DALIBO

L'Expertise PostgreSQL

Haute disponibilité avec PostgreSQL

Table des matières

Haute-disponibilité avec PostgreSQL

1 À propos des auteurs...

» Auteur : Guillaume Lelarge
» Société : DALIBO
» Date : Février 2011
» URL :
https://support.dalibo.com/kb/conferences/haute_disponibilite_avec_postgresql/

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

Cette formation (diapositives, manuels et travaux pratiques) est sous licence **CC-BY-NC-SA**.

Vous êtes libre de redistribuer et/ou modifier cette création selon les conditions suivantes :

- Paternité
- Pas d'utilisation commerciale
- Partage des conditions initiales à l'identique

Vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'œuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait

qu'ils vous soutiennent ou approuvent votre utilisation de l'œuvre).

Vous n'avez pas le droit d'utiliser cette création à des fins commerciales.

Si vous modifiez, transformez ou adaptez cette création, vous n'avez le droit de distribuer la création qui en résulte que sous un contrat identique à celui-ci.

Ceci est un résumé explicatif du Code Juridique. La version intégrale du contrat est disponible ici : http://creativecommons.org/licenses/by-nc-sa/2.0/fr/legalcode

3 Au menu

- PostgreSQL
- Généralités sur la haute-disponibilité
- Poolers de connexion
- Réplication
- Répartition de charge
- Récapitulatif
- Un sondage
- Quel futur pour PostgreSQL ?

Cette présentation fait le tour de la haute-disponibilité avec PostgreSQL.

4 PostgreSQL

- Moteur de base de données libre
- Licence BSD

- Né en 1996
- Projet dirigé par sa communauté
- Bien qu'aidé et soutenu par de nombreuses entreprises
- Un des moteurs les plus fidèles au standard SQL
- Grand nombre de fonctionnalités entreprise

5 Haute-disponibilité

- 3 composantes
  - Pooling de connexions
  - Réplication
  - Répartition de charge

6 Pooling de connexions

- Accélère la connexion au serveur
- Mise en attente des connexions

- Permet d'accéder à plusieurs serveurs
  - Répartition de charge dans le cas de pgPool-II
  - Connexion à plusieurs serveurs dans le cas de pgBouncer

6.1 pgPool-II - introduction

- Premier logiciel de pooling pour PostgreSQL, créé et maintenu par la communauté
- Licence BSD
- Dernière version 3.0.1 (19 octobre 2010)
- http://pgpool.projects.postgresql.org/

6.2 pgPool-II - avantages

- Architecture prefork
- Contrôle des ressources
- Transparent pour les applications en mode pooling de connexions
- Indépendant du langage de développement
- Support de SSL
- Mise en place simple

6.3 pgPool-II - inconvénients

- Un peu fourre-tout : pooling, répartition de charge, réplication, parallélisation
- Impact sur les méthodes d'authentification
  - trust, password, crypt, md5 et pam

6.4 pgBouncer - introduction

- Dernier logiciel de pooling pour PostgreSQL, créé et maintenu par Skype
- Licence BSD
- Dernière version 1.4 (11 janvier 2011)
- http://pgfoundry.org/projects/pgbouncer/

6.5 pgBouncer - Avantages

- Les mêmes que pour pgPool-II
- Bien plus léger que pgPool-II
- Permet la connexion à plusieurs serveurs
- Différentes méthodes de pooling
  - Session, transaction, requête
- Console d'administration
  - Statistiques d'utilisation
  - Commandes du pooler

6.6 pgBouncer - Inconvénients

- Impact sur les méthodes d'authentification
  - trust, password, crypt, md5
- Pas de support de SSL

6.7 Autres solutions de pooling

- sqlrelay
- c3po (pour Java/Hibernate)

6.8 Comment choisir un pooler

- Si vous ne pensez pas avoir besoin de répartition de charge
  - pgBouncer
- Sinon
  - pgPool-II

