

pgpool-II最新情報

pgpool-II 3.1の最新機能とは？

SRA OSS, Inc. 日本支社
石井 達夫

A man in a suit is standing at a wooden podium, speaking into a microphone. He is facing a large audience seated in a rustic room with exposed wooden beams and hanging plants. A large screen to the left of the speaker displays the presentation content. The room has a warm, industrial feel with pendant lights and a large orange door in the background.

SRA OSS, INC.

An introduction to pgpool-II Version 3

SRA OSS, Inc. Japan
Tatsuo Ishii

GSoC 2011 - Accepted Projects - Fx for Vine Linux

ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)

http://www.google-melange.com/gsoc/projects/list/google/gsoc2011

GSoC 2011 - Accepted Projects

Search
Help
Logout

Student	Title	Organization	Mentor
Karan Pratap Singh	KDE Plasma Educational Desktop	KDE	aseigo
Alexander Korotkov	Fast GiST index build	PostgreSQL Global Development Group	Heikki Linnakangas
Leonardo Sapiras	New phpPgAdmin Plugin Architecture	PostgreSQL Global Development Group	Jehan-Guillaume (iogux) de Rorthais
Luis Ochoa	Development of improvements in pgAdmin physical database designer module	PostgreSQL Global Development Group	Guillaume Lelarge
Masanori Yamazaki	Caching query results in pgpool-II	PostgreSQL Global Development Group	Tatsuo Ishii
Sivasankar	IMPROVE THE AUTO-CONFIGURATION SCRIPT IN POSTGRES BASED ON THE STATE OF THE DATABASE AND PREVIOUS USAGE OF DATA.	PostgreSQL Global Development Group	Mark Wang
Tomas Pospisil	Indexing XML data in PostgreSQL	PostgreSQL Global Development Group	Gregory Stark
Zheng Yang	Enhancing Foreign-data Wrapper (FDW) functionality for PostgreSQL	PostgreSQL Global Development Group	Dave Page

Proxy: なし

PostgreSQLはどんどん 高性能化する

- レプリケーション/クラスタリング対応
 - ストリーミングレプリケーション<--pgpool-IIの取り組み
- 単体のDBエンジンとしてもより高性能に
 - CPUスケールビリティ
 - 更新性能向上(HOT)
 - クエリオプティマイザの改良
- pgpool-IIについてお話しする前に、まず
PostgreSQLの性能向上の例を見てみましょう

実は地道に改良されている クエリオプティマイザ

- 実用的には、これが嬉しい
 - バージョンアップしただけで、アプリケーションが速くなる
 - バージョン毎に改良が加えられるので、新しいバージョンを使うのがおすすめ
 - 本日は、大きなテーブルの結合(1000万件、300万件、100万件)の性能向上をバージョンを追って紹介します
 - おまけで、MySQLもテストしてみました

テスト環境

- Vine Linux 5.2 (kernel 2.6.27)
 - mem: 4GB, CPU: Core2Duo 1.6GHz(dual core)
 - disk: 外付けUSB 2.0 2.5インチディスク
- すべてノーチューニング(MySQLはmy_large.cnf, InnoDB使用)
- PostgreSQL 8.1.23, 8.2.21, 8.3.15, 8.4.8, 9.0.4, MySQL 5.1.41
- テーブルは、1,095,594件、3,605,456件、10,543,404件、インデックスあり。3つのテーブルをJOINし、400件抽出する
- DB容量は2.1GB(PostgreSQL, MySQL)

SELECT #1(副問い合わせ使用)

