Annexe B – Mise en œuvre détaillée de Bucardo.

installation de centos 6.3 : serveur de base
agent snmp

installation de dépots supplémentaires :
avant cela, vérifier :

```
# Update Perl, if necessary
sudo yum update perl
```

Puis :

```
# Add the RPMForge repository (if not available)
#wget http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el5.rf.i386.rpm
wget http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el5.rf.x86_64.rpm
rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt
rpm -K rpmforge-release-0.5.2-2.el5.rf.*.rpm
rpm -i rpmforge-release-0.5.2-2.el5.rf.*.rpm

# Add the EPEL repository (if not available)
#wget http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-4.noarch.rpm
wget http://download.fedora.redhat.com/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm
rpm -i epel-release-5-4.noarch.rp
```

/etc/yum.repos.d/rpmforge.repo make the following changes:

```
...
[rpmforge]
...
enabled = 0
...

[rpmforge-extras]
...
enabled = 1
...
includepkgs = perl-DBD-Pg perl-DBI
...
```

In /etc/yum.repos.d/epel.repo make the following changes:

```
[epel]
...
includepkgs=perl-version perl-DBIx-Safe
```

yum install postgresql-server

puis

on installe les packages nécessaires :

sudo yum install perl-DBI perl-DBD-Pg perl-DBIx-Safe perl-version

```
wget http://bucardo.org/downloads/Bucardo-4.5.tar.gz
tar xzf Bucardo-4.5.tar.gz
cd Bucardo-4.5
perl Makefile.PL
# Ignore warnings regarding the ExtUtil::MakeMaker error version
make
sudo make install
```

PostgreSQL PL/Perl

Pour que Bucardo fonctionne de façon satisfaisante, il faut aussi installer plperl :

yum install postgresql-plperl

en ce qui concerne le sgbd postgreql, commencer par :

modifier les fichiers pg_hba.conf (host all postgres 10.11.12.0/24 trust)
et postgresql.conf (listen = '*" pour autoriser les connexions de partout)

puis installer bucardo :
Ne pas oublier dans pg_hba.conf pour les accès distants host all all ..... trust
pour autoriser tous les utilisateurs, y compris bucardo par exemple :

bucardo_ctl install

bucardo_ctl show all

ou bucardo_ctl set foo=bar pour modifier certains paramètres

préparation de la machine postgresql-centos-slave à partir
du clone postgresql-centos

modifier le nom : nano /etc/sysconfig/network
puis /etc/hosts en ajoutant l'adresse de la machine et les bons noms
puis redémarrer.

un petit coup de yum update accessoirement si ce n'est déjà fait

autoriser les connexions en modifiant les fichiers /usr/lib/pgsql/data/pg_hba.conf et postgres.conf

puis en utilisateur su postgres et en se mettant dans son répertoire home,

lancer la base de données

/usr/bin/pg_ctl -D /var/lib/pgsql/data/ -l logfile start

faire ps pour voir si le process postgres est présent

tester la connexion à présent :

psql -h 10.11.12.242 -p 5432 -U postgres

je n'ai pas besoin de l'option -W pour demander le mot de passe vu que j'ai
paramétré le mode 'trust' plutôt que 'ident' dans le fichier /var/lib/pgsql/data/pg_hba.conf

les deux bases devraient être opérationnelles

---

vérification de l'installation de pbBench :

il faudra peut être installé yum install postgresql-contrib pour avoir accès à pgbench
un répertoire /usr/share/pgsql/contrib sera créé
Créons nos tables sur les deux machines :

createdb test1 sur le host master :
[root@postgresql-centos-master ~]# createdb -h 10.11.12.241 -p 5432 -U postgres test1

createdb test1 sur le host slave depuis le host master :
[root@postgresql-centos-master ~]# createdb -h 10.11.12.242 -p 5432 -U postgres test1

puis on peuple les bases avec les tables spécifiques à pgbench :
[root@postgresql-centos-master ~]# pgbench -h 10.11.12.241 -p 5432 -U postgres -i test1
[root@postgresql-centos-master ~]# pgbench -h 10.11.12.242 -p 5432 -U postgres -i test1

Nos bases de données sont peuplées, côté maitre et côté escalve.

Continuons à activer bucardo, nous allons essayer le type pushdelta dans un premier
temps, qui correspond à une architecture du type master/slave, avec une option complémentaire
de type fullcopy, sur la table pgbench_history qui ne possède pas d'index ou de clés primaires, on ne
peut donc pas effectuer de pushdelta ou swap. On en fera donc juste une copie.

Pensons à créer un superutilisateur bucardo sur le slave, avec les options : inherit, nocreatedb, nocreaterole
and a variable search_path=bucardo, avec ou sans mot de passe, c'est sans importance si
trust est définie dans /usr/pgsql/data/pg_hba.conf

Indiquons à Bucardo qui sont nos master et slave et où les trouver :
[root@postgresql-centos-master ~]# bucardo_ctl add database test1 name=test1_master port=5432 host=10.11.12.241
Added database "test1_master"
[root@postgresql-centos-master ~]# bucardo_ctl add database test1 name=test1_slave port=5432 host=10.11.12.242
Added database "test1_slave"

puis on ajoute toutes les tables de la base test1 sauf pgbench_history que l'on regroupe dans un ensemble
lié dans une harde (herd) que l'on nomme alpha,
puis de toutes les tables, on n'ajoute que la table pgbench_history dans un autre groupement nommé beta

bucardo_ctl add all tables db=test1_master -T pgbench_history --herd=alpha –verbose
bucardo_ctl add all tables db=test1_master -t pgbench_history --herd=beta --verbose

Nous allons maintenant ajouter les "sync" autrement dit les réplications, en deux étapes
vu que les replicataions se font par regroupement (herd), donc, dans notre cas, nous allons
faire un "sync" de type "pushdata" avec le "herd" alpha, puis un "sync" de type "fullcopy"
avec le "herd" beta.

tout d'abord :
[root@postgresql-centos-master ~]# bucardo_ctl add sync benchdeltaalpha source=alpha targetdb=test1_slave
type=pushdelta

puis ensuite :
[root@postgresql-centos-master ~]#bucardo_ctl add sync benchcopybeta source=beta targetdb=test1_slave
type=fullcopy

Mais avant de démarrer la replication à l'aide bucardo, effectuons quelques vérifications

bucardo_ctl list herds :
[root@postgresql-centos-master ~]# bucardo_ctl list herds
Herd: alpha DB: test1_master Members: public.pgbench_branches, public.pgbench_tellers, public.pgbench_accounts
Used in syncs: benchdelta
Herd: beta DB: test1_master Members: public.pgbench_history
Used in syncs: benchcopy
bucardo_ctl list syncs :

