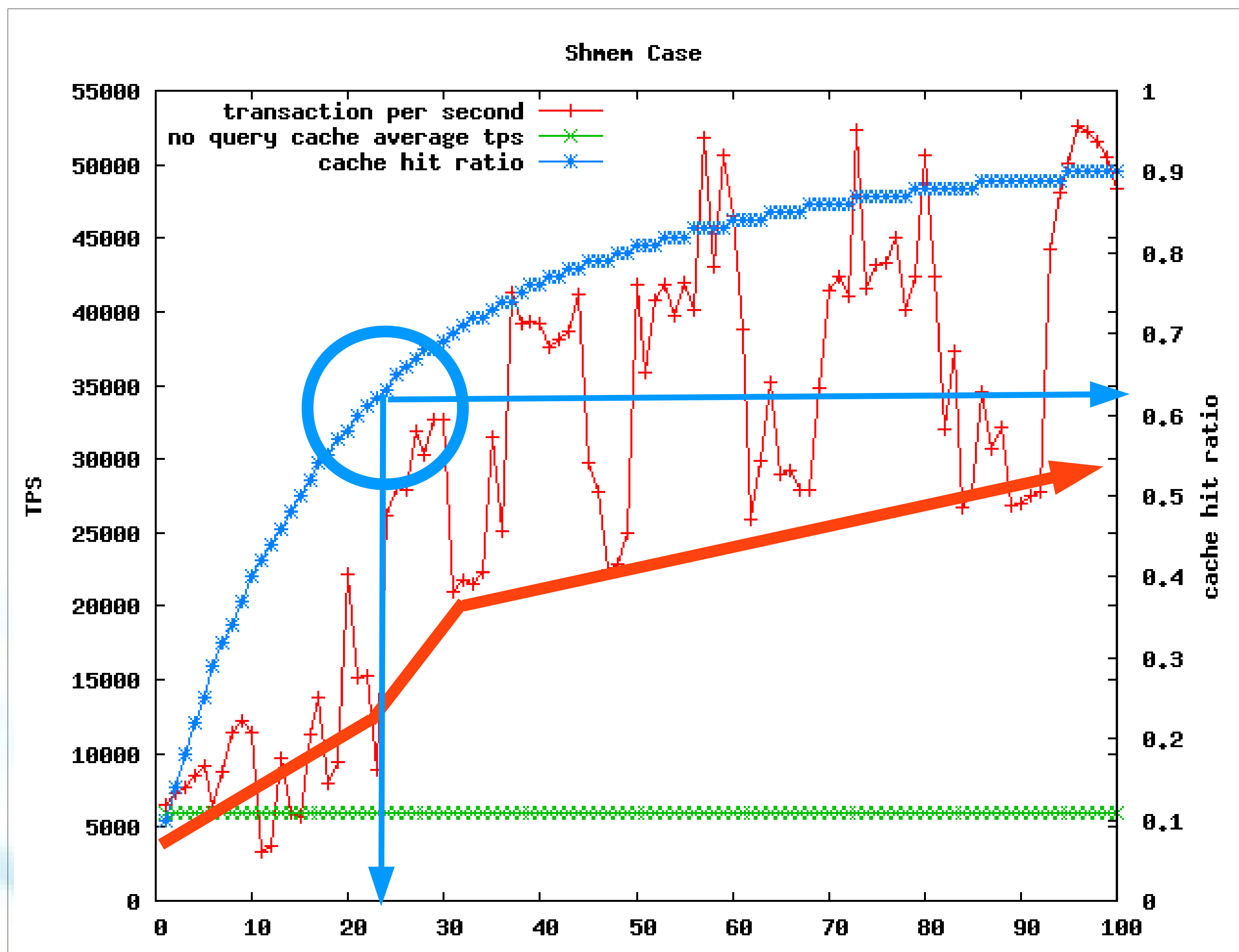
ベンチマーク

キャッシュなし：平均5979.27 TPS → 常にこの数字を上回るキャッシュヒット率は？

- pgpool.conf のクエリキャッシュ関係のパラメータはデフォルト
- pgbench -S (scale factor = 1)
- pgbench -S -t 10000を100回 → show pool_cacheでキャッシュヒット率を取得
- CORE i5 2.6 GHz dual core, メモリ 8GB、SSD、Linux kernel 2.6.35
  - memcached も同じPCで動かしている

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS,INC.

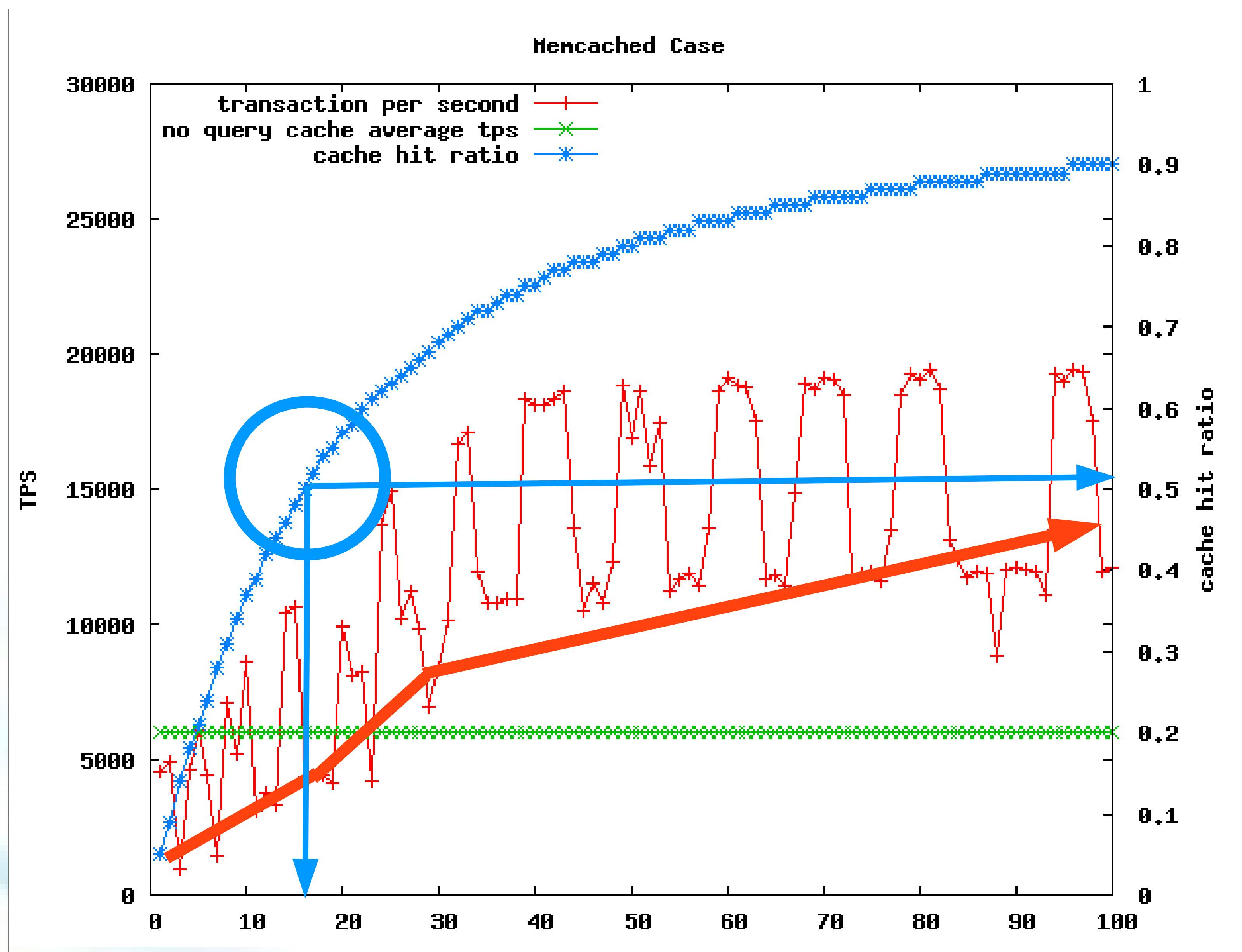
共有メモリのキャッシュヒット率



Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

SRA OSS, INC.

memcachedのキャッシュヒット率



Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

SRA OSS, INC.