SELECT

C.package_name,

C.name AS class_name,

C.id AS class_id,

M.name AS method_name,

M.id AS method_id

FROM (SELECT id, name FROM java_method WHERE name LIKE 'copy%') M

JOIN class_method_map CM on CM.method_id = M.id

JOIN java_class C on C.id = CM.class_id

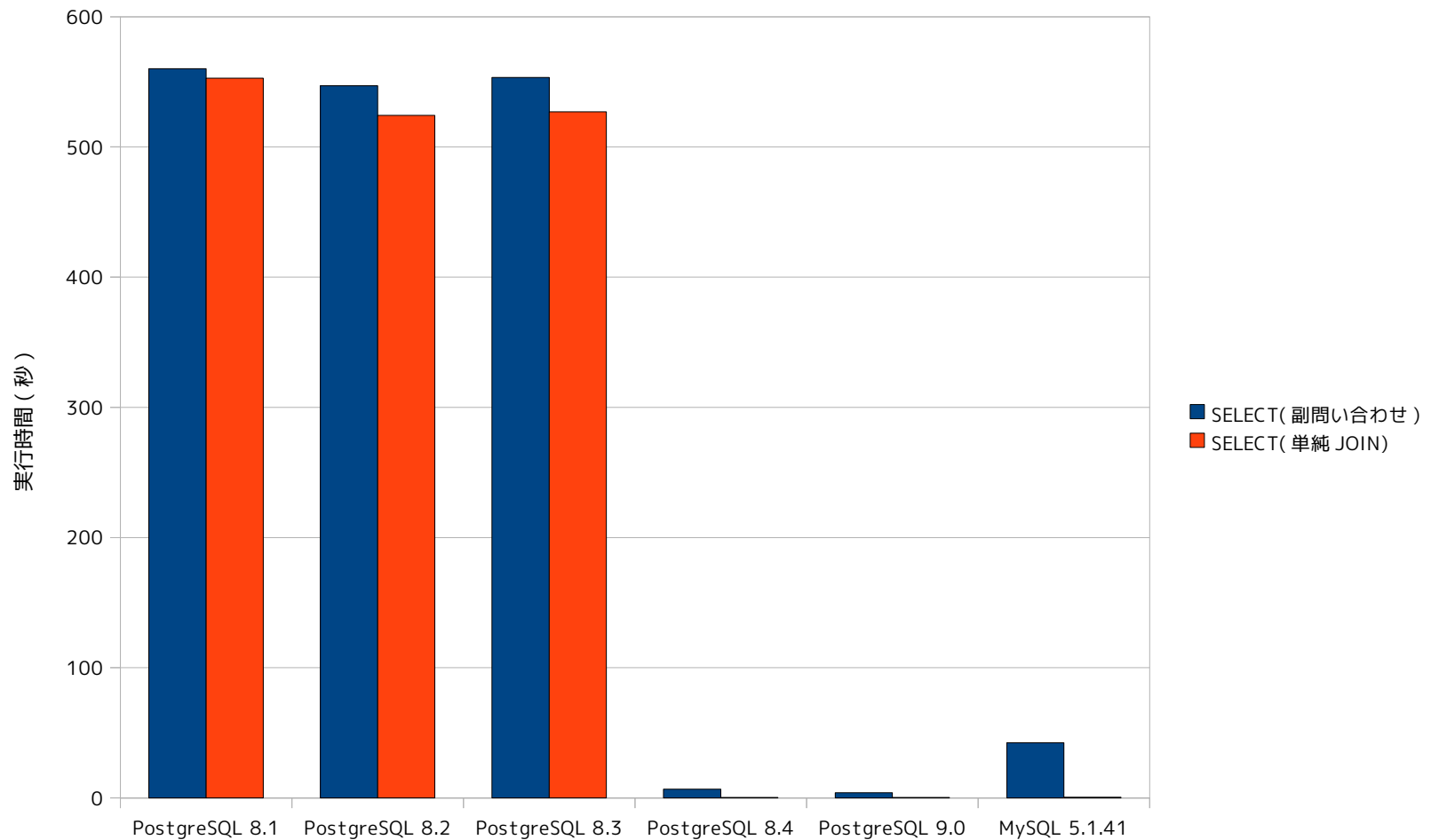
WHERE

C.name LIKE 'FileUtil%';

SELECT #2(単純JOIN)

```
SELECT
  C.package_name,
  C.name AS class_name,
  C.id AS class_id,
  M.name AS method_name,
  M.id AS method_id
FROM java_method M,
     class_method_map CM,
     java_class C
WHERE
  C.name LIKE 'FileUtil%' AND
  M.name LIKE 'copy%' AND
  CM.method_id = M.id AND
  C.id = CM.class_id;
```


PostgreSQL 8.1->9.0で140倍の 高速化！



pgpool-II 3.1の新機能

- SQLパーサの更新
 - 実はpgpool-IIは、SQLパーサを持っている
 - 複雑なSQLを確実に解析するため
 - クエリの書き換えを行うため
 - SQLパーサのコードはPostgreSQLから借用
 - PostgreSQL/pgpool-IIはどちらもBSDライセンスなので問題なし
 - PostgreSQL 9.0のパーサを導入
 - CREATE INDEX文でインデックス名の省略など、PostgreSQL 9.0構文をサポート

Black/white関数リストの改良

- Black/white関数リストとは？
 - PostgreSQLでは、SELECT文でありながら、実はデータベースを変更する関数を実行できる
 - なので、「SELECTだから負荷分散してもいいよね」とやってしまうと、痛い目に...
 - 例: currval(), nextval(), setval(), set_config()...
 - 対策は...
 - レプリケーションモードでは、そういう関数を使っているSELECTはすべてのノードにSELECTを投げる
 - マスタ/スレーブモードでは、必ずマスタ(プライマリ)に投げる