[root@postgresql-centos-master ~]# bucardo_ctl list syncs
Sync: benchcopy (fullcopy) beta => test1_slave (Active)
Sync: benchdelta (pushdelta) alpha => test1_slave (Active)

bucardo_ctl list dbs :
[root@postgresql-centos-master ~]# bucardo_ctl list dbs
Database: test1_master Status: active Conn: psql -p 5432 -U bucardo -d test1 -h 10.11.12.241
Database: test1_slave Status: active Conn: psql -p 5432 -U bucardo -d test1 -h 10.11.12.242

bucardo_ctl list tables :
[root@postgresql-centos-master ~]# bucardo_ctl list tables
Table: public.pgbench_accounts DB: test1_master PK: aid (int4)

Table: public.pgbench_branches DB: test1_master PK: bid (int4)
Table: public.pgbench_history DB: test1_master PK: none
Table: public.pgbench_tellers DB: test1_master PK: tid (int4)

Et maintenant, devant vos yeux ébahis, comme disait SCL :

bucardo_ctl start :
[root@postgresql-centos-master ~]# bucardo_ctl start
Checking for existing processes
Removing /var/run/bucardo/fullstopbucardo
Removing /var/run/bucardo/fullstopbucardo
Starting Bucardo

Après quelques secondes, le prompt réapparaît et nous pouvons vérifier la présence des démons Bucardo :

[root@postgresql-centos-master ~]# ps -Afw | grep -i Bucardo
postgres 6058 2011 0 22:01 ?   00:00:00 postgres: postgres bucardo 10.11.12.100(1984) idle
root   6529   1 0 23:20 ?   00:00:00 Bucardo Master Control Program v4.5.0. Active syncs: benchcopy,benchdelta
postgres 6530 2011 0 23:20 ?   00:00:00 postgres: bucardo bucardo [local] idle
postgres 6532 2011 0 23:20 ?   00:00:00 postgres: bucardo test1 10.11.12.241(55246) idle
root   6533 6529 0 23:20 ?   00:00:00 Bucardo Controller. Sync "benchdelta" (pushdelta) for source "alpha"
root   6534 6529 0 23:20 ?   00:00:00 Bucardo Controller. Sync "benchcopy" (fullcopy) for source "beta"
postgres 6535 2011 0 23:20 ?   00:00:00 postgres: bucardo bucardo [local] idle
postgres 6536 2011 0 23:20 ?   00:00:00 postgres: bucardo bucardo [local] idle
root   6537 6533 0 23:20 ?   00:00:00 Bucardo Kid. Sync "benchdelta": (pushdelta) "test1_master" ->
"test1_slave"
root   6538 6534 0 23:20 ?   00:00:00 Bucardo Kid. Sync "benchcopy": (fullcopy) "test1_master" -> "test1_slave"
postgres 6539 2011 0 23:20 ?   00:00:00 postgres: bucardo bucardo [local] idle
postgres 6540 2011 0 23:20 ?   00:00:00 postgres: bucardo bucardo [local] idle
postgres 6541 2011 0 23:20 ?   00:00:00 postgres: bucardo test1 10.11.12.241(55248) idle
postgres 6542 2011 0 23:20 ?   00:00:00 postgres: bucardo test1 10.11.12.241(55249) idle
root   6559 6145 0 23:23 pts/0   00:00:00 grep -i Bucardo

et voilà le résultat.....

Test Replication

Afin de vérifier que les choses se passent bien, nous allons commencer
par un simple comptage du nombre de ligne de la table "tellers"
tout d'abord, sur la base master, passer en su posgres :
bash-4.1$ psql -h 10.11.12.241 -p 5432 -d test1 -At -c 'select count(*) from pgbench_tellers;'
10
puis sur la base slave :
bash-4.1$ psql -h 10.11.12.242 -p 5432 -d test1 -At -c 'select count(*) from pgbench_tellers;'
10

encore deux contrôles sur l'une et l'autre base :
bash-4.1$ psql -h 10.11.12.241 -p 5432 -d test1 -At -c 'select * from pgbench_tellers where tid=1;'
1|1|0|
puis
bash-4.1$ psql -h 10.11.12.242 -p 5432 -d test1 -At -c 'select * from pgbench_tellers where tid=1;'
1|1|0|

A présent, on va effectuer une modif sur la base master et vérifier qu'elle se propage bien sur
la base slave répliquée :

la modification tout d'abord :
bash-4.1$ psql -h 10.11.12.241 -p 5432 -U postgres -d test1 -c "update pgbench_tellers set filler='modification' where
tid=1;"

UPDATE 1

puis la vérification :
bash-4.1$ psql -h 10.11.12.242 -p 5432 -d test1 -A -c 'select * from pgbench_tellers where tid=1;'
1|1|modification|

A présent, vu que la table pgbench_history n'a pas de clé primaire, on ne peut pas traiter
les modifications ligne à ligne, ce serait trop long et improductif, c'est pourquoi nous avons
créé un deuxième type de "sync", en "fullcopy" mais ne sera activer que manuellement
ou par script, bien sûr, en cas de besoin, allons-y donc :

Vérification de l'état actuel de cette table à la fois sur le master puis sur le slave :

```
[root@postgresql-centos-master ~]#  psql -h 10.11.12.241 -p 5432 -d test1 -U postgres -c
        'select count(*) from pgbench_history'
 count
-------
     0
(1 ligne)
```

Remarque, avec l'option -At, on affiche que le résultat 0, par la décoration...separateur, nom de colonne
et avec l'option -U, on indique sous quel nom on se connecte, ainsi, pas besoin de faire un "su postgres"
la même chose pour la table sur le slave

```
[root@postgresql-centos-slave ~]# psql -h 10.11.12.242 -p 5432 -d test1 -At -U postgres -c
        'select count(*) from pgbench_history'
0
```

voilà, les deux résultats sont conformes, on va procéder à des modifications sur notre base master,
avec pgbench, réalisons 3 transactions depuis 1 seul client (valeur par défaut si -c n non précisé),
la table pgbench_history va donc s'incrémenter de 3 transactions (valeur par défaut = 10 si -t non précicé)
on peut utiliser l'option -T s où s représente un nombre de secondes durant lesquelles s'effectueront
des transactions.

```
[root@postgresql-centos-master ~]# pgbench -h 10.11.12.241 -p 5432 -U postgres -c 1 -t 3 test1
starting vacuum...end.
transaction type: TPC-B (sort of)
scaling factor: 1
query mode: simple
number of clients: 1
number of transactions per client: 3
number of transactions actually processed: 3/3
tps = 106.598444 (including connections establishing)
tps = 122.209549 (excluding connections establishing)
```