7 Solutions de réplication

- But fréquent : pouvoir basculer sur un autre serveur si le premier tombait
- Autres buts
  - Disposer d'un serveur de tests
  - Disposer d'un serveur de génération de rapports
- Types de réplication
- Diffusion des modifications

7.1 Asynchrone Asymétrique

- Écritures uniquement sur le maître
- Mise à jour différée des tables sur l'autre serveur

7.2 Asynchrone Symétrique

- Écritures sur les deux « maîtres »
- Mise à jour différée des tables sur l'autre serveur
- Difficile d'avoir un respect d'ACID

7.3 Synchrone Asymétrique

- Écritures uniquement sur le maître
- Mise à jour immédiate des tables sur l'autre serveur
- Source de lenteurs

7.4 Synchrone Symétrique

- Écritures sur les deux « maîtres »
- Mise à jour immédiate des tables sur l'autre serveur

7.5 Warm Standby - Introduction

- Réplication interne de PostgreSQL
- Archivage, puis re-jeu des journaux de transactions
  - Le serveur maître envoie les journaux qu'il génère à l'esclave
  - L'esclave les rejoue dès réception

7.6 Warm Standby - Avantages

- Simple à mettre en place
- Et des outils pour automatiser encore plus:
  - Pitrtools (de Command Prompt)
  - Walmgr (de Skype)
- Interne, donc suivant au plus près les évolutions de PostgreSQL
- Réplique toutes les modifications

- Données comme structure des bases

7.7 Warm Standby - Inconvénients

- Pas de granularité sur les objets à répliquer
  - Tout le cluster
- Perte au pire d'un journal de transactions
- Esclaves non accessibles

7.8 Hot Standby

- Esclaves en lecture seule
- Possibilité de choisir entre
  - Réplication très proche
  - Requêtes longues pour rapport

7.9 Streaming Replication

- Réplication en flux
- Asynchrone
- Un processus sur le maître, un processus sur l'esclave

7.10 HS/SR - Inconvénients

- Toujours pas de granularité
- Switchover complexe
- Administration/monitoring complexe
- Pas de réplication synchrone
- Pas de maître/maître

7.11 Slony - Introduction

- Premier outil de réplication libre (PostgreSQL)
- Créé par un développeur majeur de PostgreSQL
- Licence BSD
- Dernières versions

- 1.2.22 (6 décembre 2010) : à partir de PostgreSQL 7.3
- 2.0.6 (6 décembre 2010) : à partir de PostgreSQL 8.3

- http://slony.info/

7.12 Slony - Techniques

- Réplication par trigger
- 1 maître sur un ensemble de tables et séquences
- Les autres nœuds sont esclaves pour cet ensemble…
- Mais peuvent être maîtres sur d'autres ensembles
- Cascade des serveurs possible

7.13 Slony - Avantages

- Choix des objets à répliquer
- Esclaves en lecture seule
- Mise à jour majeure de PostgreSQL facilitée
- Quelques outils simplifient la vie
  - Outils alt-Perl
  - Outils slony1-ctl

7.14 Slony - Inconvénients

- Pas de réplication de la structure d'une base
- Pas de réplication des Large Objects
  - Par contre, pas de soucis avec les Bytea
- Pas de réplication du TRUNCATE
- Complexe à maîtriser
- Une documentation qui, bien que complète, laisse à désirer

7.15 Londiste - Introduction

- Outil créé et maintenu par Skype
- Licence BSD
- Dernière version : 2.1.12 (10 novembre 2010)
- http://pgfoundry.org/projects/skytools/

7.16 Londiste - Techniques

- Réplication par trigger
- Un ensemble de tables a un maître et un esclave
- Mais pas de set comme pour Slony

7.17 Londiste - Avantages

- Très simple à mettre en place
  - Pas de configuration compliquée
- Choix des objets à répliquer
- Esclaves en lecture seule
- Outil bien conçu
  - Beaucoup de finition bien réalisée