Black/white関数リストの改良(2)

- 問題は、どうやって書き込みを行う関数を見つけるか？
 - 関数の属性がVOLATILEなら、DBを変更する、という議論は PostgreSQL 8.0より前では成り立たない
 - 結局人間が指定するしかない。それが Black/white関数リスト
 - Black関数リスト
 - DBを変更する関数をコンマ区切りで指定する。このリストに現れない関数はDBに書き込みを行わないと見なす
 - White関数リスト
 - DBを変更しない関数をコンマ区切りで指定する。このリストに現れない関数はDBに書き込みを行うと見なす
 - Black、Whiteのどちらかを指定する
 - 通常はBlack関数リストを使う

Black/white関数リストの改良(3)

- しかし、ユーザが大量に書き込みを行う関数を自作している環境では、Black関数リストの記述が長くなる
- そこで、正規表現で関数を指定できるようにした
 - `my_project_function1,`
`my_project_function2,my_project_function3.`
`.. --> my_project_function.*`

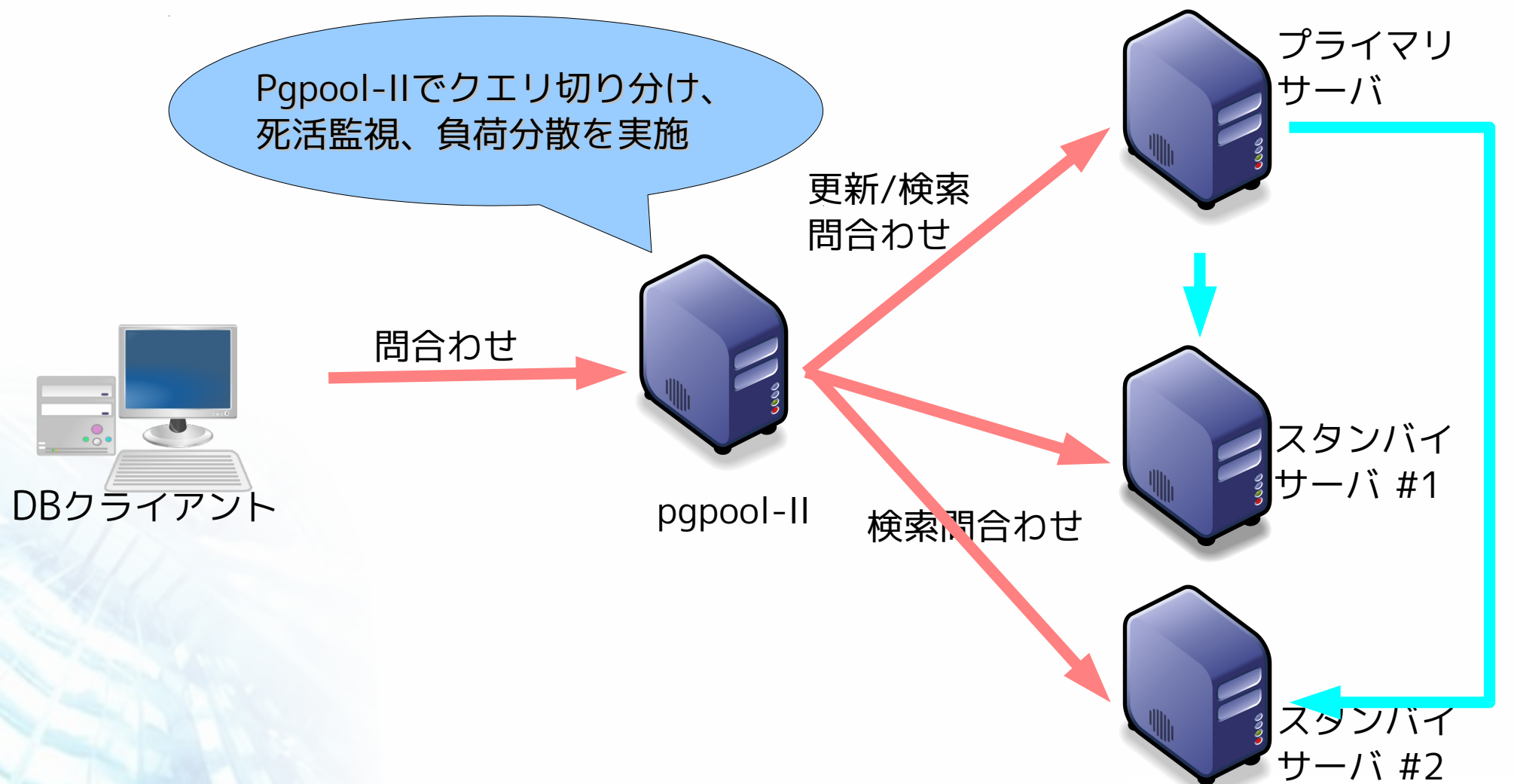
syslogへのログ出力が可能に

- 以前のおすすめ
 - pgpool -n |/usr/sbin/rotatelogs
/var/log/pgpool/pgpool.log.%A 86400
- 3.1では簡単にログをsyslogに出せる
 - log_destination = 'syslog'
 - syslog_facility = 'LOCAL0'
 - syslog_ident = 'pgpool'