Revérifions notre table pgbench_history sur le master et le slave

```
[root@postgresql-centos-master ~]# psql -h 10.11.12.241 -p 5432 -d test1
        -At -U postgres -c 'select count(*) from pgbench_history'
3
```

puis sur la table dans la base slave :

```
[root@postgresql-centos-master ~]# psql -h 10.11.12.242 -p 5432 -d test1
        -At -U postgres -c 'select count(*) from pgbench_history'
3
```

le résultat est en tout point identique

PETITE EXPERIENCE SUR LES MACHINES :

1) vérification de la table d'historique :

```
[root@postgresql-centos-master ~]# psql -h 10.11.12.242 -p 5432 -d test1  -U postgres -c 'select * from
pgbench_history'
 tid | bid |  aid  | delta |           mtime            | filler
```

```
-----+-----+-------+-------+----------------------------+--------
   9 |   1 | 51373 |   702 | 2013-02-20 16:04:48.259544 |
   8 |   1 | 56722 | -1047 | 2013-02-20 16:04:48.273834 |
  10 |   1 | 38749 |  2626 | 2013-02-20 16:04:48.281483 |
(3 lignes)

[root@postgresql-centos-master ~]# psql -h 10.11.12.241 -p 5432 -d test1  -U postgres -c 'select * from
pgbench_history'
 tid | bid |  aid  | delta |           mtime            | filler
-----+-----+-------+-------+----------------------------+--------
   9 |   1 | 51373 |   702 | 2013-02-20 16:04:48.259544 |
   8 |   1 | 56722 | -1047 | 2013-02-20 16:04:48.273834 |
  10 |   1 | 38749 |  2626 | 2013-02-20 16:04:48.281483 |
(3 lignes)
```

2) modifions la table pgbench_history sur le slave en effectuant des transactions :

```
[root@postgresql-centos-slave ~]# pgbench -h 10.11.12.242 -p 5432 -U postgres -c 1 -t 3 test1
starting vacuum...end.
transaction type: TPC-B (sort of)
scaling factor: 1
query mode: simple
number of clients: 1
number of transactions per client: 3
number of transactions actually processed: 3/3
tps = 193.448543 (including connections establishing)
tps = 241.021933 (excluding connections establishing)
```

3) revérifions les tables depuis la machine master :

```
[root@postgresql-centos-master ~]# psql -h 10.11.12.241 -p 5432 -d test1  -U postgres -c 'select * from
pgbench_history'
 tid | bid |  aid  | delta |           mtime            | filler
-----+-----+-------+-------+----------------------------+--------
   9 |   1 | 51373 |   702 | 2013-02-20 16:04:48.259544 |
   8 |   1 | 56722 | -1047 | 2013-02-20 16:04:48.273834 |
  10 |   1 | 38749 |  2626 | 2013-02-20 16:04:48.281483 |
(3 lignes)

[root@postgresql-centos-master ~]# psql -h 10.11.12.242 -p 5432 -d test1  -U postgres -c 'select * from
pgbench_history'
 tid | bid |  aid  | delta |           mtime            | filler
-----+-----+-------+-------+----------------------------+--------
   9 |   1 | 62825 |  4543 | 2013-02-20 18:14:14.037465 |
   2 |   1 |  2699 | -2000 | 2013-02-20 18:14:14.042645 |
   8 |   1 | 40211 | -3194 | 2013-02-20 18:14:14.046633 |
(3 lignes)
```

on note que les données ont bien été modifiées sur le serveur slave par mes soins.

4) appliquons une série de transaction sur le serveur master (fonctionnement normal en prod) :

```
[root@postgresql-centos-master ~]# pgbench -h 10.11.12.241 -p 5432 -U postgres -c 1 -t 3 test1
starting vacuum...end.
transaction type: TPC-B (sort of)
scaling factor: 1
query mode: simple
number of clients: 1
number of transactions per client: 3
number of transactions actually processed: 3/3
tps = 105.500070 (including connections establishing)
tps = 117.720923 (excluding connections establishing)
```

5) revérifions les tables côté serveur maitre et esclave :

```
[root@postgresql-centos-master ~]# psql -h 10.11.12.241 -p 5432 -d test1  -U postgres -c 'select * from
pgbench_history'
 tid | bid |  aid  | delta |           mtime            | filler
-----+-----+-------+-------+----------------------------+--------
   5 |   1 | 19893 | -1618 | 2013-02-20 18:21:19.922908 |
   4 |   1 | 47195 | -1375 | 2013-02-20 18:21:19.935843 |
   8 |   1 | 28897 | -3165 | 2013-02-20 18:21:19.94218  |
(3 lignes)

[root@postgresql-centos-master ~]# psql -h 10.11.12.242 -p 5432 -d test1  -U postgres -c 'select * from
pgbench_history'
 tid | bid |  aid  | delta |           mtime            | filler
-----+-----+-------+-------+----------------------------+--------
   5 |   1 | 19893 | -1618 | 2013-02-20 18:21:19.922908 |
   4 |   1 | 47195 | -1375 | 2013-02-20 18:21:19.935843 |
   8 |   1 | 28897 | -3165 | 2013-02-20 18:21:19.94218  |
(3 lignes)
```

On constate donc que la réplication complète "fullcopy" a bien eu lieu sur le slave après
les transactions sur le maitre.

Par ailleurs, la commande "bucardo_ctl kick benchcopy " permet d'appliquer la réplication
à tout moment, en voici la preuve :
j'applique des transactions sur le slave

```
[root@postgresql-centos-slave ~]# pgbench -h 10.11.12.242 -p 5432 -U postgres -c 1 -t 3 test1
starting vacuum...end.
transaction type: TPC-B (sort of)
scaling factor: 1
query mode: simple
number of clients: 1
number of transactions per client: 3
number of transactions actually processed: 3/3
tps = 193.448543 (including connections establishing)
tps = 241.021933 (excluding connections establishing)
```

