

Extrait du Easter-eggs - Spécialiste GNU/Linux

<http://www.easter-eggs.com>

Systeme de synchronisation conditionnel pour PostgreSQL

- Études de cas - Développement applicatif -

Date de mise en ligne : jeudi 26 juin 2008

Easter-eggs - Spécialiste GNU/Linux

Il existe des outils tel que [SQLSync](#) permettant de synchroniser les données d'une base PostgreSQL. Mais cet outil ne permet pas de faire une copie partielle des données.

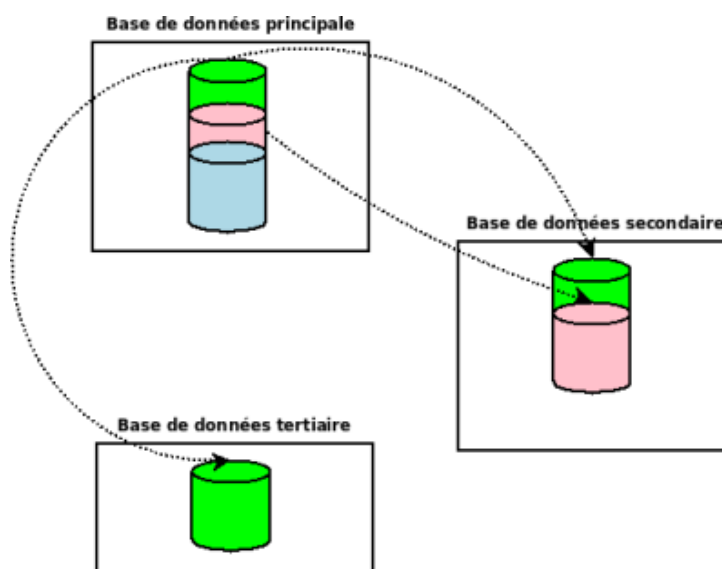
Dans la plupart des cas, il faut programmer la synchronisation soi-même, l'objectif de cet article est de vous proposer une solution simple et élégante.

La problématique

Des données sensibles se trouvent dans la base principale et il ne faut synchroniser qu'une partie de ces données vers les autres bases.

L'architecture dispose des caractéristiques suivantes :

<!-- htmlA -->



<!-- htmlB -->

- Le schéma est identique sur chaque base ;
- Les opérations d'écriture se font seulement sur la base principale ;
- Les bases sont accessibles en réseau.

La solution

Pour résoudre le problème, plusieurs approches ont été testées avant de retenir l'utilisation d'une fonction de hachage dont voici le principe :

- chaque table doit disposer d'un identifiant unique. (champ id de type SERIAL) ;
- un programme récupère le couple id et la sum de hachage pour chaque table de chaque base qui doivent être synchronisées ;
- il compare le couple et si celui-ci est différent alors les données sont mises à jour.

Implémentation au niveau de la base de données

Le calcul de la somme de hachage se fait au niveau de la base de données à l'aide d'une fonction :

```
SELECT id,md5 FROM get_table_md5('nom_de_la_table') ;

id |          md5 +----+----- 4 | 9f3bcd2fae528244669613ae0466cc3c 5 |
1f3bcd2fa24528244669613ae66cc3c3zd ... (42 row)

Voici le code de la fonction PL/Perl : CREATE LANGUAGE plperl ; CREATE TYPE table_md5 AS (id INTEGER,
md5 TEXT) ; CREATE OR REPLACE FUNCTION get_table_md5(varchar) RETURNS SETOF table_md5 AS $$
my ($rv, $status, $nrows, $row) ; # Get Table OID $rv = spi_exec_query("SELECT c.oid AS oid FROM
pg_catalog.pg_class c LEFT JOIN pg_catalog.pg_namespace n ON n.oid = c.renamespace WHERE
c.relname LIKE '$_[0]';"); $status = $rv->status ; $nrows = $rv->processed ; return undef if ($nrows != 1) ;
my $oid = $rv->rows[0]->oid ; # Get Table attributs $rv = spi_exec_query("SELECT attname,attypid FROM
pg_catalog.pg_attribute a WHERE a.attrelid = $oid AND a.attnum > 0 AND NOT a.attisdropped"); $status =
$rv->status ; $nrows = $rv->processed ; return undef if ($nrows < 1) ; my $atts ; my $atts_not_null ;
foreach my $rn (0 .. $nrows - 1) if ($rv->rows[$rn]->attypid == 16) $atts .= " (CASE WHEN $rv->rows[$rn]->attname
THEN 't' ELSE 'f' END) AS $rv->rows[$rn]->attname" ; else $atts .= " $rv->rows[$rn]->attname ::text" ; $atts_not_null
.= " (CASE WHEN $rv->rows[$rn]->attname IS NULL THEN ' ' ELSE $rv->rows[$rn]->attname END)" ; $atts .= ',' ;
if ($rn != $nrows - 1) ; $atts_not_null .= ' || ' if ($rn != $nrows - 1) ; # Calc MD5 my $sql = "SELECT id,
MD5($atts_not_null) AS md5 FROM (SELECT $atts FROM $_[0]) AS $_[0] ;" ; $rv = spi_exec_query($sql) ;
$nrows = $rv->processed ; foreach my $rn (0 .. $nrows - 1) return_next( id => $rv->rows[$rn]->id, md5 =>
$rv->rows[$rn]->md5 ) ; return undef ; $$ LANGUAGE plperl ;
```

Pour utiliser cette fonction vous avez besoin du support du langage de programmation PL/Perl dans PostgreSQL, celui-ci se trouve dans le paquet [Debian](#) : postgresql-plperl-

La gestion des conditions

Le choix des données à synchroniser se fait individuellement pour chaque table à l'aide de la clause WHERE de la manière suivante :

```
SELECT id,md5 FROM get_table_md5('nom_de_la_table') WHERE id IN(SELECT id FROM nom_de_la_table
WHERE nom_de_la_table.champ LIKE 'sync') ;
```

```
id |          md5 +----+----- 1 | fdd56eabd4bb997e453e33f0022d46c1 (1 row)
```

Implantation du script de synchronisation

La synchronisation des données peut être réalisée à l'aide de n'importe quel langage de programmation, il suffit juste de disposer d'un accès à la base de données.

Voici un [exemple](#) du script écrit par Emmanuel Saracco en PHP5 qui repose sur les modules [PEAR](#) suivants :

- [DB](#) : pour l'accès à la base de données.
- [Console_Getopt](#) : pour la gestion de la ligne de commande.

Ce script nécessite un fichier de configuration, en voici un [exemple](#). Ce fichier contient la configuration des différentes base de données et les conditions sous-forme de clauses where.

Conclusion

Cette méthode de synchronisation fonctionne en production chez un client pour une base de données de plus de 100 tables. Elle permet de synchroniser le contenu d'une base vers deux autres.

Les avantages de cette méthode sont les suivants :

- le fait de pouvoir choisir le contenu de la clause where permet de réaliser des règles de synchronisation très complexes ;
- les performances sont très bonnes car les transferts réseau sont limités ;
- il est possible d'écrire le script de synchronisation avec n'importe quel langage de programmation.