DALIBO

L'Expertise PostgreSQL

Introduction à la réplication avec PostgreSQL 9.0

Table des matières

Réplication avec PostgreSQL 9.0

1 À propos des auteurs...

» Auteur : Guillaume Lelarge
» Société : DALIBO
» Date : Janvier 2011
» URL :
https://support.dalibo.com/kb/conferences/replication_postgresql_9.0_rapide/

2 Licence

- Licence Creative Common BY-NC-SA
- 3 contraintes de partage :
  - Citer la source (dalibo)
  - Pas d'utilisation commerciale
  - Partager sous licence BY-NC-SA

3 Journaux de transactions

- Contiennent toutes les modifications des fichiers de la base
- Par instance (*ie*, toutes les bases)
- Informations de bas niveau
  - Bloc par bloc, fichier par fichier
  - Ne contient pas la requête elle-même
- Déjà utilisé en cas de crash du serveur
  - Rejeu des transactions non synchronisées sur les fichiers

Chaque transaction, implicite ou explicite, réalisant des modifications sur la structure ou les données d'une base est tracée dans les journaux de transactions. Ces derniers contiennent des informations d'assez bas-niveau, comme les blocs modifiés sur un fichier suite, par exemple, à un UPDATE. La requête elle-même n'apparaît jamais. Les journaux de transactions sont valables pour toutes les bases de données de l'instance.

Les journaux de transactions sont déjà utilisés en cas de *crash* du serveur. Lors du redémarrage, PostgreSQL rejoue les transactions qui n'auraient pas été synchronisées sur les fichiers de données.

Comme toutes les modifications sont disponibles dans les journaux de transactions et que PostgreSQL sait rejouer les transactions à partir des journaux, il suffit d'archiver les journaux sur une certaine période de temps pour pouvoir les rejouer.

3.1 PITR

- Point In Time Recovery
- Possibilité de rejouer
  - Tous les journaux de transactions
  - Jusqu'à un certain point dans le temps
  - Jusqu'à un certain identifiant de transaction
- Rejeu à partir d'une sauvegarde des fichiers à un instant *t*
- Disponible à partir de PostgreSQL 8.0

La technologie PITR est disponible depuis la version 8.0. Cette dernière permet le rejeu de tous les journaux de transactions préalablement archivés ou tous les journaux jusqu'à un certain point dans le temps, ou encore tous les journaux jusqu'à un certain identifiant de transaction.

Pour cela, il est nécessaire d'avoir une sauvegarde des fichiers de l'instance (réalisée à chaud une fois l'archivage activé) et des journaux archivés depuis cette sauvegarde.

3.2 Warm Standby

- Esclave mis à jour en permanence
- Fichier par fichier
  - Fichier == un journal de transactions
- Esclave non disponible pendant la restauration
- En 8.3, ajout de l'outil pg_standby

L'idée du serveur en Warm Standby est de rejouer en permanence les journaux de transactions archivés. Autrement dit, quand le serveur maître a terminé de travailler sur un journal de

transactions, il l'archive sur un deuxième serveur où il sera récupéré par le serveur PostgreSQL esclave qui le rejouera dès la fin de la copie.

C'est un système de réplication complet. Mais deux gros inconvénients apparaissent : le délai de prise en compte des modifications dépend de l'activité du serveur maître (plus ce dernier sera actif, plus il enverra rapidement un journal de transactions, plus le serveur esclave sera à jour) et le serveur esclave n'est pas disponible, y compris pour des requêtes en lecture seule.

Plutôt que d'avoir à écrire son propre outil, la version 8.3 propose dans les modules contrib un outil appelé pg_standby. De même, Skype propose son propre outil appelé walmgr. Un dernier outil permet aussi de faciliter la restauration : pitrtools.

4 Version 9.0

- Comble deux grosses lacunes des anciennes versions
- Esclaves disponibles en lecture seule
- Réplication avec moins de lag
  - Pratiquement synchrone
  - Mais techniquement toujours asynchrone

4.1 Hot Standby - Introduction

- Patch développé par la société 2nd Quadrant
- Financement par un grand nombre de sociétés
- Deux ans de développement
- Initialement prévue pour la version 8.4
- Permet l'accès en lecture seule aux serveurs esclave

4.2 Hot Standby - Configuration

- Maître
  - Configuration d'archivage habituelle
  - postgresql.conf : wal_level = 'hot_standby'
  - Redémarrage du maître
- Esclave
  - Configuration normale d'un Warm Standby
  - postgresql.conf : hot_standby = 'on'
  - Redémarrage de l'esclave

4.3 Streaming Replication - Introduction

- Patch basé sur le développement réalisé par NTT
- Permet d'avoir moins de lag dans la réplication
- Deux nouveaux processus:
  - walreceiver sur l'esclave
  - walsender sur le maître
  - L'esclave se connecte au maître
- Semi-asynchrone

- Pas synchrone mais très rapide malgré tout

4.4 Streaming Replication - Configuration

- Maître
  - postgresql.conf : max_wal_senders = 1
  - pg_hba.conf : autoriser connexion esclave
  - Redémarrage du maître
- Esclave
  - recovery.conf : restore_command, archive_cleanup_command, trigger_file
  - recovery.conf : standby_mode (on), primary_conninfo
  - Redémarrage de l'esclave

4.5 Administration

- Failover
  - Créer le fichier indiqué par trigger_file
- Switchover
  - Arrêter le maître, créer le fichier indiqué par trigger_file, reconstruire le maître en tant qu'esclave

- Supervision
  - pg_is_in_recovery(), pg_last_xlog_receive_location(), pg_last_xlog_replay_location()
  - Plugin munin, action check_postgres.pl

4.6 Avantages / Inconvénients

- Avantages
  - Facile à mettre en place, ne nécessite aucune modification des applications, ne désactive aucune fonctionnalité, dispose des esclaves en lecture seule avec un lag très léger
- Inconvénients
  - Administration complexe, supervision difficile, implication sur la sécurité mal appréhendée, pas de réplication synchrone, pas de réplication maître/maître, réplication de l'instance complète

5 Et les prochaines versions ?

- 9.1
  - Nouvel attribut REPLICATION pour les rôles
  - Nouvelle vue pg_stat_replication

- Nouvelle vue pg_stat_database_conflicts
- Module contrib pg_basebackup
- 9.X ?
  - Réplication synchrone

6 Conclusion

- La réplication interne est fiable et rapide
- Elle a aussi des défauts
- Mais les développeurs travaillent à les corriger
- On a le temps pour une petite démo ?