Revérifions nos deux tables à nouveaux :

```
[root@postgresql-centos-master ~]# psql -h 10.11.12.241 -p 5432 -d test1  -U postgres -c 'select * from
pgbench_history'
 tid | bid |  aid  | delta |           mtime            | filler
-----+-----+-------+-------+----------------------------+--------
   5 |   1 | 19893 | -1618 | 2013-02-20 18:21:19.922908 |
   4 |   1 | 47195 | -1375 | 2013-02-20 18:21:19.935843 |
   8 |   1 | 28897 | -3165 | 2013-02-20 18:21:19.94218  |
(3 lignes)

[root@postgresql-centos-master ~]# psql -h 10.11.12.242 -p 5432 -d test1  -U postgres -c 'select * from
pgbench_history'
 tid | bid |  aid  | delta |           mtime            | filler
-----+-----+-------+-------+----------------------------+--------
   2 |   1 | 98270 |  3578 | 2013-02-20 18:32:26.570679 |
   4 |   1 | 27274 | -3110 | 2013-02-20 18:32:26.576447 |
   4 |   1 | 34223 |   279 | 2013-02-20 18:32:26.579758 |
(3 lignes)
```

on constate bien en effet que la modification a bien eu lieu
sur le poste slave, ce qui est normal, la base est accessible de façon tout
à fait ordinaire, mais ce n'est pas son but, évidemment, mais à présent,
je vais appliquer manuellement la commande : "bucardo_ctl kick benchcopy" pour répliquer.

```
[root@postgresql-centos-master ~]# bucardo_ctl kick benchcopy
Kicked sync benchcopy
```

et enfin une dernière vérification de nos tables :

```
[root@postgresql-centos-master ~]# psql -h 10.11.12.241 -p 5432 -d test1  -U postgres -c 'select * from
pgbench_history'
 tid | bid |  aid  | delta |           mtime            | filler
-----+-----+-------+-------+----------------------------+--------
   5 |   1 | 19893 | -1618 | 2013-02-20 18:21:19.922908 |
   4 |   1 | 47195 | -1375 | 2013-02-20 18:21:19.935843 |
   8 |   1 | 28897 | -3165 | 2013-02-20 18:21:19.94218  |
(3 lignes)

[root@postgresql-centos-master ~]# psql -h 10.11.12.242 -p 5432 -d test1  -U postgres -c 'select * from
pgbench_history'
 tid | bid |  aid  | delta |           mtime            | filler
-----+-----+-------+-------+----------------------------+--------
   5 |   1 | 19893 | -1618 | 2013-02-20 18:21:19.922908 |
   4 |   1 | 47195 | -1375 | 2013-02-20 18:21:19.935843 |
   8 |   1 | 28897 | -3165 | 2013-02-20 18:21:19.94218  |
(3 lignes)
```

Et voilà, tout est bien ok. donc, avec un simple script en cron, on pourrait synchroniser
des tables d'un master à un slave ou plusieurs d'ailleurs.

Quelques commandes utiles d'informations :

```
[root@postgresql-centos-master ~]# bucardo_ctl status
Days back: 3  User: bucardo  Database: bucardo  PID of Bucardo MCP: 6529
Name       Type  State PID   Last_good Time  I/U/D Last_bad Time
==========+=====+=====+=====+=========+=====+=====+========+====
benchcopy | F   |idle |19071|19h11m54s|0s   |3/0/3|unknown |
benchdelta| P   |idle |19070|19h31m9s |0s   |7/0/7|unknown |

[root@postgresql-centos-master ~]# bucardo_ctl status benchcopy
Days back: 3  User: bucardo  Database: bucardo
================================================================================
Sync name:            benchcopy
Current state:        idle (PID = 19071)
Type:                 fullcopy
Source herd/database: beta / test1_master
Target database:      test1_slave
Tables in sync:       1
Last good:            19h 12m 33s (time to run: 0s)
Last good time:       Feb 20, 2013 18:40:35  Target: test1_slave
Ins/Upd/Del:          3 / 0 / 3
Last bad:             unknown
PID file:             /var/run/bucardo/bucardo.ctl.sync.benchcopy.pid
PID file created:     Thu Feb 21 13:52:21 2013
Status:               active
Limitdbs:             0
Priority:             0
Checktime:            none
Overdue time:         00:00:00
Expired time:         00:00:00
Stayalive:            yes      Kidsalive: yes
Rebuild index:        0        Do_listen: no
Ping:                 yes      Makedelta: no
Onetimecopy:          0
Custom select:        no
```

```
Post-copy analyze:    yes
Delete method:        delete

[root@postgresql-centos-master ~]# bucardo_ctl status benchdelta
Days back: 3  User: bucardo  Database: bucardo
================================================================================
Sync name:            benchdelta
Current state:        idle (PID = 19070)
Type:                 pushdelta
Source herd/database: alpha / test1_master
Target database:      test1_slave
Tables in sync:       3
Last good:            19h 32m 8s (time to run: 0s)
Last good time:       Feb 20, 2013 18:21:20  Target: test1_slave
Ins/Upd/Del:          7 / 0 / 7
Last bad:             unknown
PID file:             /var/run/bucardo/bucardo.ctl.sync.benchdelta.pid
PID file created:     Thu Feb 21 13:52:21 2013
Status:               active
Limitdbs:             0
Priority:             0
Checktime:            none
Overdue time:         00:00:00
Expired time:         00:00:00
Stayalive:            yes      Kidsalive: yes
Rebuild index:        0        Do_listen: no
Ping:                 yes      Makedelta: no
Onetimecopy:          0

Informations utiles sur les processus en cours :

[root@postgresql-centos-master ~]# ps aux | grep -i ".*postgres.*"
postgres  2011  0.0  0.5 216396  5248 pts/0    S    Feb19   0:07 /usr/bin/postgres -D /var/lib/pgsql/data
postgres  2012  0.0  0.1 179408  1140 ?        Ss   Feb19   0:14 postgres: logger process
postgres  2014  0.0  0.8 216464  8544 ?        Ss   Feb19   1:18 postgres: writer process
postgres  2015  0.0  0.1 216396  1328 ?        Ss   Feb19   1:04 postgres: wal writer process
postgres  2016  0.0  0.1 216672  1692 ?        Ss   Feb19   0:19 postgres: autovacuum launcher process
postgres  2017  0.0  0.1 179672  1352 ?        Ss   Feb19   0:17 postgres: stats collector process
postgres  6530  0.0  0.6 218312  6284 ?        Ss   Feb19   0:01 postgres: bucardo bucardo [local] idle
postgres  6532  0.0  0.5 217764  5548 ?        Ss   Feb19   0:00 postgres: bucardo test1 10.11.12.241(55246) idle
postgres 19072  0.0  0.5 217756  5412 ?        Ss   13:52   0:00 postgres: bucardo bucardo [local] idle
postgres 19073  0.0  0.5 217756  5416 ?        Ss   13:52   0:00 postgres: bucardo bucardo [local] idle
postgres 19076  0.0  0.5 217768  5328 ?        Ss   13:52   0:00 postgres: bucardo bucardo [local] idle
postgres 19077  0.0  0.5 217768  5320 ?        Ss   13:52   0:00 postgres: bucardo bucardo [local] idle
postgres 19078  0.0  0.3 217624  3956 ?        Ss   13:52   0:00 postgres: bucardo test1 10.11.12.241(55297) idle
postgres 19079  0.0  0.3 217624  3952 ?        Ss   13:52   0:00 postgres: bucardo test1 10.11.12.241(55298) idle
root     19307  0.0  0.0 105312   936 pts/0    S+   14:18   0:00 grep -i .*postgres.*

[root@postgresql-centos-master ~]# ps aux | grep -i ".*bucardo.*"
root      6529  0.2  1.6 213776 16920 ?        S    Feb19   5:04 Bucardo Master Control Program v4.5.0. Active syncs:
benchcopy,benchdelta
postgres  6530  0.0  0.6 218312  6284 ?        Ss   Feb19   0:01 postgres: bucardo bucardo [local] idle
postgres  6532  0.0  0.5 217764  5548 ?        Ss   Feb19   0:00 postgres: bucardo test1 10.11.12.241(55246) idle
root     19070  0.0  1.6 213776 16664 ?        S    13:52   0:01 Bucardo Controller. Sync "benchdelta" (pushdelta) for
source "alpha"
root     19071  0.0  1.6 213776 16664 ?        S    13:52   0:01 Bucardo Controller. Sync "benchcopy" (fullcopy) for
source "beta"
postgres 19072  0.0  0.5 217756  5412 ?        Ss   13:52   0:00 postgres: bucardo bucardo [local] idle
postgres 19073  0.0  0.5 217756  5416 ?        Ss   13:52   0:00 postgres: bucardo bucardo [local] idle
root     19074  0.0  1.6 213900 16680 ?        S    13:52   0:00 Bucardo Kid. Sync "benchdelta": (pushdelta)
"test1_master" -> "test1_slave"
root     19075  0.0  1.6 213900 16648 ?        S    13:52   0:00 Bucardo Kid. Sync "benchcopy": (fullcopy) "test1_master"
-> "test1_slave"
```

```
postgres 19076  0.0  0.5 217768  5328 ?        Ss   13:52   0:00 postgres: bucardo bucardo [local] idle
postgres 19077  0.0  0.5 217768  5320 ?        Ss   13:52   0:00 postgres: bucardo bucardo [local] idle
postgres 19078  0.0  0.3 217624  3956 ?        Ss   13:52   0:00 postgres: bucardo test1 10.11.12.241(55297) idle
postgres 19079  0.0  0.3 217624  3952 ?        Ss   13:52   0:00 postgres: bucardo test1 10.11.12.241(55298) idle
root     19309  0.0  0.0 105312   936 pts/0    S+   14:18   0:00 grep -i .*bucardo.*
```