7.18 Londiste - Inconvénients

- Pas de réplication de la structure d'une base
- Pas de réplication des Large Objects
  - Par contre, pas de soucis avec les Bytea
- Pas de support de l'exécution de scripts SQL sur le maître et sur l'esclave en même temps

- Pas de serveurs en cascade
- Pratiquement pas de documentation

7.19 Bucardo - Introduction

- Outil créé et maintenu par la société End Point Corporation
- Licence BSD
- Dernière version : 4.4.0 (14 octobre 2009)
- http://bucardo.org/wiki/Bucardo

7.20 Bucardo - Techniques

- Réplication par trigger
- Démon Perl à l'écoute de NOTIFY
- Utilise une base de données pour stocker ses méta-données
- Maître/maître possible
  - Dans ce cas, deux nœuds uniquement

7.21 Bucardo - Avantages

- Seul outil à faire du maître/maître
- Un outil utilisateur pour mettre en place la réplication
- Outil en fort développement
- Documentation en très forte augmentation

7.22 Bucardo - Inconvénients

- Nécessite au minimum PostgreSQL 8.1
- Ne fonctionne pas sous Windows, mais devrait accepter tout système Unix

7.23 Replicator - Introduction

- Outil créé par Command Prompt
- Auparavant connu sous le nom de Mammoth Replicator
- Licence BSD
- Dernière version : 1.8.2 (9 septembre 2009)
- https://public.commandprompt.com/projects/replicator

7.24 Replicator - Techniques

- Extension à PostgreSQL (patch)
- Nouvelles instructions SQL
  - ALTER TABLE une_table ENABLE REPLICATION ON SLAVE 0;
  - PROMOTE

7.25 Replicator - Avantages

- Réplication des données, mais aussi:
  - des « Large Objects »
  - des rôles
  - des droits

7.26 Replicator - Inconvénients

- Réplication d'une seule base par cluster
- Difficile de supprimer la réplication d'une base
- Une table répliquée ne peut plus voir son schéma modifié
- Une nouvelle version de PostgreSQL nécessite une nouvelle version du patch

7.27 pgPool-II - Techniques

- Réplication par requêtes SQL
- Une requête est envoyée à pgPool-II…
- Qui se charge de l'envoyer à tous les serveurs PostgreSQL

7.28 pgPool-II - Avantages

- Réplication de la structure d'une base
- Très simple à mettre en œuvre
- Fonctionnement très simple à comprendre
- Tous les nœuds sont disponibles en lecture

7.29 pgPool-II - Inconvénients

- Prérequis importants:
  - Ne pas utiliser les fonctions now(), random(), etc.
  - Les clients doivent passer par pgPool-II (SPOF)
- Si un serveur tombe, il faut entièrement le reconstruire

7.30 rubyrep - Introduction

- Outil communautaire, mais non spécifique à PostgreSQL
- Licence MIT
- Dernière version : 1.0.9 (14 octobre 2009)
- http://www.rubyrep.org/

7.31 rubyrep - Techniques 1/2

- Mode scan
  - Détecte les différences de contenu entre deux tables

- Mode sync
  - Synchronise deux tables
  - Résolution des conflits personnalisable

7.32 rubyrep - Techniques 2/2

- Mode réplication
  - Réplication par triggers
  - Gère maître/maître comme maître/esclave
    - Gestion des conflits personnalisable
  - Configuration automatique des triggers
  - Détection automatique des nouvelles tables

7.33 rubyrep - Avantages

- Mode scan et sync très intéressant pour les réplications à un instant t
- Maître/maître possible
- Détection automatique des nouvelles tables

7.34 rubyrep - Inconvénients

- Ruby…

7.35 DRBD - Introduction

- Outil créé par LINBIT HA-Solutions GmbH
- Licence GPL
- Dernière version : 8.3.10
- http://www.drbd.org/