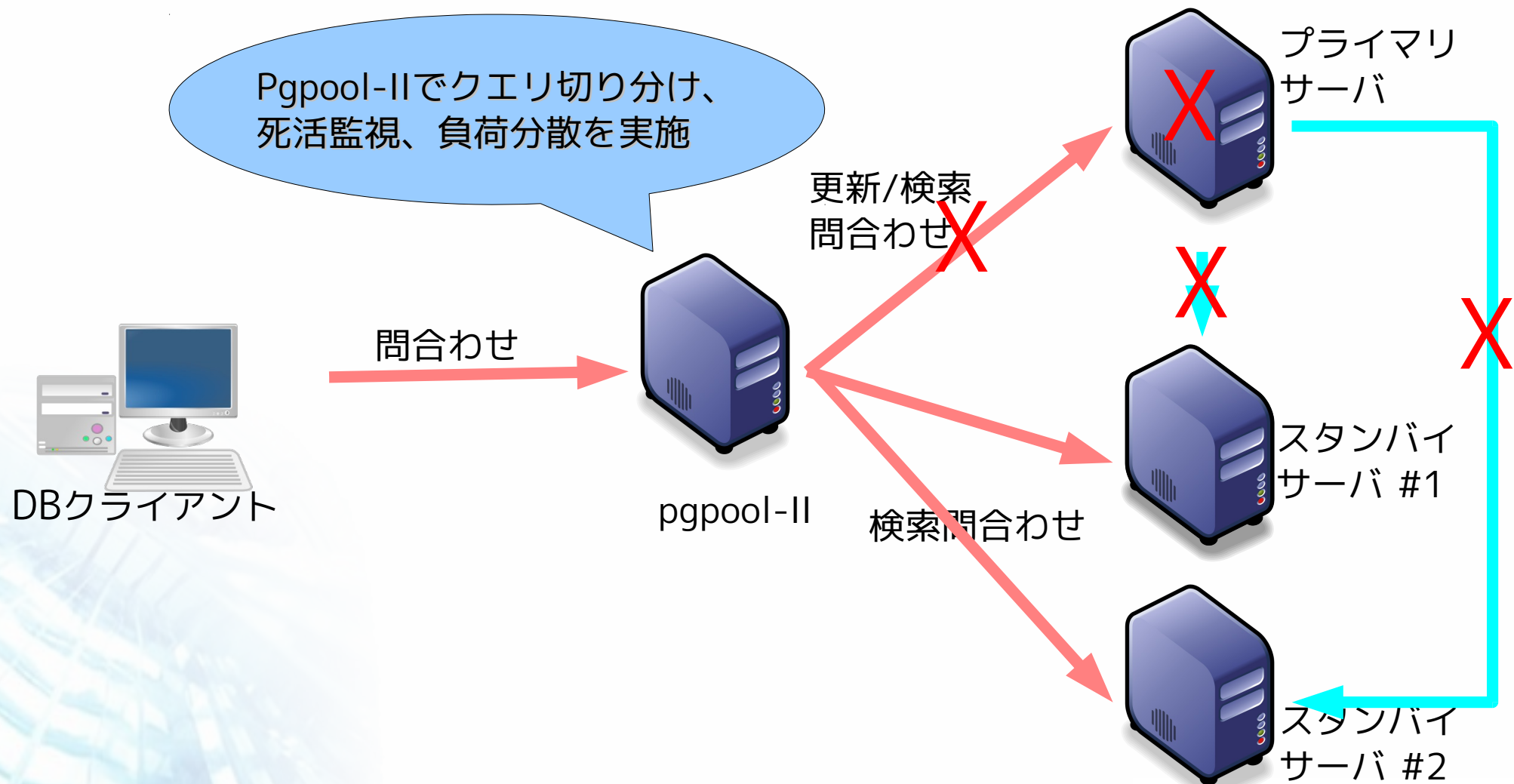
フォロースターコマンド

- pgpool-II+ストリーミングレプリケーションで、3台以上のPostgreSQLを使う構成のときに有効
- フォロースターコマンドがないと、なぜまずいか？を最初に説明します