Complétons encore ces informations par quelques commandes utiles concernant
les synchronisations de bucardo :

```
[root@postgresql-centos-master ~]# bucardo_ctl list syncs
Sync: benchcopy   (fullcopy )  beta  =>  test1_slave  (Active)
Sync: benchdelta  (pushdelta)  alpha =>  test1_slave  (Active)

[root@postgresql-centos-master ~]# bucardo_ctl list dbs
Database: test1_master  Status: active  Conn: psql -p 5432 -U bucardo -d test1 -h 10.11.12.241
Database: test1_slave   Status: active  Conn: psql -p 5432 -U bucardo -d test1 -h 10.11.12.242

[root@postgresql-centos-master ~]# bucardo_ctl list dbgroups
Database group: test1  Members: test1_slave

[root@postgresql-centos-master ~]# bucardo_ctl list tables
Table: public.pgbench_accounts  DB: test1_master  PK: aid (int4)
Table: public.pgbench_branches  DB: test1_master  PK: bid (int4)
Table: public.pgbench_history   DB: test1_master  PK: none
Table: public.pgbench_tellers   DB: test1_master  PK: tid (int4)

[root@postgresql-centos-master ~]# bucardo_ctl list sequences
There are no entries in the 'goat' table.

[root@postgresql-centos-master ~]# bucardo_ctl list herds
Herd: alpha  DB: test1_master  Members: public.pgbench_branches, public.pgbench_tellers, public.pgbench_accounts
  Used in syncs: benchdelta
Herd: beta   DB: test1_master  Members: public.pgbench_history
  Used in syncs: benchcopy

[root@postgresql-centos-master ~]# bucardo_ctl list herd alpha --verbose
Herd: alpha  DB: test1_master  Members: public.pgbench_branches, public.pgbench_tellers, public.pgbench_accounts
  Used in syncs: benchdelta

[root@postgresql-centos-master ~]# bucardo_ctl list herd beta --verbose
Herd: beta  DB: test1_master  Members: public.pgbench_history
  Used in syncs: benchcopy
```

MISE AU POINT DE LA REPLICATION DE TYPE "SWAP"
configuration master/master

clonage de la machine virtelle master ci-dessus.
modification du fichier hosts avec le nouveau nom de host, à savoir
postgresql-centos-master-2 à l'adresse 10.11.12.243

puis clonage de la base avec modifications de l'adresse mac
modification du fichier /etc/udev/rules.d/70-persistent-net.rules
suppression du paragraphe concernant eth0 qui correspond à la configuration
originale, puis modification du fichier /etc/sysconfig/network-scripts/ifcfg-eth0, et
notamment l'adresse mac

modifier le nom : nano /etc/sysconfig/network
puis /etc/hosts en ajoutant l'adresse de la machine et les bons noms
puis redémarrer.

si ce n'est déjà fait, stoppons bucardo.
désinstallation de bucardo et postgresql
unlink /..../bucardo (voir dans le répertoire de l'installeur : make uninstall)

yum remove postgresql

Notre futur deuxième master peut à présent être déployé.

[root@postgresql-centos-master-2 ~]# yum install perl-DBI perl-DBD-Pg perl-DBIx-Safe perl-version postgresql-plperl

reparamétrer les fichiers pg_hba.conf et postgresql.conf

bucardo_ctl install

bucardo_ctl show all

ou bucardo_ctl set foo=bar pour modifier certains paramètres,
j'en profite pour changer les champs d'email.
ce qui donne après un bucardo_ctl show all :

```
[root@postgresql-centos-master-2 ~]# bucardo_ctl show all
audit_pid               = 0
autosync_ddl            = newcol
bucardo_current_version   = 4.5.0
bucardo_version         = 4.5.0
ctl_checkabortedkids_time = 30
ctl_checkonkids_time      = 10
ctl_createkid_time        = 0.5
ctl_nothingfound_sleep    = 0.2
ctl_pingtime            = 600
default_email_from        = master-2@postgresql-centos-master-2.exponentielhippy.fr
default_email_host        = 10.11.12.251
default_email_to          = admin@zimbra.exponentielhippy.fr
email_debug_file          =
endsync_sleep           = 1.0
host_safety_check         =
kid_abort_limit         = 3
kid_abort_sleep         = 1
kid_nodeltarows_sleep     = 0.8
kid_nothingfound_sleep    = 0.3
kid_pingtime            = 60
kid_serial_sleep        = 10
log_conflict_details      = 0
log_conflict_file       = bucardo_conflict.log
log_showline            = 0
log_showpid             = 0
log_showtime            = 3
max_delete_clause         = 200
max_select_clause         = 500
mcp_dbproblem_sleep       = 15
mcp_loop_sleep          = 0.1
mcp_pingtime            = 60
piddir                  = /var/run/bucardo
reason_file             = bucardo.restart.reason.log
stats_script_url        = http://www.bucardo.org/
stopfile                = fullstopbucardo
syslog_facility         = LOG_LOCAL1
tcp_keepalives_count      = 0
tcp_keepalives_idle       = 0
tcp_keepalives_interval   = 0
warning_file            = bucardo.warning.log
```

