Mixons les nouveautés PostgreSQL 10

Partitionnement déclaratif et réplication logique

Thibaut Madelaine

novembre 2017

DALIBO
L'expertise PostgreSQL

QUI SUIS-JE ?

- Thibaut Madelaine
- chez Dalibo depuis 1 an
- Longtemps développeur logiciel
- Désormais également DBA, formateur, consultant

PostgreSQL version 10

- Sortie le 5 octobre 2017
- supportée jusqu'en octobre 2022

DE TRÈS NOMBREUSES ÉVOLUTIONS

- le changement de numérotation
- les changements de nommages
- des améliorations de performances
- …

DEUX BELLES NOUVEAUTÉS

- Partitionnement natif
- Réplication logique

Partitionnement

- Petit rappel sur l’ancien partitionnement
- Nouveau partitionnement
- Nouvelle syntaxe
- Quelques limitations

ANCIEN PARTITIONNEMENT

- Le partitionnement par héritage se base sur
  - la notion d'héritage (1 table mère et des tables filles)
  - des triggers pour orienter les insertions vers les tables filles
  - des contraintes d'exclusion pour optimiser les requêtes
- Disponible depuis longtemps

LIMITATIONS DE L'ANCIEN PARTITIONNEMENT

- maintenance fastidieuse
- performance dégradée à cause du trigger
- pas de contrainte d'unicité globale

NOUVEAU PARTITIONNEMENT

- Mise en place et administration simplifiées car intégrées au moteur
- Plus de trigger
  - insertions plus rapides
  - routage des données insérées dans la bonne partition
  - erreur si aucune partition destinataire

CHANGEMENT DU CATALOGUE SYSTÈME

- nouvelles colonnes dans `pg_class`
- nouveau catalogue `pg_partitioned_table`

NOUVEAUX ORDRES SQL

- attacher/détacher une partition
- contrainte implicite de partitionnement
- expression possible pour la clé de partitionnement
- sous-partitions possibles

EXEMPLE DE PARTITIONNEMENT PAR LISTE

- Créer une table partitionnée :

```
CREATE TABLE t1(c1 integer, c2 text) PARTITION BY LIST (c1);
```

- Ajouter une partition :

```
CREATE TABLE t1_a PARTITION OF t1 FOR VALUES IN (1, 2, 3);
```

- Détacher la partition :

```
ALTER TABLE t1 DETACH PARTITION t1_a;
```

- Attacher la partition :

```
ALTER TABLE t1 ATTACH PARTITION t1_a FOR VALUES IN (1, 2, 3);
```

EXEMPLE DE PARTITIONNEMENT PAR INTERVALLES

- Créer une table partitionnée :

```
CREATE TABLE t2(c1 integer, c2 text) PARTITION BY RANGE (c1);
```

- Ajouter une partition :

```
CREATE TABLE t2_1 PARTITION OF t2 FOR VALUES FROM (1) TO (100);
```

- Détacher une partition :

```
ALTER TABLE t2 DETACH PARTITION t2_1;
```

CLÉ DE PARTITIONNEMENT MULTI-COLONNES

- Clé sur plusieurs colonnes acceptée
  - uniquement pour le partitionnement par intervalles
- Créer une table partitionnée avec une clé multi-colonnes :

```
CREATE TABLE t3(c1 integer, c2 text, c3 date)
  PARTITION BY RANGE (c1, c3);
```

- Ajouter une partition :

```
CREATE TABLE t3_a PARTITION of t3 FOR VALUES
  FROM (1,'2017-08-10') TO (100, '2017-08-11');
```

LIMITATIONS

- La table mère ne peut pas avoir de données
- La table mère ne peut pas avoir d'index
  - ni PK, ni UK, ni FK pointant vers elle
- Pas de colonnes additionnelles dans les partitions
- L'héritage multiple n'est pas permis
- Valeurs nulles acceptées dans les partitions uniquement si la table partitionnée le permet
- Pas d'insertion dans des partitions distantes

ET DANS LA VERSION 11 ?

- Nouvelle méthode de partitionnement par clé de hachage

Place à la démo !