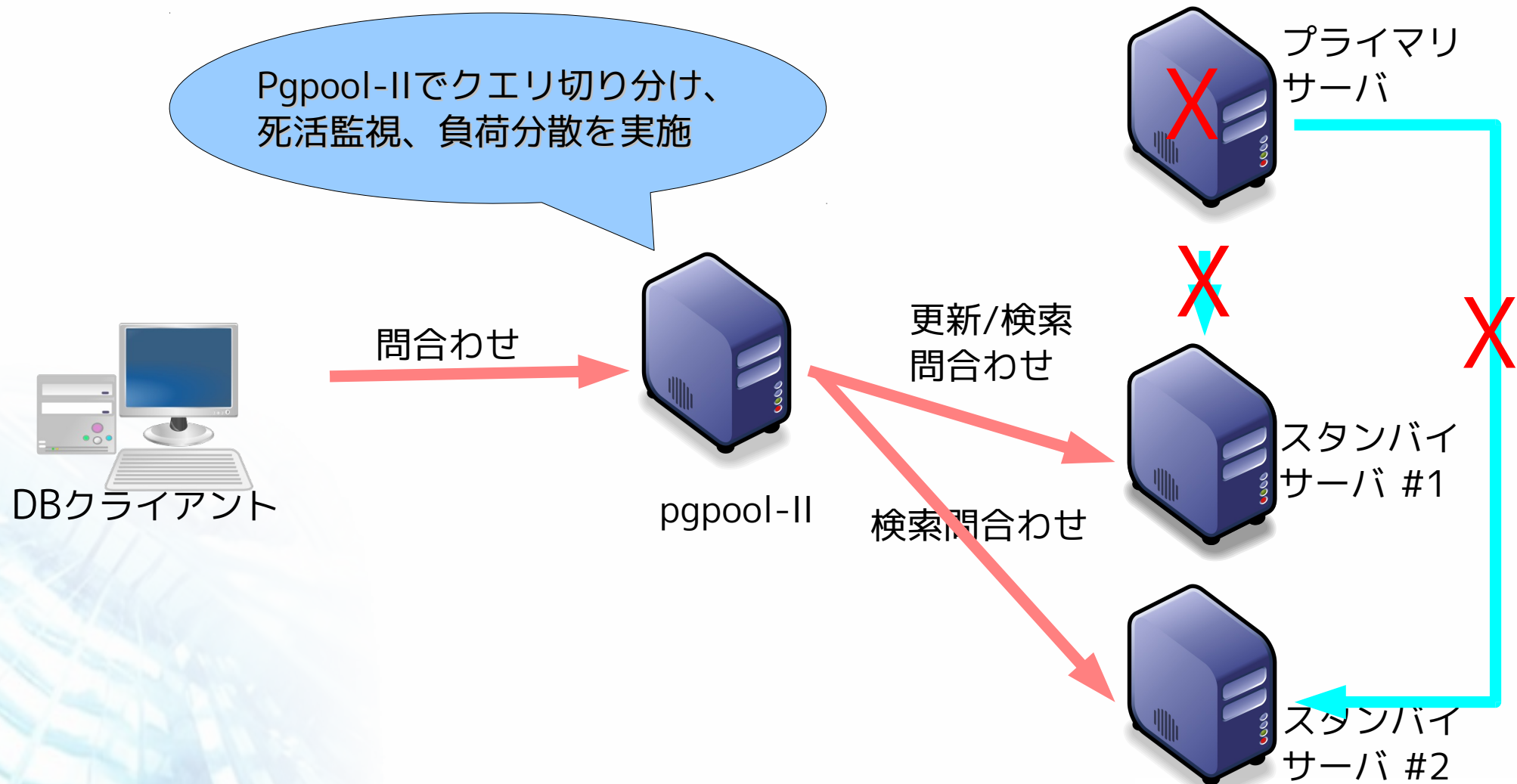
PostgreSQL 3台構成で運用中



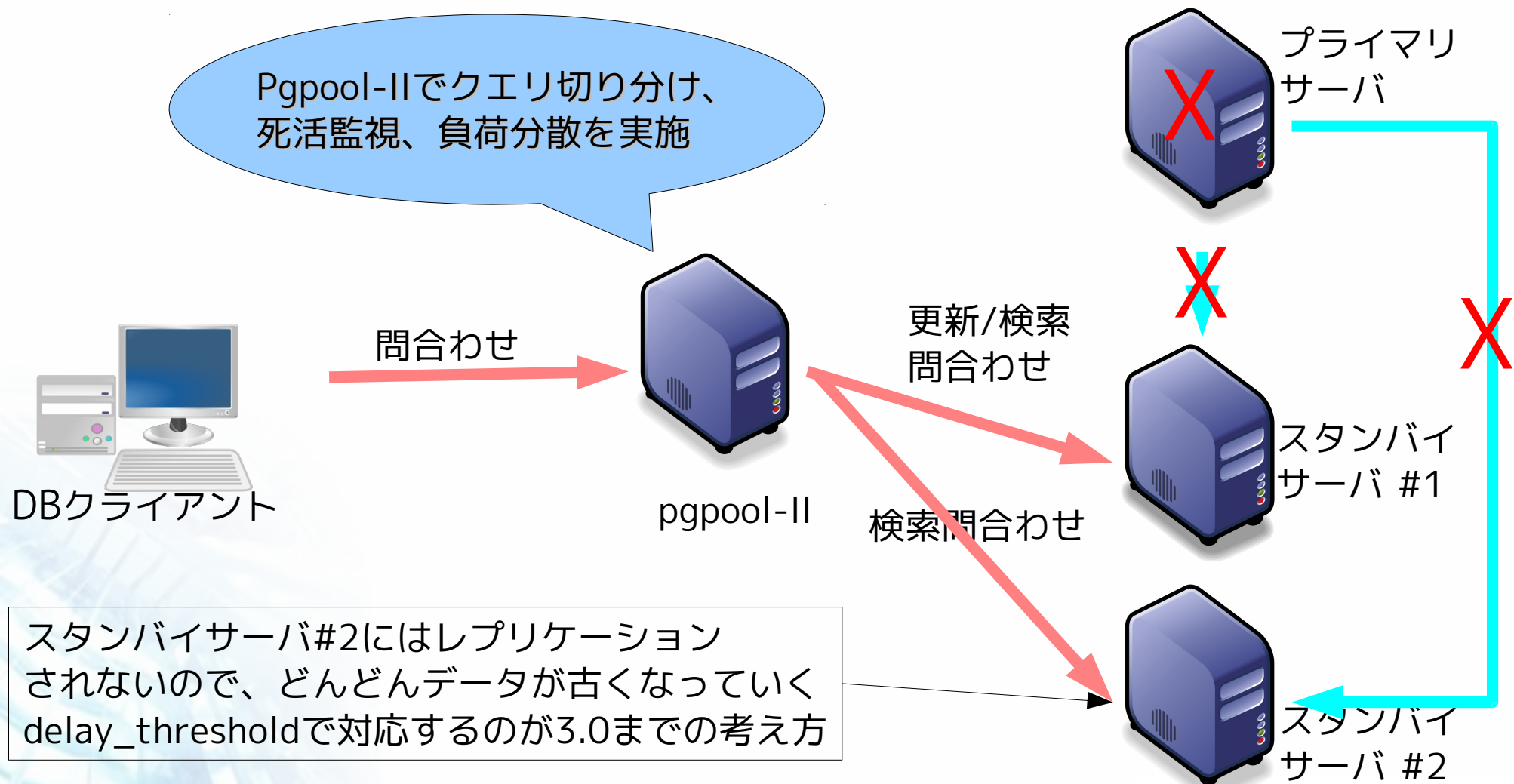
プライマリサーバが故障！