Vérification des différents objets du schéma de bucardo :

```
[root@postgresql-centos-master-2 ~]# bucardo_ctl list
Usage: list <type> [options]
Shows information about items in the internal Bucardo database.
The type is one of: code, db, dbgroup, table, sequence, herd, sync, customcode
For more information, run: $progname help list <type>
[root@postgresql-centos-master-2 ~]# bucardo_ctl list code
There are no entries in the 'db' table.
[root@postgresql-centos-master-2 ~]# bucardo_ctl list db
There are no entries in the 'db' table.
[root@postgresql-centos-master-2 ~]# bucardo_ctl list dbgroup
There are no entries in the 'dbgroup' table.
[root@postgresql-centos-master-2 ~]# bucardo_ctl list table
There are no entries in the 'goat' table.
[root@postgresql-centos-master-2 ~]# bucardo_ctl list sequence
There are no entries in the 'goat' table.
[root@postgresql-centos-master-2 ~]# bucardo_ctl list herd
There are no entries in the 'herd' table.
[root@postgresql-centos-master-2 ~]# bucardo_ctl list sync
There are no entries in the 'sync' table.
[root@postgresql-centos-master-2 ~]# bucardo_ctl list customcode
There are no entries in the 'db' table.
```

Tout est bien vide, nous allons pouvoir paramétrer notre réplicataion
de type "swap", c'est à dire en master/master, nous allons choisir comme base
source ou left la base master-2 située à l'adresse 10.11.12.243, et comme base
target ou right la base master à l'adresse 10.11.12.241, que nous utilisions juste
avant pour notre réplication "pushdelta" ou master/slave.

Créons d'abord nos db sur le serveur master-2
puis indiquons les 'db' qui constituent notre infrastructure master/master :

```
[root@postgresql-centos-master-2 ~]# bucardo_ctl add database test1 name=test1_master2 host=10.11.12.243
port=5432
Added database "test1_master2"
[root@postgresql-centos-master-2 ~]# bucardo_ctl add database test1 name=test1_master host=10.11.12.241 port=5432
Added database "test1_master"
[root@postgresql-centos-master-2 ~]# bucardo_ctl list db
Database: test1_master   Status: active  Conn: psql -p 5432 -U bucardo -d test1 -h 10.11.12.241
Database: test1_master2  Status: active  Conn: psql -p 5432 -U bucardo -d test1 -h 10.11.12.243
```

REMARQUE : attention aux caractères interdits dans le nommage des 'database', pas de tiret par exemple

Puis on ajoute les tables nécessaires de notre base test1 à un nouveau groupement (herd) ou une nouvelle
harde nommé "gamma", sans la table pgbench_history vu que je vais réaliser une configuration
de type "swap", c'est à dire master/master, et qu'elle ne possède pas de clé primaire :

Créons notre harde avec ses tables et sa database source

```
[root@postgresql-centos-master-2 ~]# bucardo_ctl add all table -T pgbench_history db=test1_master2
        standard_conflict=source herd=gamma --verbose
Creating herd: gamma
New tables:
  public.pgbench_accounts
  public.pgbench_branches
  public.pgbench_tellers
New tables added: 3
Already added: 0
```

Avant de lancer la création de la synchronisation de type swap (master/master), bien gérer les
conflits en précisant dans le champ standard_conflicts de la table goat de toutes les lignes ce que l'on
souhaite, soit source, target, etc.....
faire un update goat set standard_conflict='target' le cas échéant car je veux que le host postgresql-centos-master soit le

'super-maître'.

[root@postgresql-centos-master-2 ~]# bucardo_ctl add sync benchswap source=gamma targetdb=test1_master type=swap
Added sync "benchswap"
[root@postgresql-centos-master-2 ~]# bucardo_ctl start
Checking for existing processes
Removing /var/run/bucardo/fullstopbucardo
Removing /var/run/bucardo/fullstopbucardo
Starting Bucardo

Nous allons ajouter la recopie de la table pgbench_history de test1_master à test1_master2, pour
cela, on va compléter la sync benchcopy qui comprend déjà la recopie vers la base test1_slave
sur une autre machine. Sur notre premier master, nous allons créer un dbgroup avec les deux tables test1_slave et test1_master2 afin que ce soit notre cible pour la recopie de la table history, aussi bien pour la réplication maître/maître que maître/esclave.

[root@postgresql-centos-master ~]# bucardo_ctl stop
Creating /var/run/bucardo/fullstopbucardo ... Done

Avant cela, on identifie la table
[root@postgresql-centos-master ~]# bucardo_ctl add db test1 name=test1_master2 host=10.11.12.243 user=bucardo port=5432
Added database "test1_master2"

[root@postgresql-centos-master ~]# bucardo_ctl add dbgroup test1_history test1_slave test1_master2
Added database "test1_slave" to group "test1_history"
Added database "test1_master2" to group "test1_history"
Added database group "test1_history"

[root@postgresql-centos-master ~]# bucardo_ctl deactivate benchcopy
Deactivating sync benchcopy
[root@postgresql-centos-master ~]# bucardo_ctl remove sync benchcopy
Removed database "benchcopy"
Note: table triggers (if any) are not automatically removed!
On définit à nouveau notre 'fullcopy' en prenant pour cible le dbgroup contenant la table pgbench_history de master-2 et slave, afin que cette table soit identique à celle sur master, qui je le rappelle, est notre 'super-maître'.

[root@postgresql-centos-master ~]# bucardo_ctl add sync benchcopy source=beta targetgroup=test1_history type=fullcopy
Added sync "benchcopy"

[root@postgresql-centos-master ~]# bucardo_ctl start
Checking for existing processes
Removing /var/run/bucardo/fullstopbucardo
Starting Bucardo
[root@postgresql-centos-master ~]# bucardo_ctl kick benchcopy
Kicked sync benchcopy

après cette dernière commande, on a normalement synchronisé nos history
sur les bases slave et master2 avec pour source master.