制限事項

- **暗黙的な更新が認識できない**
  - VIEWもキャッシュされるが、VIEWが参照しているテーブルが更新されてもキャッシュは有効
  - トリガによって暗黙的に更新されるテーブルが認識できない
  - 外部キーが指定されていて、ON DELETE CASCADEなどで他のテーブルの行が暗黙的に更新されたことが認識できない
- **スキーマを認識できない**
  - スキーマが異なっていても、DB名、テーブル名が同じならば同じテーブルと見なされる

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS,INC.

クエリキャッシュ関連のパラメータ (1)

- 全般
  - memory_cache_enabled（クエリキャッシュ有効）
  - memqcache_method（キャッシュストレージ選択）
- キャッシュの制御など
  - memqcache_expire （キャッシュの寿命秒）
  - memqcache_auto_cache_invalidation（自動削除有効）
  - memqcache_maxcache（SELECT結果の最大サイズ）
  - white_memqcache_table_list, black_memqcache_table_list（キャッシュ対象・非対象テーブル名リスト）

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

クエリキャッシュ関連のパラメータ (2)

- 共有メモリ関連
  - memqcache_total_size（共有メモリ使用最大サイズ）
  - memqcache_max_num_cache（キャッシュ最大数）
  - memqcache_cache_block_size（共有メモリの分割ブロックサイズ）
- memcached 関連
  - memqcache_memcached_host（memcachedのホスト名）
  - memqcache_memcached_port（memcachedのポート番号）

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

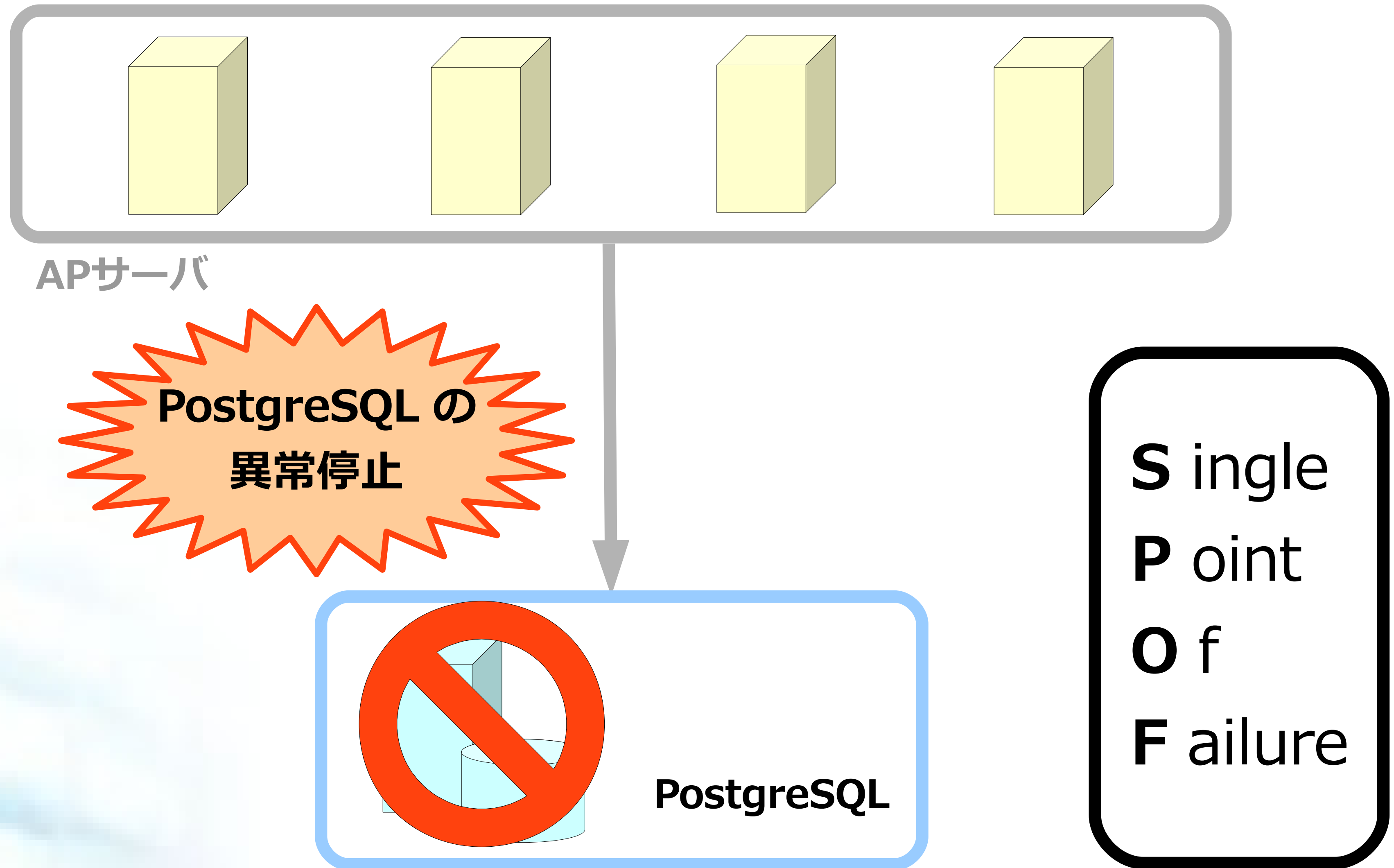
2. Watchdog

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

High Availability とは?