- création de table partitionnée en 9.6 et 10
- étude des limitations
- maintenance

DÉMO PARTITIONNEMENT : CRÉATION

DÉMO PARTITIONNEMENT : LIMITATIONS

- création d’index
- mise à jour
- insertion de données hors limite

DÉMO PARTITIONNEMENT : MAINTENANCE

Réplication logique

- Petit rappel sur la réplication physique
- Qu’est-ce que la réplication logique ?
- Fonctionnement
- Limitations
- Exemples

RÉPLICATION PHYSIQUE

- Réplication de toute l'instance
  - au niveau bloc
  - par rejeu des journaux de transactions
- Quelques limitations :
  - intégralité de l'instance
  - même architecture (x86, ARM⋯)
  - même version majeure
  - pas de requête en écriture sur le secondaire

RÉPLICATION LOGIQUE - PRINCIPE

- Réutilisation de l'infrastructure existante
  - réplication en flux
  - slots de réplication
- Réplique les changements sur une seule base de données
  - d'un ensemble de tables défini
- Uniquement INSERT / UPDATE / DELETE
  - pas les DDL, ni les TRUNCATE

FONCTIONNEMENT

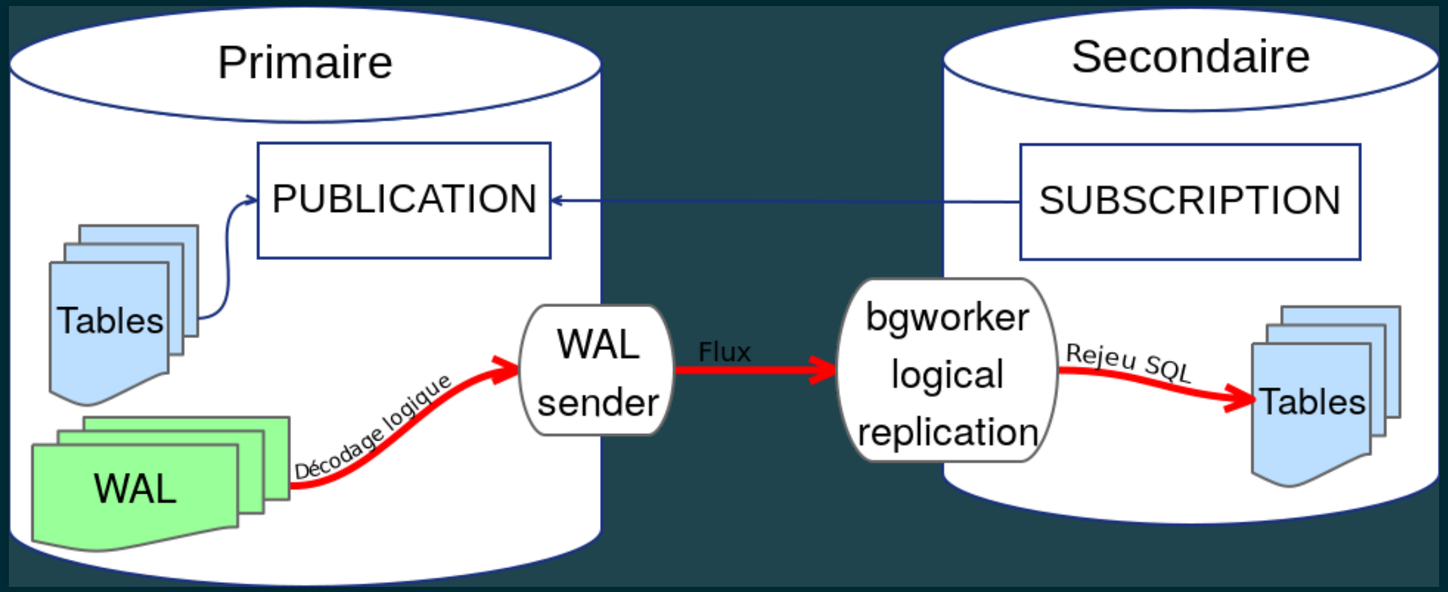

Schéma du fonctionnement de la réplication logique

LIMITATIONS

- Non répliqué :
  - Schéma
  - Séquences
  - Large objects
- Pas de publication des tables parents du partitionnement
- Ne convient pas comme fail-over
- Contrainte d'unicité nécessaire pour `UPDATE` et `DELETE`

DÉMO RÉPLICATION LOGIQUE : PUBLICATION

DÉMO RÉPLICATION LOGIQUE : SOUSCRIPTION

DÉMO RÉPLICATION LOGIQUE : MODIFICATION DES DONNÉES

Des questions ?