En résumé, le duo master/slave fonctionne correctement, de plus,
chaque opération sur le master est répliquée instantanément
sur le master-2 automatiquement grâce au mécanisme master/master
mis en place sur le master-2 et vice et versa.

Appliquons quelques vérifications :
tout d'abord, envoyons une rafale de transactions sur le master depuis
une autre machine que l'un des 3 impliquées dans notre cluster

root@debian:~# ./pgbench -h 10.11.12.241 -p 5432 -U postgres -c 1 -t 10 test1
starting vacuum...end.

transaction type: TPC-B (sort of)
scaling factor: 1
query mode: simple
number of clients: 1
number of threads: 1
number of transactions per client: 10
number of transactions actually processed: 10/10
tps = 128.710068 (including connections establishing)
tps = 135.865873 (excluding connections establishing)

Après vérification des tables sur les 3 hosts, master, slave et master-2,
on constate que la réplication a bien eu lieu.
Pour l'heure, il n'est pas tenu compte du délai de propagation, mais il faudra y venir.

A présent, nous allons envoyer une rafale de transactions sur le master-2 depuis
une autre machine, comme ci-dessus, et en toute logique, la réplication se fera
correctement sur le host master.

Allons-y :
root@debian:~# ./pgbench -h 10.11.12.243 -p 5432 -U postgres -c 1 -t 10 test1
starting vacuum...end.
transaction type: TPC-B (sort of)
scaling factor: 1
query mode: simple
number of clients: 1
number of threads: 1
number of transactions per client: 10
number of transactions actually processed: 10/10
tps = 136.286201 (including connections establishing)
tps = 144.446049 (excluding connections establishing)

En effet, comme prévu, les tables master-2 et master ont été répliquées,
la table pgbench_history de master-2 a bien été ajusté suite à la rafale,
mais la table slave n'a subit aucune modification, ce qui est conforme vu
que ce sont les modifications sur master qui engendrent les pushdelta sur
le slave, et par ailleurs, ce sont les modifications sur le master qui engendrent
une modification de la table pgbench_history, c'est pourquoi seule la table pgbench_history
de la table master-2 est actualisée.

Mais si l'on veut pouvoir faire de la redondance de charge avec le master/master, il faut
créer un autre slave spécifique pour le master-2.

Nous avons bien nos deux synchronisations, swap et fullcopy, on peut redémarrer
bucardo.
[root@postgresql-centos-master-2 ~]# bucardo_ctl start
Checking for existing processes
Removing /var/run/bucardo/fullstopbucardo
Starting Bucardo

[root@postgresql-centos-master-2 ~]# ps -Af | grep -i bucardo
postgres 1586 1578 0 10:34 ?   00:00:00 postgres: bucardo test1 postgresql-centos-
master.exponentielhippy.fr(57897) idle
postgres 1587 1578 0 10:34 ?   00:00:00 postgres: bucardo test1 postgresql-centos-
master.exponentielhippy.fr(57901) idle
postgres 1637 1578 0 10:41 ?   00:00:00 postgres: postgres bucardo 10.11.12.100(4437) idle
root   1833   1 0 11:06 ?   00:00:00 Bucardo Master Control Program v4.5.0. Active syncs:
benchcopyeta,benchswapgamma
postgres 1834 1578 0 11:06 ?   00:00:00 postgres: bucardo bucardo [local] idle
postgres 1836 1578 0 11:06 ?   00:00:00 postgres: bucardo test1 postgresql-centos-master-2(34900) idle
root   1837 1833 0 11:06 ?   00:00:00 Bucardo Controller. Sync "benchcopyeta" (fullcopy) for source "eta"
root   1838 1833 0 11:06 ?   00:00:00 Bucardo Controller. Sync "benchswapgamma" (swap) for source "gamma"
postgres 1839 1578 0 11:06 ?   00:00:00 postgres: bucardo bucardo [local] idle
postgres 1840 1578 0 11:06 ?   00:00:00 postgres: bucardo bucardo [local] idle

```
root      1841  1837  0 11:06 ?        00:00:00 Bucardo Kid. Sync "benchcopyeta": (fullcopy) "test1_master2" ->
"test1_master"
root      1842  1838  0 11:06 ?        00:00:00 Bucardo Kid. Sync "benchswapgamma": (swap) "test1_master2" ->
"test1_master"
postgres  1843  1578  0 11:06 ?        00:00:00 postgres: bucardo bucardo [local] idle
postgres  1844  1578  0 11:06 ?        00:00:00 postgres: bucardo bucardo [local] idle
postgres  1845  1578  0 11:06 ?        00:00:00 postgres: bucardo test1 postgresql-centos-master-2(34903) idle
postgres  1846  1578  0 11:06 ?        00:00:00 postgres: bucardo test1 postgresql-centos-master-2(34904) idle
root      1847  1837  0 11:06 ?        00:00:00 Bucardo Kid. Sync "benchcopyeta": (fullcopy) "test1_master2" ->
"test1_slave"
postgres  1848  1578  0 11:06 ?        00:00:00 postgres: bucardo bucardo [local] idle
postgres  1849  1578  0 11:06 ?        00:00:00 postgres: bucardo test1 postgresql-centos-master-2(34907) idle
root      1853  1552  0 11:06 pts/0    00:00:00 grep -i bucardo
```

appliquons manuellement les deux sync.

```
[root@postgresql-centos-master-2 ~]# bucardo_ctl kick benchcopyeta
Kicked sync benchcopyeta

[root@postgresql-centos-master-2 ~]# bucardo_ctl kick benchswapgamma
Kicked sync benchswapgamma
```

Nous allons renommer les synchronisations du host master, dont la seule raison est une cohérence dans le nommage des différents objets de la configuration :
plutôt que "benchcopy" sur le master, appelons cette sync "benchcopybeta" puisqu'il s'agit de la harde beta en mode fullcopy qui comprend les tables d'historiques pgbench_history des hosts master-2 et slave.
enfin, plutôt que "benchdelta" sur le master, appelons cette sync "benchdeltaalpha" puisqu'il s'agit de la harde alpha qui comprend toutes les tables pgbench_xxx sauf history du host master vers la cible slave.
voyez plutôt :

```
[root@postgresql-centos-master ~]# bucardo_ctl list herd
Herd: alpha  DB: test1_master  Members: public.pgbench_branches, public.pgbench_tellers, public.pgbench_accounts
  Used in syncs: benchdeltaalpha
Herd: beta   DB: test1_master  Members: public.pgbench_history
  Used in syncs: benchcopybeta



[root@postgresql-centos-master ~]# bucardo_ctl list sync
Sync: benchcopy   (fullcopy )  beta  =>  test1_history  (Active)
Sync: benchdelta  (pushdelta)  alpha =>  test1_slave    (Active)
[root@postgresql-centos-master ~]# bucardo_ctl remove sync benchdelta
Cannot remove active sync "benchdelta": please deactivate it first
[root@postgresql-centos-master ~]# bucardo_ctl deactivate benchdelta
Deactivating sync benchdelta
[root@postgresql-centos-master ~]# bucardo_ctl deactivate benchcopy
Deactivating sync benchcopy
[root@postgresql-centos-master ~]# bucardo_ctl remove sync benchdelta
Removed database "benchdelta"
Note: table triggers (if any) are not automatically removed!
[root@postgresql-centos-master ~]# bucardo_ctl remove sync benchcopy
Removed database "benchcopy"
Note: table triggers (if any) are not automatically removed!

[root@postgresql-centos-master ~]# bucardo_ctl list sync
There are no entries in the 'sync' table.
```