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

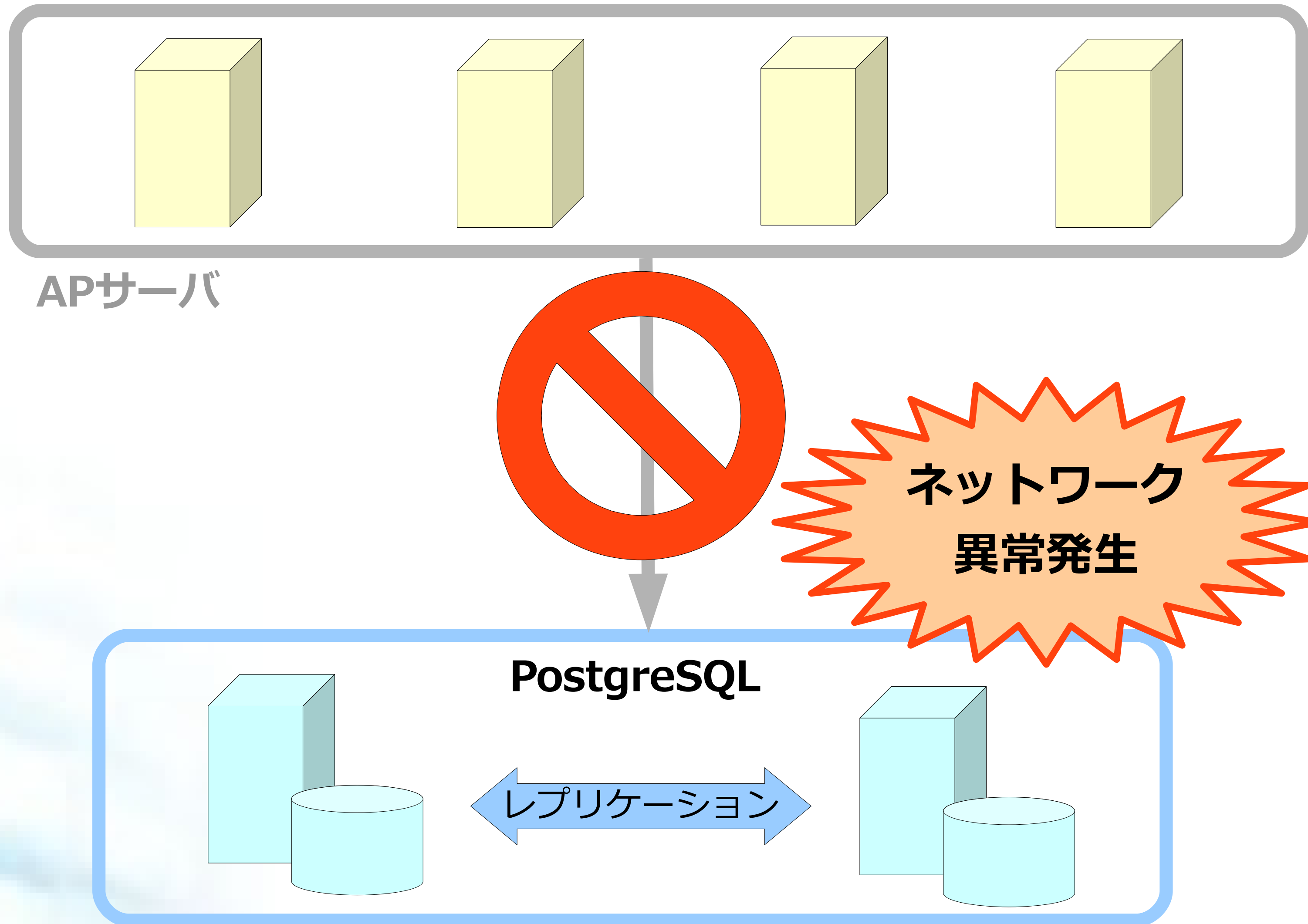
1台のPostgreSQL



Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

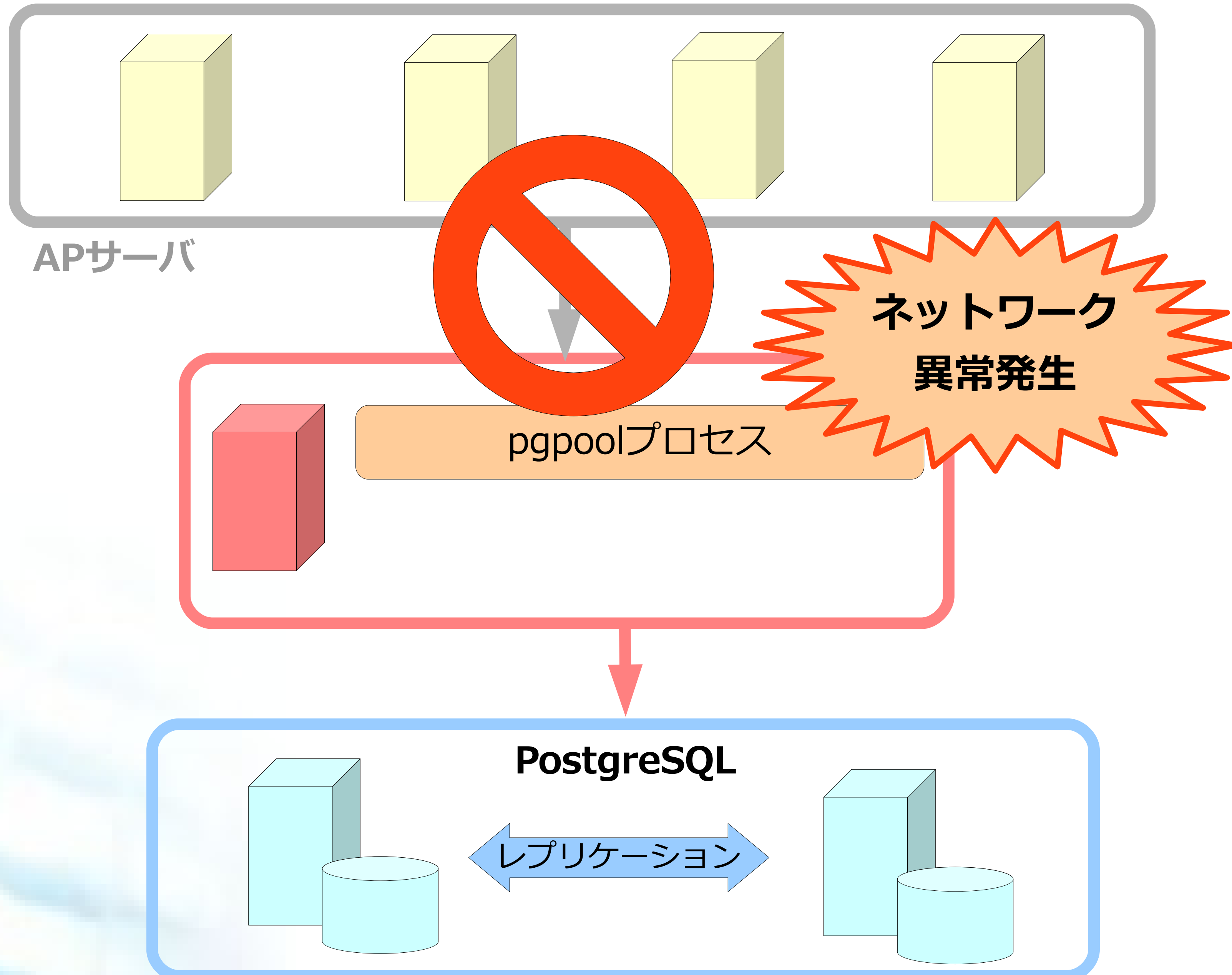
SRA OSS, INC.