スタンバイサーバ#1が「昇格」(promote) 旧プライマリサーバを切り離す



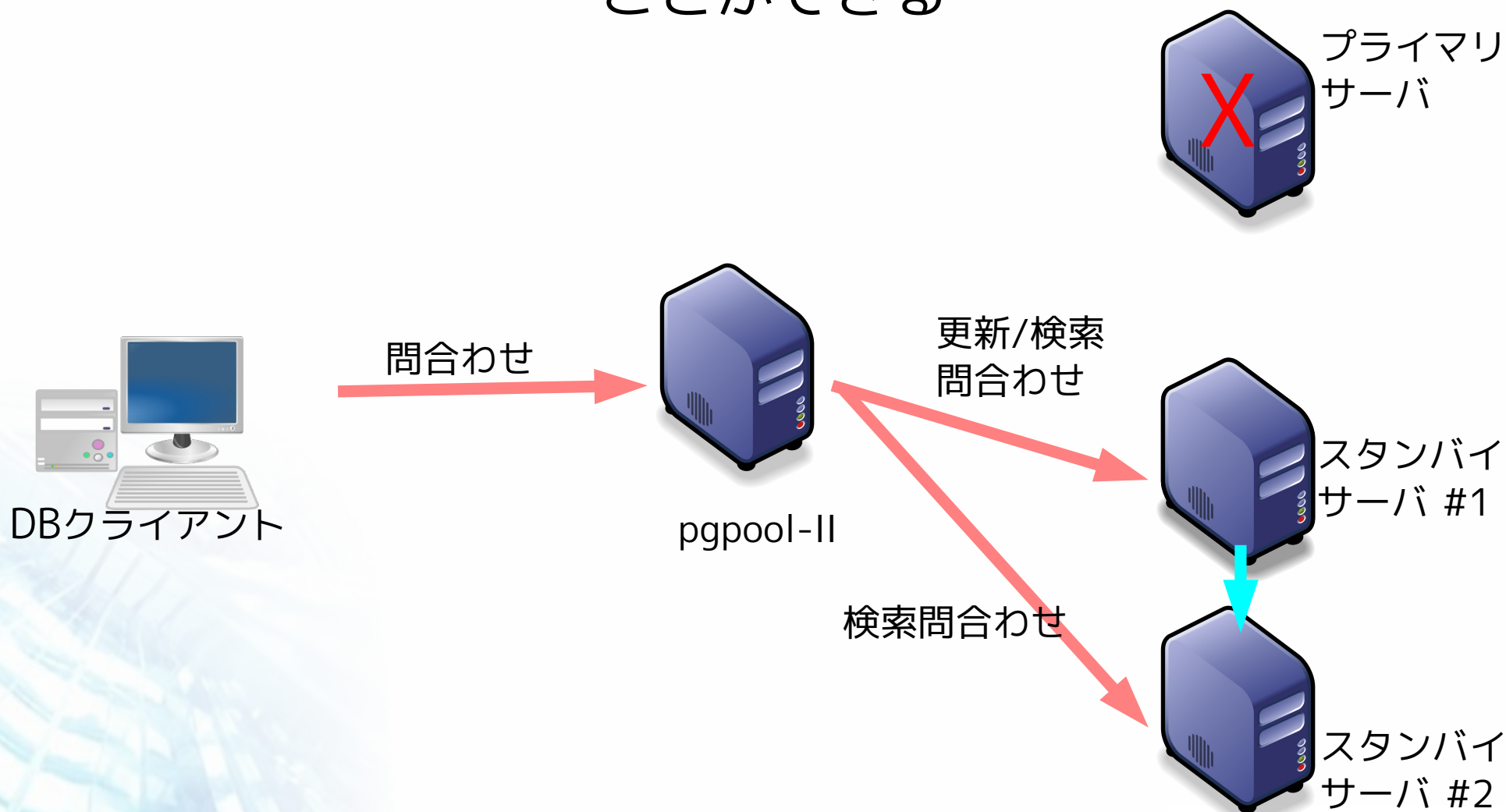
しかし、スタンバイサーバ#2は浮いたまま



フォローマスターコマンドとは

- スタンバイが昇格した後に実行するスクリプトを定義できる機能
- スクリプトには以下のものを書くのがおすすめ
 - スタンバイサーバ#2を停止(pg_ctlをssh経由で実行)
 - スタンバイサーバ#1をプライマリとして、スタンバイサーバ#2を「オンラインリカバリ」する(pcp_recovery_nodeの実行)

