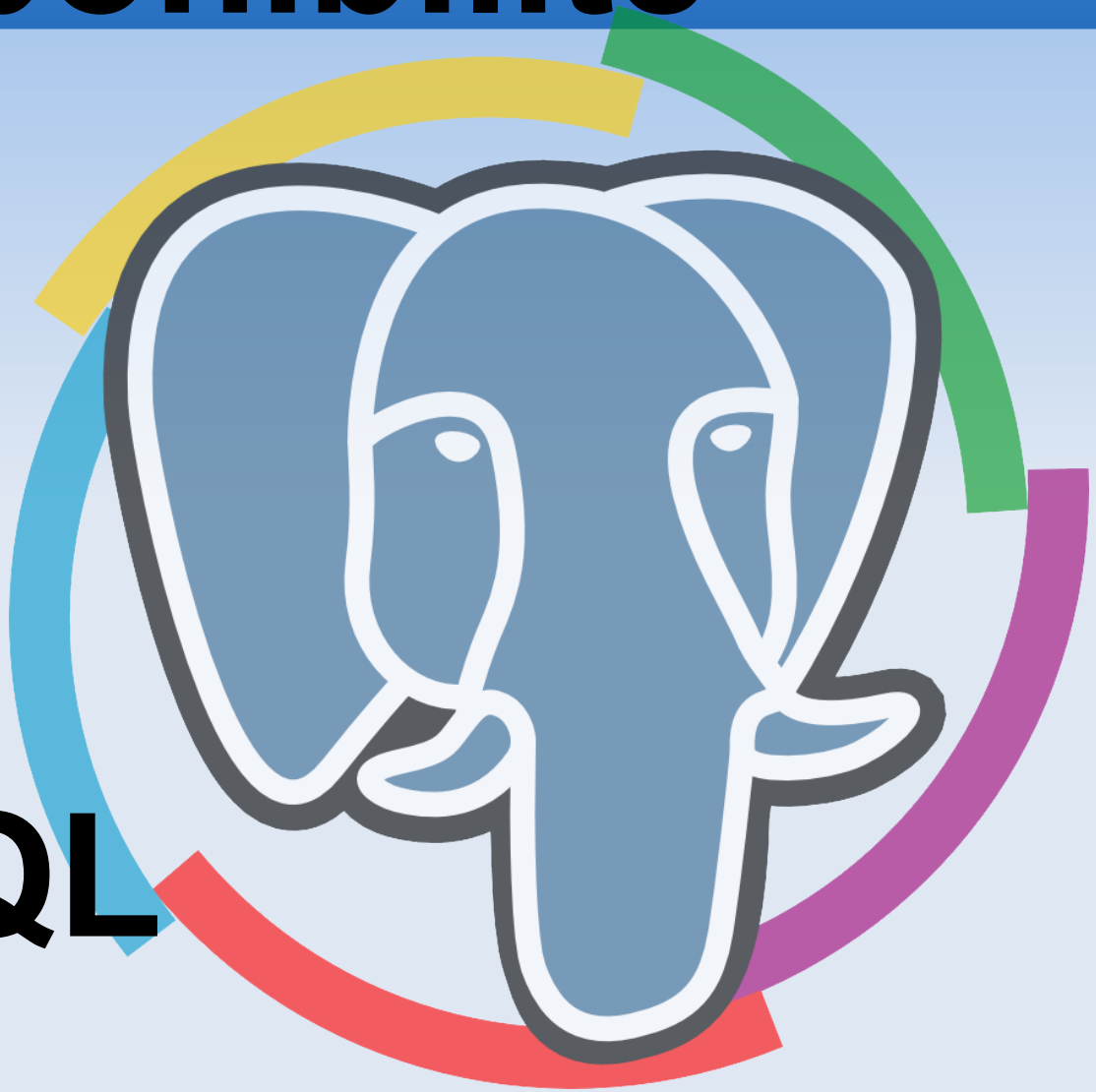
Haute disponibilité

avec



PostgreSQL

v.1.0
16/10/2009

Guillaume Lelarge

- Membre fondateur de PostgreSQLfr
- Contributeur de longue date
  - Traduction de la documentation
  - Patchs pour de nombreux projets (pgAdmin)
  - Patchs pour PostgreSQL
- Promotion de PostgreSQL
  - Articles dans linux mag
  - Conférences
  - Blog
- Consultant chez Dalibo

Sommaire

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

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

Haute disponibilité

- Pooling de connexions
- Réplication
  - type de réplication
  - diffusion des modifications
- Répartition de charge