レプリケーションしているPostgreSQL



Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

レプリケーション+pgpoolでの振り分け

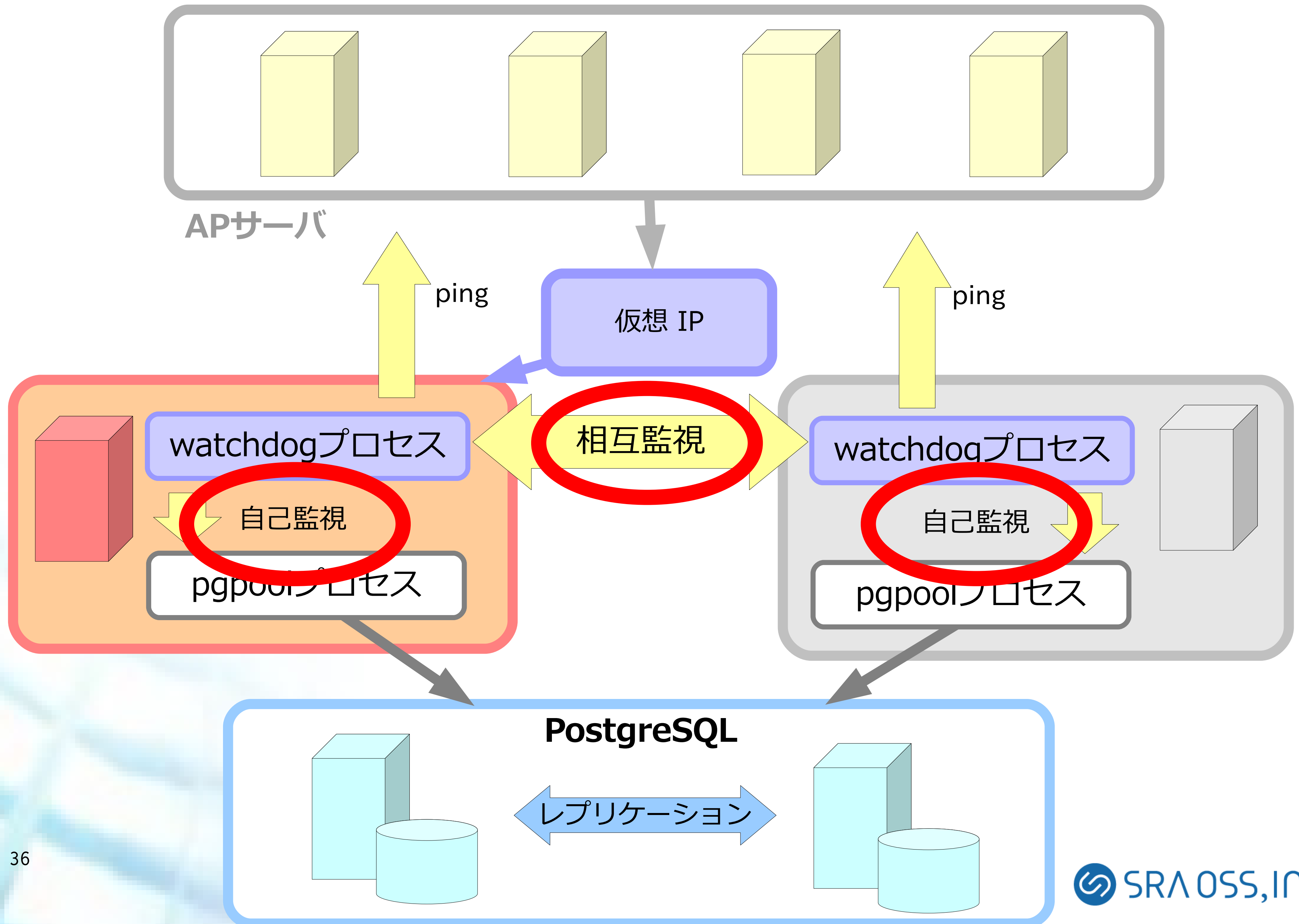


Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS,INC.

watchdogを使った Active/Standby 構成

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

watchdogを使ったActive / Standby構成



SRA OSS, INC.

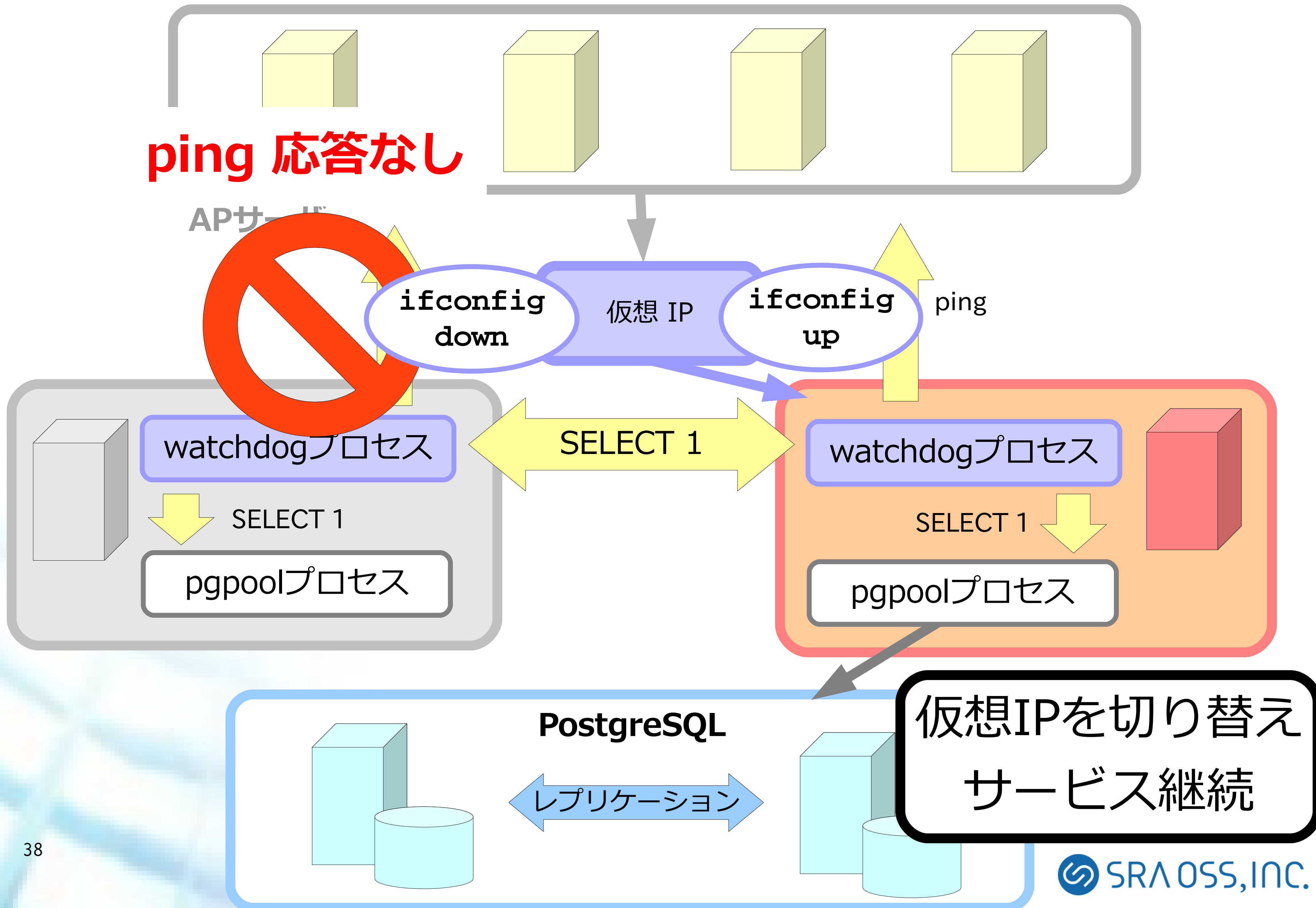
watchdog の死活監視の詳細

wd_lifecheck_interval 秒おきに、以下を実行する
（デフォルトでは10 秒おき）

1. 上位サーバ全部に ping してみる
2. 自分と同居している pgpool-II に wd_lifecheck_query のクエリを実行してみる
3. 自分以外の watchdog と同居している pgpool-II に wd_lifecheck_query のクエリを実行してみる
   - wd_life_point 回リトライ
   - デフォルトではSELECT 1 する