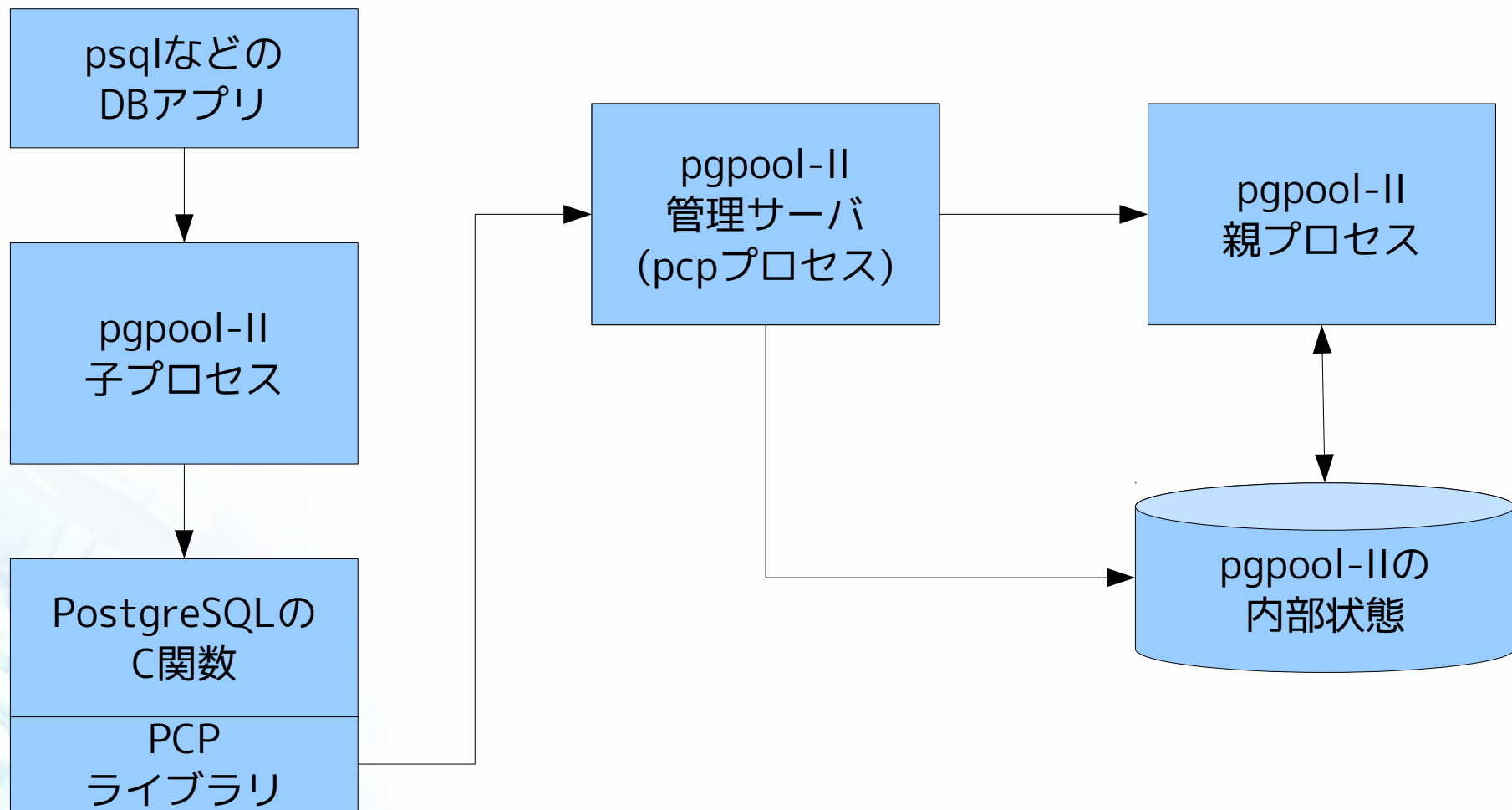
フォロースターコマンドを使って、
スタンバイサーバ#2を新プライマリサーバに同期させる
ことができる



管理機能の改良

- SELECT文を使ってpgpool-IIの状態を取り出せる機能が追加
 - CREATE FUNCTIONまたはCREATE EXTENSION用のモジュールの提供
 - この機能を使って、pgAdmin, phpPgAdminからpgpool-IIの状態をチェック可能になる！？
 - show pool_statusと同等の機能を持つpcpコマンド、およびユーザ定義関数も追加。SELECT文+WHERE句で、欲しい情報を検索できる

C関数として呼び出せる pgpool-II管理コマンド



実行例

```
CREATE SCHEMA pgpool;
CREATE EXTENSION pgpool_adm WITH SCHEMA pgpool ;
SELECT * FROM pgpool.pcp_node_info(0, 'localhost', 10, 9898,
    'pgpool_adm', 'password');
```

host	port	status	weight
localhost	5491	Waiting for connection to start	1.46602e+13

実行例

```
test=# SELECT * FROM pcpl_status('localhost', 10, 9898, 'postgres', 'pgpoolAdmin')
test=# where item like '%time%';
```

item	value	description
child_life_time	0	if idle for this seconds, child exits
connection_life_time	0	if idle for this seconds, connection closes
authentication_timeout	60	maximum time in seconds to complete client authentication
print_timestamp	1	if true print time stamp to each log line
health_check_timeout	10	health check timeout
pcpl_timeout	10	PCP timeout for an idle client
recovery_timeout	60	max time in seconds to wait for the recovering node's

postmaster
(7 rows)

今後の予定 & 将来やりたいこと

- 今後の予定
 - pgpool-II 3.1の5月中の正式リリースを目指す
- 将来やりたいこと
 - メモリーベースのクエリキャッシュ
 - 今もクエリキャッシュはあるが、今ひとつ使いにくく、性能が出ない
 - 大量INSERTの自動バッチ化
 - pgpool-II自体の組み込みHA機能
 - 簡単に単一障害点を避ける

参考URL

- pgpool-IIのダウンロードはこちら
 - <http://pgfoundry.org/projects/pgpool/>
 - 開発ML(英語)もこちらから登録できます
- メインWebサイト
 - <http://pgpool.projects.postgresql.org/>
- SRA OSS's website
 - <http://pgpool.srasoss.jp>
 - 日本語ML(pgpool-general-jp)登録もこちらから
- Twitter
 - @pgpool2