



L'expertise PostgreSQL



Auteur : Guillaume Lelarge <guillaume.lelarge@dalibo.com>
Version : 1.0 du 19 juin 2007
Type : Note Technique
Licence : GNU Free Documentation License Copyright (c) 2007 dalibo S.A.R.L.
Il est autorisé de copier, distribuer et/ou modifier ce document sous
dans les termes de la “GNU Free Documentation Licence, Version
1.2”, ou toute version ultérieure publiée par la “Free Software Founda-
tion”; Le texte complet de la licence est disponible sur Internet à
l’adresse suivante : <http://www.gnu.org/copyleft/fdl.html>

Résumé

Présentation de trois applications de concentration de connexions à PostgreSQL

Historique des révisions

Version	Date	Libellé
1.0	19/06/2007	Version initiale

Table des matières

1	Introduction	5
2	pgpool	6
2.1	Compilation et installation	6
2.2	Configuration	6
2.3	Utilisation	7
2.4	Pour aller plus loin	7
3	pgpool-II	8
3.1	Compilation et installation	8
3.2	Configuration	9
3.3	Utilisation	9
3.4	Pour aller plus loin	9
4	pgbouncer	10
4.1	Compilation et installation	10
4.2	Configuration	10
4.3	Utilisation	11
4.4	Pour aller plus loin	11
5	Comparatifs sur un test très simple	12
6	Conclusion	14

1

Introduction

Le temps de mise en place de la connexion au serveur PostgreSQL est souvent important. Dans le cas de serveurs web ou d'outils qui se connectent très fréquemment à la base, un système de spooling de connexions est souvent une bonne solution. Il en existe trois principales.

2

pgpool

Il s'agit du plus ancien outil de gestion des connexions. Son installation est très simple, sa configuration l'est aussi.

2.1 Compilation et installation

Récupérer la dernière version :

```
wget http://pgfoundry.org/frs/download.php/980/pgpool-3.1.1.tar.gz
```

La déballer :

```
tar xvfz pgpool-3.1.1.tar.gz
```

Configurer la compilation et compiler l'outil :

```
cd pgpool-3.1.1
./configure
make
```

Enfin, installer l'outil et ses fichiers de configuration :

```
make install
```

2.2 Configuration

Commencer par copier les fichiers de configuration exemple pour avoir le bon nom :

```
cp pgpool.conf.sample pgpool.conf
cp pool_hba.conf.sample pool_hba.conf
```

Si vous utilisez une connexion TCP/IP (et non pas une connexion par socket Unix), il faut préciser le nom de l'hôte PostgreSQL. Pour cela, la variable `backend_host_name` doit être renseignée.

Voici les quelques variables intéressantes :

- `port`, port d'écoute de `pgpool`... il est possible d'utiliser le port standard de PostgreSQL, le 5432 ;
- `secondary_backend_host_name`, intéressant si vous voulez utiliser `pgpool` dans un contexte de répartition de charges ;
- `num_init_children`, nombre de connexions déjà prêtes ;
- `max_pool`, nombre de pools de connexions pour un processus fils.
- `load_balance_mode`, à activer si vous voulez utiliser `pgpool` dans un contexte de répartition de charges ;

2.3 Utilisation

Pour l'exécuter, il suffit simplement de le lancer :

```
pgpool
```

2.4 Pour aller plus loin

<http://pgpool.projects.postgresql.org/>

3

pgpool-II

C'est une nouvelle version de pgpool, entièrement recodée. Elle comprend de nombreuses améliorations sur les deux fonctionnalités les plus en vogue : la répartition de charge et la réplication. Le spooler de connexions est toujours disponible. Son installation est aussi simple mais sa configuration est plus complexe, malheureusement sans gain en performance.

3.1 Compilation et installation

Récupérer la dernière version :

```
wget http://pgfoundry.org/frs/download.php/1398/pgpool-II-1.1.1.tar.gz
```

La déballer :

```
tar xvfz pgpool-II-1.1.1.tar.gz
```

Configurer la compilation et compiler l'outil :

```
cd pgpool-II-1.1.1
./configure
make
```

Enfin, installer l'outil et ses fichiers de configuration :

```
make install
```

3.2 Configuration

Commencer par copier les fichiers de configuration exemple pour avoir le bon nom :

```
cp pcp.conf.sample pcp.conf
cp pgpool.conf.sample pgpool.conf
cp pool_hba.conf.sample pool_hba.conf
```

Là-aussi, si vous utilisez une connexion TCP/IP (et non pas une connexion par socket Unix), il faut préciser le nom de l'hôte PostgreSQL. Pour cela, la variables suivantes doivent être renseignées :

- backend_hostname0, nom d'hôte du serveur ;
- backend_port0, numéro de port ;
- backend_weight0, poids.