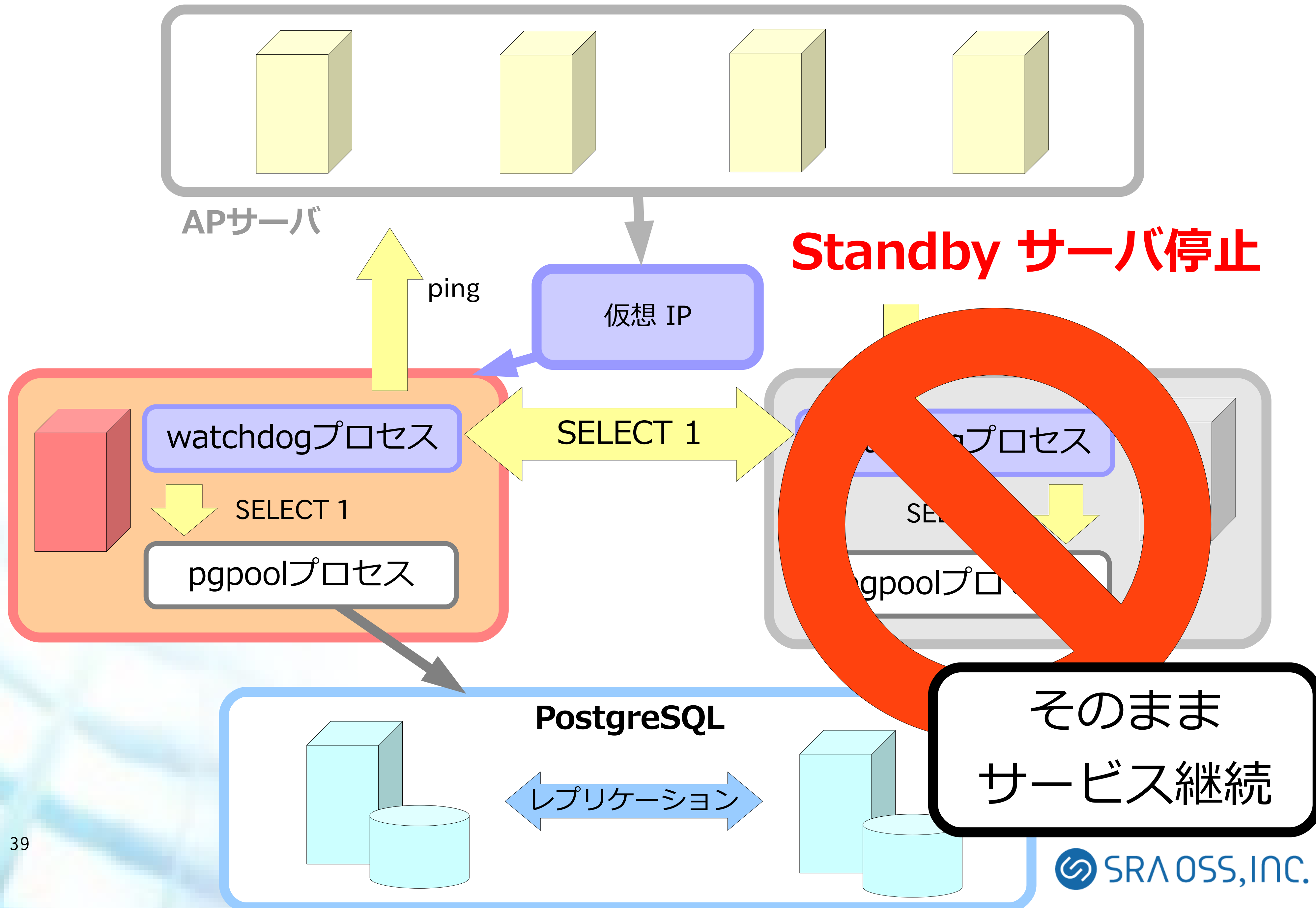
Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS,INC.

死活監視(1) 上流サーバへのping失敗



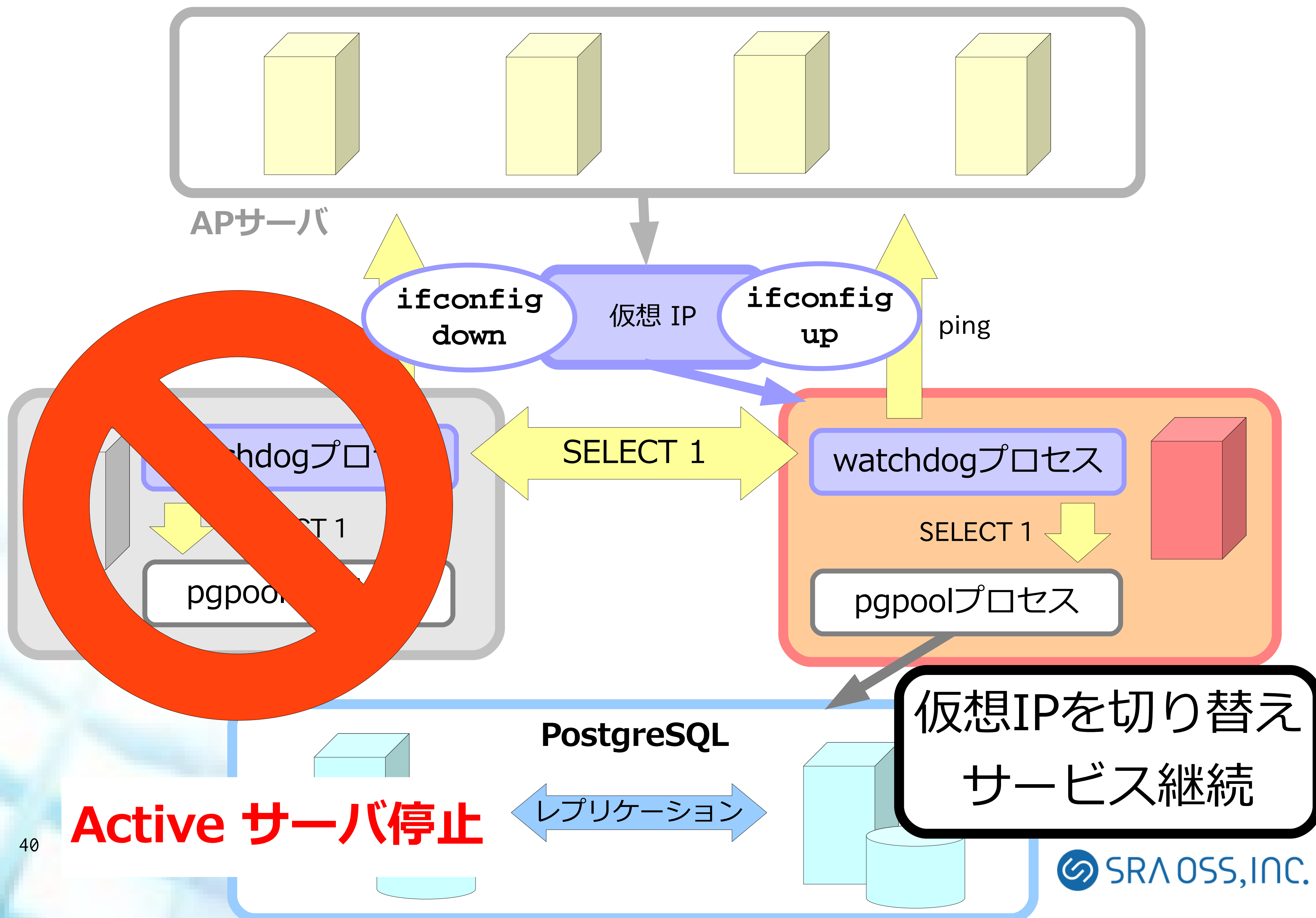
SRA OSS, INC.