toutes les synchronisations précédentes ont été supprimées

```
[root@postgresql-centos-master ~]# bucardo_ctl add sync benchdeltaalpha source=alpha targetdb=test1_slave
type=pushdelta
Added sync "benchdeltaalpha"

[root@postgresql-centos-master ~]# bucardo_ctl list dbgroup
Database group: test1_history  Members: test1_master2, test1_slave
```

```
[root@postgresql-centos-master ~]# bucardo_ctl add sync benchcopybeta source=beta targetgroup=test1_history
type=fullcopy
Added sync "benchcopybeta"

[root@postgresql-centos-master ~]# bucardo_ctl list sync
Sync: benchcopybeta    (fullcopy )  beta  =>  test1_history  (Active)
Sync: benchdeltaalpha  (pushdelta)  alpha =>  test1_slave    (Active)

[root@postgresql-centos-master ~]# bucardo_ctl start
Checking for existing processes
Removing /var/run/bucardo/fullstopbucardo
Starting Bucardo
```

Voilà pour le renommage

Rappelons la configuration : 2 maîtres et 1 esclave, une réplication de type master/master en mode trigger avec Bucardo 4.5 et une deux réplications de type master/slave.
configuration master/master : (mode swap)
postgresql-centos-master(10.11.12.241) / postgresql-centos-master-2(10.11.12.243)

configurations master/slave :
postgresql-centos-master(10.11.12.241) / postgresql-centos-slave(10.11.12.242) (modes pushdelta et copy)

Rappelons également que bucardo s'appuie sur des scripts perl, et le langage plpgsql, et que les packages perl DBI pour perl DataBase Interface sont installés sur les configs bucardo, c'est à dire les deux masters.

Une bonne pratique quant à l'utilisation de bucardo, est le nettoyage (purge) des tables des deltas et des queries, à savoir dans la DB bucardo et dans la base test1 en ce qui me concerne. Ce qui donne sur ma configuration pour les deltas sur le schéma bucardo de la DB test1 sur les deux master et master2 :

```
root@debian:~# psql -X -q -h 10.11.12.241 -p 5432 -U bucardo -d test1 -c "select bucardo_purge_delta('10
minutes'::interval)"

                bucardo_purge_delta
-------------------------------------------------------------------------------
 Rows deleted from bucardo_delta: 1772961 Rows deleted from bucardo_track: 2945904
(1 row)

root@debian:~# psql -X -q -h 10.11.12.243 -p 5432 -U bucardo -d test1 -c "select bucardo_purge_delta('10
minutes'::interval)"

                bucardo_purge_delta
-------------------------------------------------------------------------------
 Rows deleted from bucardo_delta: 2508744 Rows deleted from bucardo_track: 4617450
(1 row)
```

A présent, occupons-nous des tables des queries sur les deux master et master2 dans la DB bucardo, schéma bucardo bien sûr :

```
root@debian:~# psql -X -q -h 10.11.12.241 -p 5432 -U bucardo -d bucardo -c "select bucardo_purge_q_table('5
minutes'::interval)"

NOTICE:  Rows left in q table: 0
 bucardo_purge_q_table
-----------------------
                  2278
(1 row)

root@debian:~# psql -X -q -h 10.11.12.243 -p 5432 -U bucardo -d bucardo -c "select bucardo_purge_q_table('5
minutes'::interval)"
```

```
NOTICE:  Rows left in q table: 0
 bucardo_purge_q_table
-----------------------
                  2146
(1 row)
```

bien entendu, vu qu'il faut régulièrement effectuer cette opération, il suffit d'ajouter ces commandes dans les tables cron des serveurs eux-mêmes, plutôt que sur une autre machine, comme je l'ai fait, cela dit, c'est possible. ah, j'oubliais, mettre en place sur les trois serveurs une actualisation par le protocole ntp des horloges des machines le plus sûrement possible, et aussi souvent que nécessaire, au moins une fois par jour, voir 3 ou 4 fois, sur un serveur de temps le plus fiable possible, bien entendu.

Par ailleurs, une fois les traitements de replication effectués, ces traitements sont stockés dans la table bucardo, schema freezer, à savoir une table par jour d'exploitation, aussi est il conseillé de faire le ménage régulièrement, sauf si l'on souhaite conserver toutes les modifs effectuées, mais ce schéma peut alors devenir très volumineux, et causer des ralentissements.

on peut par exemple executer journalièrement par cron interposé le bucardo_purge.sh suivant :

```
#!/bin/bash
echo ========BEGIN DAILY PURGE ============== >> /var/log/bucardo-purge
date >> /var/log/bucardo-purge
echo ======================================== >> /var/log/bucardo-purge
PGPASSWORD=`grep dbpass /usr/local/bin/bucardo_ctl | awk -F '=' '{print $2;}'`
export PGPASSWORD
/usr/bin/psql -U bucardo -d bucardo < /usr/local/share/bucardo/bucardo_daily_purge.sql >> /var/log/bucardo-purge
2>&1
DROP_OLD_CHILD_Q="DROP TABLE freezer.child_q_"
DROP_OLD_CHILD_Q+=`/bin/date --date 'last Sunday' "+%Y%m%d"`
DROP_OLD_CHILD_Q+=';'
echo $DROP_OLD_CHILD_Q | /usr/bin/psql -U bucardo -d bucardo >> /var/log/bucardo-purge 2>&1
echo =========END DAILY PURGE ============== >> /var/log/bucardo-purge
```

Ainsi, vous pourrez garder une semaine de modification par exemple, jusqu'au dernier dimanche. Tout est possible.

et voici donc la commande sql pour finaliser bucardo_daily_purge.sql :

```
DELETE FROM q WHERE (started < now() + '1 day ago'::interval OR \
        ended < now() + '1 day ago'::interval OR aborted < now() + \
        '1 day ago'::interval OR cdate < now() + '1 day ago'::interval) AND \
        (ended IS NULL OR aborted IS NULL);
```