Il existe beaucoup plus de variables intéressantes. Notamment, si on laisse de côté les variables communes avec pgpool, les variables de trace permettent de mieux déboguer en cas de problème : log_statement, log_connection, log_hostname.

3.3 Utilisation

Pour l'exécuter, il suffit simplement de le lancer :

```
pgpool
```

3.4 Pour aller plus loin

<http://pgpool.projects.postgresql.org/pgpool-II/en/>

4

pgbouncer

Il fait partie des outils proposés tout dernièrement, sous licence BSD, par Skype.

4.1 Compilation et installation

Récupérer la dernière version :

```
wget http://pgfoundry.org/frs/download.php/1399/pgbouncer-1.0.8.tgz
```

La déballer :

```
tar xvfz pgbouncer-1.0.8.tgz
```

Configurer la compilation et compiler l'outil :

```
cd pgbouncer-1.0.8
./configure
make
```

Enfin, installer l'outil et ses fichiers de configuration :

```
make install
```

4.2 Configuration

Les utilisateurs autorisés doivent être listés dans le fichier `userlist.txt` qui se compose de plusieurs lignes à ce format :

“utilisateur” : “mot de passe”

Un fichier de configuration d'exemple peut aussi être copié :

```
mkdir etc
cp share/doc/pgbouncer/pgbouncer.ini etc
```

Toutes les bases concernées par le spooling doivent être précisées. Il est même nécessaire d'indiquer la phrase de connexion. Il est donc possible de changer le serveur et le port de connexion, mais aussi l'utilisateur, voire même la base, ce qui décuple les possibilités d'utilisation de cet outil.

Voici un exemple d'une base ajoutée :

```
tests = host=127.0.0.1 dbname=tests
```

Un conseil : ne conservez les traces que si nécessaire. Sinon désactivez-les :

- désactiver “logfile = pgbouncer.log” en ajoutant un point-virgule devant la ligne ;
- initialiser log_connections et log_connections à 0.

4.3 Utilisation

Pour l'exécuter, il suffit simplement de le lancer :

```
pgbouncer -d etc/pgbouncer.ini
```

4.4 Pour aller plus loin

<https://developer.skype.com/SkypeGarage/DbProjects/PgBouncer>

5

Comparatifs sur un test *très* simple

Le script de création de la base de tests :

```
guillaume@laptop :~/tests_pgpool$ cat base.sh
#!/bin/sh

createdb tests

psql tests <<FIN_PSQL
BEGIN;
CREATE TABLE t1 (id serial, i integer);
INSERT INTO t1 (i) SELECT valeur_de_i FROM generate_series(10000, 1, -
1) AS valeur_de_i;
COMMIT;
FIN_PSQL
```

Ce script crée une base nommée tests. Elle lui ajoute une table, t1, qu'elle remplit de valeurs entières.

Le script de tests :

```
guillaume@laptop :~/tests_pgpool$ cat tests.sh
#!/bin/sh

export PGPORT=${1 :5432}

for index in `seq 1 10000`
do
  psql -c "SELECT * FROM t1 WHERE id=$index" tests >/dev/null
done
```

Ce script va exécuter 10000 fois l'outil psql. Cela permet ainsi de tester surtout la rapidité de la connexion au serveur PostgreSQL.

Voici le test qui correspond à une connexion directe au serveur PostgreSQL :

```
guillaume@laptop :~/tests_pgpool$ time ./tests.sh 5432
real    2m24.126s
user    0m31.802s
sys     0m29.442s
```

Voici le test qui passe par pgpool :

```
guillaume@laptop :~/tests_pgpool$ time ./tests.sh 9999
real    1m41.227s
user    0m31.650s
sys     0m26.822s
```

pgpool-II a exactement les mêmes performances que pgpool. Le gain est déjà important (30%) avec la configuration de base. En passant num_init_children à 64, voire même à 128, on ne gagne rien sur ce test, ce qui n'est pas très étonnant. Même remarque lorsqu'on passe max_spool de 4 à 16.

Voici le test qui passe par pgbouncer :

```
guillaume@laptop :~/tests_pgpool$ time ./tests.sh 6000
real    1m34.954s
user    0m29.902s
sys     0m25.438s
```

La configuration a désactivé les traces. Le gain est encore supérieur.

6

Conclusion

pgpool est le premier outil proposant cette fonctionnalité pour PostgreSQL. Son ancienneté est à la fois un avantage et un inconvénient : il a fait ses preuves mais accuse aussi son âge. pgpool-II est la deuxième génération qui répond précisément à ce problème. De nombreuses améliorations sont disponibles pour un grand nombre de fonctionnalités. Malheureusement, le spooling de connexions ne semble pas faire partie des fonctionnalités améliorées.

pgbouncer est certainement un projet très prometteur, autant pour le spooling de connexions que pour les fonctionnalités de type répartition de charge et réplication. Cependant, il lui reste encore à faire ses preuves.