死活監視(2) Standby の pgpool-II 停止感知



SRA OSS, INC.

死活監視(3) Active の pgpool-II 停止感知

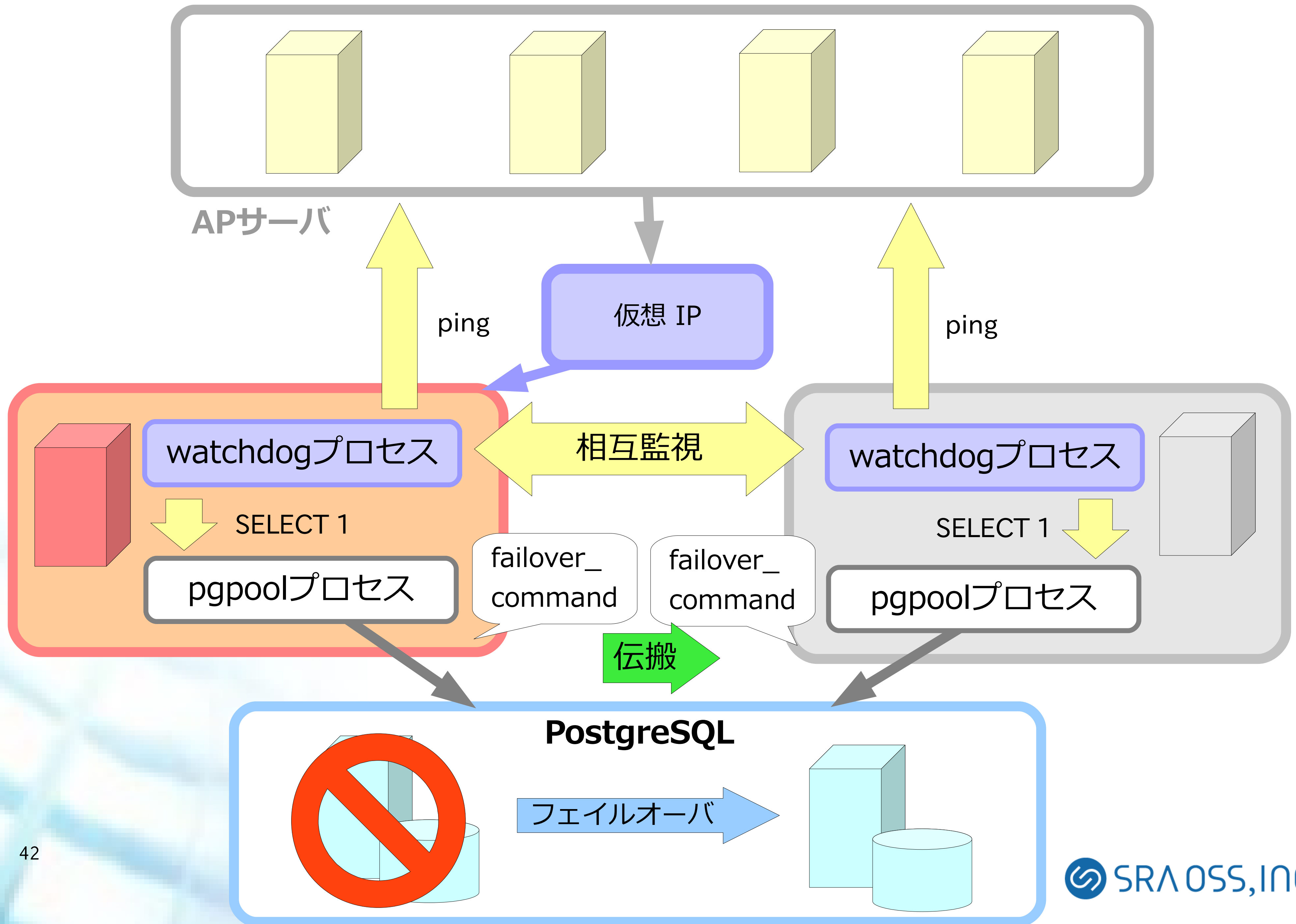


SRA OSS, INC.