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

pgPool-II - introduction

- Premier logiciel de pooling pour PostgreSQL, créé et maintenu par la communauté
- Licence BSD
- Dernière version 2.2.5 (4 octobre 2009)
- http://pgpool.projects.postgresql.org/

pgPool-II - avantages

- Architecture prefork
- Contrôle des ressources
- Transparent pour les applications en mode pooling de connexions
- Indépendant du langage de développement

- Mise en place simple

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
- Pas de support du SSL

pgBouncer - introduction

- Dernier logiciel de pooling pour PostgreSQL, créé et maintenu par Skype
- Licence BSD
- Dernière version 1.3.1 (6 juillet 2009)
- https://developer.skype.com/SkypeGarage/DbProjects/PgBouncer

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

pgBouncer - inconvénients

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

Autres solutions de pooling

- sqlrelay
- Java/Hibernate

Comment choisir...

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

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

Asynchrone, asymétrique

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

Asynchrone, symétrique

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

- Difficile d'avoir un respect d'ACID

Synchrone, asymétrique

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

- Source de lenteurs

Synchrone, symétrique

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

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

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

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

Slony - introduction

- Premier outil de réplication libre (PostgreSQL)
- Créé par un développeur majeur de PostgreSQL
- Licence BSD
- Dernières versions
  - 1.2.17 (14 octobre 2009) : pour PostgreSQL 7.3 à 8.4
  - 2.0.2 (8 mai 2009) : pour PostgreSQL 8.3
  - PostgreSQL 8.4 pourra utiliser la 1.2.17 et la 2.0.3
- http://slony.info/

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

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

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

Londiste - introduction

- Outil créé et maintenu par Skype
- Licence BSD
- Dernière version : 2.1.10 (31 août 2009)
- https://developer.skype.com/SkypeGarage/DbProjects/SkyTools

Londiste - techniques

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

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

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

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

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

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

Bucardo - inconvénients

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

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://projects.commandprompt.com/public/replicator

Replicator - techniques

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

Replicator - avantages

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

Replicator - inconvénients

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

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

PgPool-II - avantages

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

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

rubyrep - introduction

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

rubyrep – techniques 1/2

- Mode scan
  - Détecte les différences de contenu entre deux tables
- Mode sync
  - Synchronise deux tables
  - Résolution des conflits personnalisable

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

rubyrep - avantages

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

rubyrep - inconvénients

- Ruby...

DRBD - introduction

- Outil créé par LINBIT HA-Solutions GmbH

- Licence GPL
- Dernière version : 8.3.4 (6 octobre 2009)

- http://www.drbd.org/

DRBD - techniques

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

DRBD - avantages

- Simple à mettre en œuvre
- Réplication synchrone
- Documentation excellente
- Intégration facile à HeartBeat / Pacemaker

DRBD - inconvénients

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

Red Hat Cluster Suite - introduction

- 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)
-

Red Hat Cluster Suite - avantages

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

Red Hat Cluster Suite - 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 !

Autres projets

- 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

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

pgPool-II

- Pool circulaire de serveur
- Requêtes en écriture sur le serveur 0
  - Réplication réalisée par pgPool-II ou Slony
- 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

PL/proxy

- Langage créé par Skype
- Partitionnement horizontal

Tableau récapitulatif

Réplication	Type	Esclave	Réplication du schéma	Synchrone
Warm Standby	Journaux	Indisponible	Oui	Non
Slony	Trigger	Lecture seule	Non	Non
Londiste	Trigger	Lecture seule	Non	Non
Bucardo	Trigger	Lecture/Écriture	Non	Non
Replicator	Interne	Oui	Oui	Non
pgPool-II	Requêtes	Lecture seule	Oui	Non
rubyrep	Trigger	Lecture/Écriture	Non	Non
DRBD	Blocs disques	Indisponible	Oui	Oui
Red Hat Cluster	Disque partagé	Indisponible	Oui	Oui

Sondage www.postgresql.org

- Sondage effectué

Outils	Réponses	Pourcentage
pgPool-II	23	11.79%
Bucardo	9	4.61%
Slony-I	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
  - Londiste grimpe bien, suivi par Bucardo

Pour terminer, de la lecture

- Manuel officiel
  - http://docs.postgresqlfr.org/current/
  - http://www.postgresql.org/docs/books/
- Hors-série 44

  GNU/Linux Magazine France

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 monitorer son cluster
  - de se préparer à un éventuel Failover

PGDay 2009

- 6 et 7 novembre 2009
- Paris, dans les locaux de Télécom Paristech
- 25 conférences, dont dix en français
- Inscription obligatoire

Merci...

- d'être venu
- et de m'avoir écouté :)

- Vos questions sont les bienvenues !