7.36 DRBD - Techniques

- Réplication des blocs disque
- Sous-couche disque
- Module noyau
- Quelques outils pour la configuration

7.37 DRBD - Avantages

- Simple à mettre en œuvre
  - Intégré à Linux depuis la 2.6.33
- Réplication synchrone
- Documentation excellente
- Intégration facile à HeartBeat / Pacemaker

7.38 DRBD - Inconvénients

- Pas de granularité sur la réplication
  - Tout le cluster obligatoirement
- Esclaves indisponibles
- Lenteur à prévoir dû au caractère synchrone
- Deux serveurs uniquement

7.39 RHCS - Introduction

- Red Hat Cluster Suite
- Réplication par disque partagé
- Un seul moteur fonctionnel à un instant t
- Bascule automatique suite à une heuristique
  - Perte du réseau (ping)
  - Perte de salle – panne électrique (disque de quorum)

7.40 RHCS - Avantages

- Forcément synchrone
- Bascule simple à mettre en place
- Disque partagé, donc pas de lenteur en écriture

7.41 RHCS - Inconvénients

- Pas de granularité sur la réplication
- Esclaves indisponibles
- Attention à ce que les deux serveurs PostgreSQL ne soient pas exécutés en même temps
  - Sinon, catastrophe assurée !

- SPOF sur la baie

7.42 Autres solutions de réplication

- PGCluster
  - Ancienne implémentation trop lente
  - Nouveau prototype en cours de conception/codage
- Postgres-R
  - Patch, prototype, maître/maître synchrone
  - Fournit en plus pooling de connexions et répartition de charge
- CyberCluster
- Tungsten / Continuent

8 Répartition de charge

- But : répartir les requêtes sur plusieurs serveurs pour accélérer le tout
- Deux moyens principalement
  - Répartir les requêtes suivant leur nombre

- Répartir les requêtes suivant la charge des serveurs

8.1 pgPool-II - Répartition

- Pool circulaire de serveur
- Requêtes en écriture sur le serveur 0
  - Réplication réalisée par pgPool-II, Slony ou la Streaming Replication
- Requêtes en lecture sur l'un des serveurs disponibles
- Répartition par nombre de requêtes, et non pas charge réelle des serveurs
- Possibilité de préciser un poids pour chaque serveur

8.2 PL/proxy

- Langage créé par Skype
- Partitionnement horizontal

9 Tableau récapitulatif

Réplication	Type	Esclave	Schéma	Synchrone
Warm Standby	Journaux	Indisponible	Oui	Non
HS/SR	Journaux	Lecture seule	Oui	Non
Slony	Triggers	Lecture seule	Non	Non
Londiste	Triggers	Lecture seule	Non	Non
Bucardo	Triggers	Lecture/Écriture	Non	Non
Replicator	Interne	Oui	Oui	Non
pgPool-II	Requêtes	Lecture seule	Oui	Non
rubyrep	Triggers	Lecture/Écriture	Non	Non
DRBD	Blocs disques	Indisponible	Oui	Oui
RHCS	Disque partagé	Indisponible	Oui	Oui

10 Sondage

Outils	Réponse	Pourcentage
pgPool-II	23	11,79%
Bucardo	9	4,61%
Slony	74	37,95%
Londiste	39	20,00%
Continuent	5	2,56%

pgCluster	12	6,15%
DRBD	13	6,67%
Autres	20	10,26%

Chiffres en évolution depuis avril 2009: Slony descend beaucoup et Londiste grimpe bien, suivi par Bucardo.

11 Conclusion

- Quel que soit le projet choisi pour répliquer les données, il ne faut pas oublier :
  - de bien définir son besoin
  - d'identifier tous les SPOF
  - de redonder chaque service jugé critique
  - de bien superviser son cluster
  - de répéter les opérations de switchover et failover