

# Changer une base PostgreSQL de « tablespace »

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 janvier 2012

<https://www.bortzmeyer.org/postgresql-changer-tablespace.html>

---

Un des principaux mécanismes de gestion de l'espace disque dans PostgreSQL est le *"tablespace"* <<http://www.postgresql.org/docs/8.4/interactive/manage-ag-tablespaces.html>>. Un *"tablespace"* est un répertoire où on place des données du SGBD. Mais, si on change d'avis, comment changer une base de *"tablespace"* ?

La *"tablespace"* par défaut d'une base se déclare à la création :

```
% createdb --tablespace grosdisque experience
```

Ici, la base *"experience"* est créée sur le *"tablespace"* *"grosdisque"* (créé précédemment par CREATE TABLESPACE). On peut afficher les *"tablespaces"* par défaut des bases avec le catalogue système de PostgreSQL :

```
=> SELECT datname AS database, spcname AS tablespace, spclocation AS directory
       FROM pg_database INNER JOIN pg_tablespace
       ON pg_tablespace.oid = pg_database.dattablespace;
 database | tablespace | directory
-----+-----+-----
 templat1 | pg_default |
  essais  | pg_default |
 experience | grosdisque | /some/where/big
 ...
```

Et si on s'est trompé, si on a oublié de mettre la base sur le bon *"tablespace"*, si la base a grossi <<https://www.bortzmeyer.org/postgresql-taille-bases.html>> au delà de ce qui était prévu ? Depuis la version 8.4 de PostgreSQL, il existe un moyen simple, la commande ALTER DATABASE :

```
essais=> ALTER DATABASE experience SET TABLESPACE autreendroit;
ERROR:  database "experience" is being accessed by other users
DETAIL:  There are 1 other session(s) using the database.
```

(Ah oui, il faut qu'aucune session n'accède à la table, ce qui peut être contraignant.)

```
essais=> ALTER DATABASE experience SET TABLESPACE autreendroit;
ALTER DATABASE
```

Et si on gère une base dans une version antérieure de PostgreSQL, et qu'on ne peut pas migrer ? Rien n'est perdu. Il y a bien sûr la solution « bourrin » d'une commande `pg_dump`, d'une re-création de la base sur le nouveau *"tablespace"*, puis d'un `pg_restore`. C'est très lent, et cela empêche d'accéder à la base en écriture pendant ce temps.

Une solution plus astucieuse est documentée par Lode <<http://blog.lodeblomme.be/2008/03/15/move-a-postgresql-database-to-a-different-tablespace/>> : elle utilise le fait que le *"tablespace"* n'est pas forcément par base mais peut être configuré par table et qu'il est possible, même avant la version 8.4 de PostgreSQL, de changer une table de *"tablespace"*. Le principe est donc :

```
essais=> ALTER DATABASE experience SET default_tablespace = autreendroit;
```

(Cette première commande changera le tablespace pour les *\*futures\** tables.)

```
essais=> ALTER TABLE premiere_table SET TABLESPACE autreendroit;
essais=> ALTER TABLE deuxieme_table SET TABLESPACE autreendroit;
...
```

Oui, il faut le faire pour toutes les tables, et pour les index également. Ce n'est pas très pratique. Lode automatisait avec PHP, je préfère le faire avec le shell :

```
% psql --tuples-only -c 'SELECT tablename FROM pg_tables' experience > tmp/tables
% for table in $(cat tmp/tables); do
  echo $table
  psql -c "ALTER TABLE $table SET TABLESPACE autreendroit;" experience
done
```

Et même chose avec les index :

```
% psql --tuples-only -c 'SELECT indexname FROM pg_indexes' experience > tmp/indexes
% for idx in $(cat tmp/indexes); do
  echo $idx
  psql -c "ALTER INDEX $idx SET TABLESPACE autreendroit;" experience
done
```

C'est bien plus rapide que sauvegarder/restaurer. Rappelez-vous bien que cela ne change pas le *"tablespace"* de la base. Il apparaîtra toujours comme l'ancien. Mais toutes les données seront bien dans le nouveau *"tablespace"*.

Après, à vous de voir si cela ne serait pas plus simple de migrer vers un PostgreSQL  $\geq$  8.4. Mais les grosses bases de données sont souvent des choses fragiles, avec lesquelles on ne peut pas jouer comme on veut. J'ai récemment utilisé la vieille méthode pour une base qu'il aurait été risqué de migrer.