DBノード状態の共有

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

pgpoolの足並み統一： フェイルオーバの伝搬

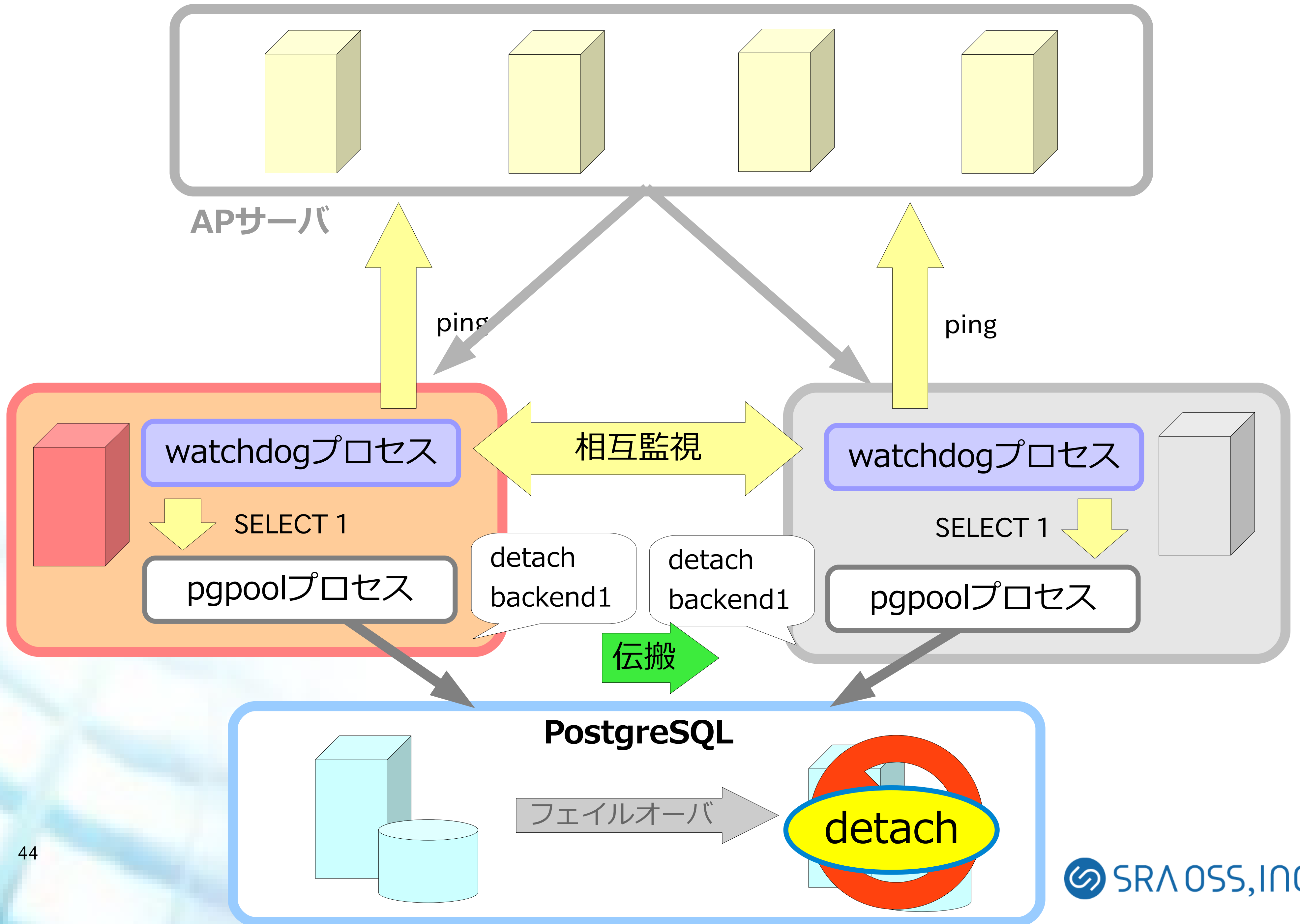


SRA OSS, INC.

watchdogを使った Multi-Master 構成

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS,INC.

watchdog を使った multi-master 構成



SRA OSS, INC.

伝搬する内容

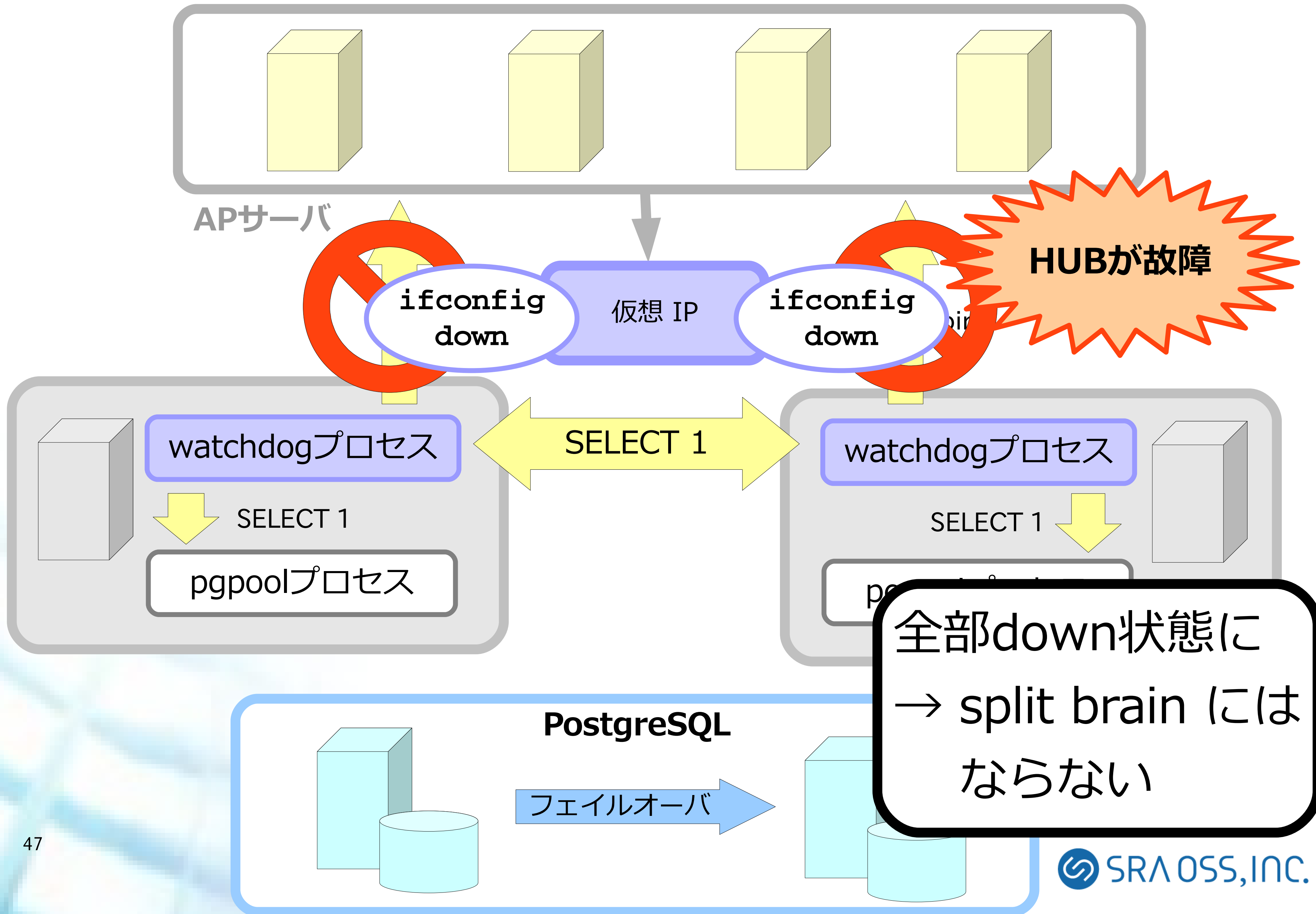
- **failover command 実行時**
  - 全ての pgpool で同じ failover_command が実行
  - 以下の情報を他の pgpool へ通知
    - コマンド種別
      - ノードの追加　(attach)
      - ノード切り離し　(detach)
      - プライマリへの昇格　(promote)
    - 対象ノードのリスト
- **recovery 2nd stage (replication mode)**
  - 開始時と終了時に、他の pgpool へ通知する
    - 開始前に全ての pgpool で接続終了するまで待機
    - リカバリ中は全ての pgpool で新規接続の受付を中止

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

watchdog の注意事項

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

split brain ? （意図に反して、全サーバがActiveになろうとすること）



SRA OSS, INC.

注意事項 (1)

- **root 権限で pgpool-II を起動する**
  - ifconfig, arping コマンドを実行するため
  - failover.sh 内での パスワード無し ssh に工夫が必要。postgres ユーザに su してから　ssh を実行するなど。
- **watchdog の状態を取得する手段が不十分**
  - pcp コマンド, SHOW コマンドで取得できるよう拡充したい。
- **オンラインリカバリの 2nd stage 中に pgpool が落ちた場合に、不具合あり**
  - 他の pgpool でリカバリ終了が検出できず、いつまでたっても接続が受け付けられなくなることがあり得る。

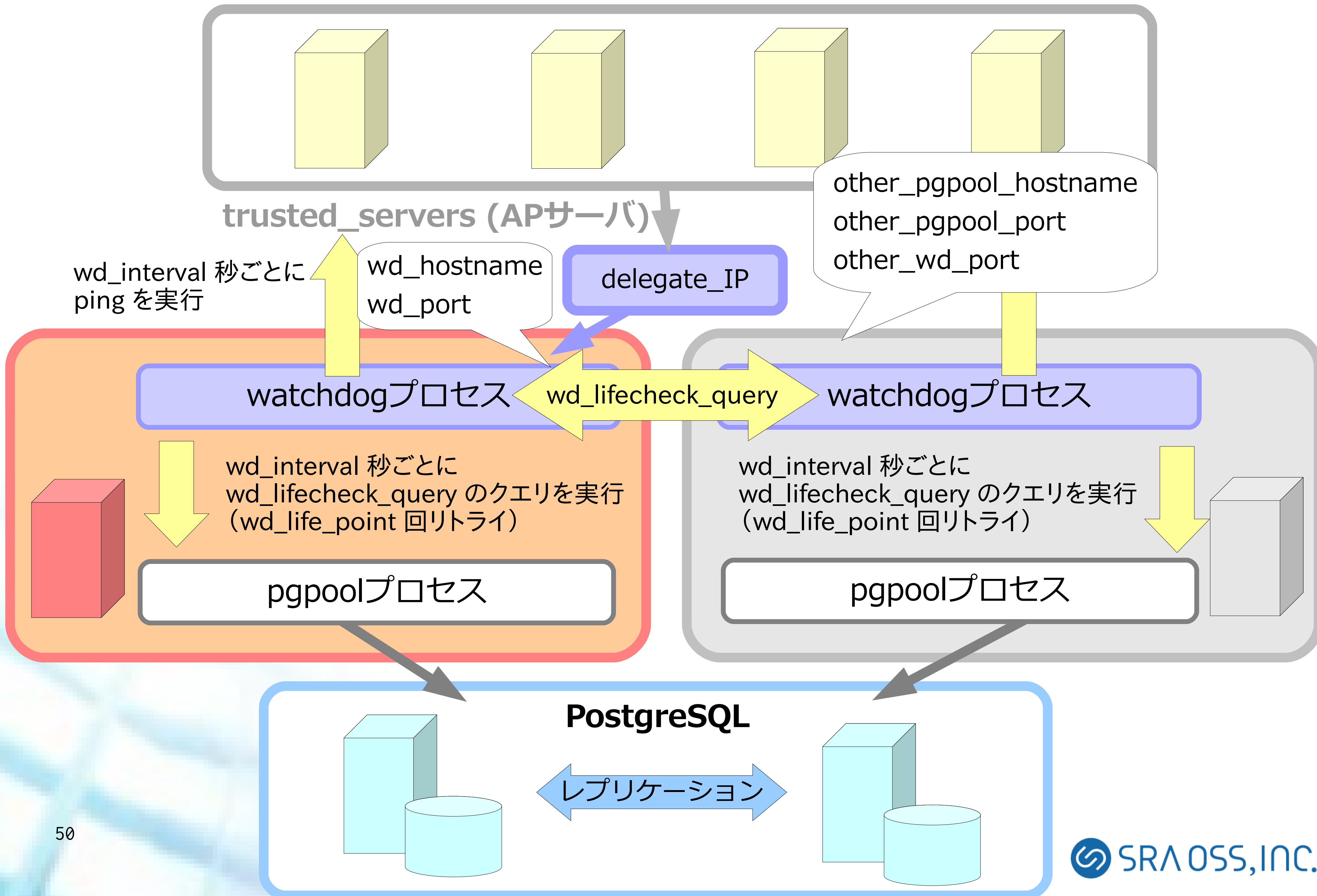
Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

注意事項 (2)

- **サーバ間で時刻が同期している必要がある**
  - 新しいアクティブを決めるのに「起動時刻」を用いる
- **ホスト名の表記に注意**
  - wd_hostname を “localhost” としてはいけない。他サーバとの弁別に使うため。
  - 他の pgpool の other_pgpool_hostname の値と一致している必要がある。
  - 省略すると、自動的に hostname コマンドの結果。
- 仮想環境でない実マシンでだけ発生する問題がある？

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS,INC.

watchdog 関連パラメータ（pgpool.conf）



SRA OSS, INC.

watchdog のデモ

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

OSSPC16 (Active)

```
#--------------------------------------------------------------
# WATCHDOG
#--------------------------------------------------------------
use_watchdog = on
                                    # Activates watchdog
trusted_servers = 'localhost'
                                    # trusted server list whic
                                    # to confirm network conne
                                    # (hostA,hostB,hostC,...)
delegate_IP = '133.137.177.143'
                                    # delegate IP address
wd_hostname = 'osspc16'
                                    # Host name or IP address
wd_port = 19000
                                    # port number for watchdog
wd_interval = 3
                                    # lifecheck interval (sec)
ping_path = '/bin'
                                    # ping command path
ifconfig_path = '/sbin'
                                    # ifconfig command path
if_up_cmd = 'ifconfig eth0:0 inet $_IP_$ netmask 255.255.255.0
                                    # startup delegate IP comm
if_down_cmd = 'ifconfig eth0:0 down'
                                    # shutdown delegate IP com
arping_path = '/usr/sbin'           # arping command path
arping_cmd = 'arping -U $_IP_$ -w 1'
                                    # arping command
wd_life_point = 3
                                    # lifecheck retry times
wd_lifecheck_query = 'SELECT 1'
                                    # lifecheck query to pgpoo
# Other pgpool Connection Settings
other_pgpool_hostname0 = 'osspc20'
                                    # Host name or IP address
other_pgpool_port0 = 19999
                                    # Port number for othet pg
other_wd_port0 = 19000
                                    # Port number for othet wa
```

OSSPC20 (Standby)

```
#--------------------------------------------------------------
# WATCHDOG
#--------------------------------------------------------------
use_watchdog = on
                                    # Activates watchdog
trusted_servers = 'localhost'
                                    # trusted server list which
                                    # to confirm network connect
                                    # (hostA,hostB,hostC,...)
delegate_IP = '133.137.177.143'
                                    # delegate IP address
wd_hostname = 'osspc20'
                                    # Host name or IP address o
wd_port = 19000
                                    # port number for watchdog s
wd_interval = 3
                                    # lifecheck interval (sec) >
ping_path = '/bin'
                                    # ping command path
ifconfig_path = '/sbin'
                                    # ifconfig command path
if_up_cmd = 'ifconfig eth0:0 inet $_IP_$ netmask 255.255.255.0'
                                    # startup delegate IP comman
if_down_cmd = 'ifconfig eth0:0 down'
                                    # shutdown delegate IP comma
arping_path = '/usr/sbin'           # arping command path
arping_cmd = 'arping -U $_IP_$ -w 1'
                                    # arping command
wd_life_point = 3
                                    # lifecheck retry times
wd_lifecheck_query = 'SELECT 1'
                                    # lifecheck query to pgpool
# Other pgpool Connection Settings
other_pgpool_hostname0 = 'osspc16'
                                    # Host name or IP address to
other_pgpool_port0 = 19999
                                    # Port number for othet pgpo
other_wd_port0 = 19000
                                    # Port number for othet watc
```


Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.

まとめ

pgpool-II 3.2 では、2 つの新機能が追加されました！

1. On Memory Query Cache

- SELECTの結果をキャッシュして、高速化
- ロードバランスと併用すると、かなりの効果が期待できる

2. Watchdog

- pgpool-II の HA 構成
- マルチマスタ構成、アクティブ/スタンバイ構成で、PostgreSQL のより安全な運用ができる

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.

おわりに

- pgpool-IIのダウンロードはこちら
  http://pgfoundry.org/projects/pgpool/
- メインWebサイト
  http://pgpool.projects.postgresql.org/
- SRA OSS's website
  http://pgpool.srasoss.jp/
- Twitter
  @pgpool2

Copyright © 2012 SRA OSS, Inc. Japan All rights reserved.
SRA OSS, INC.