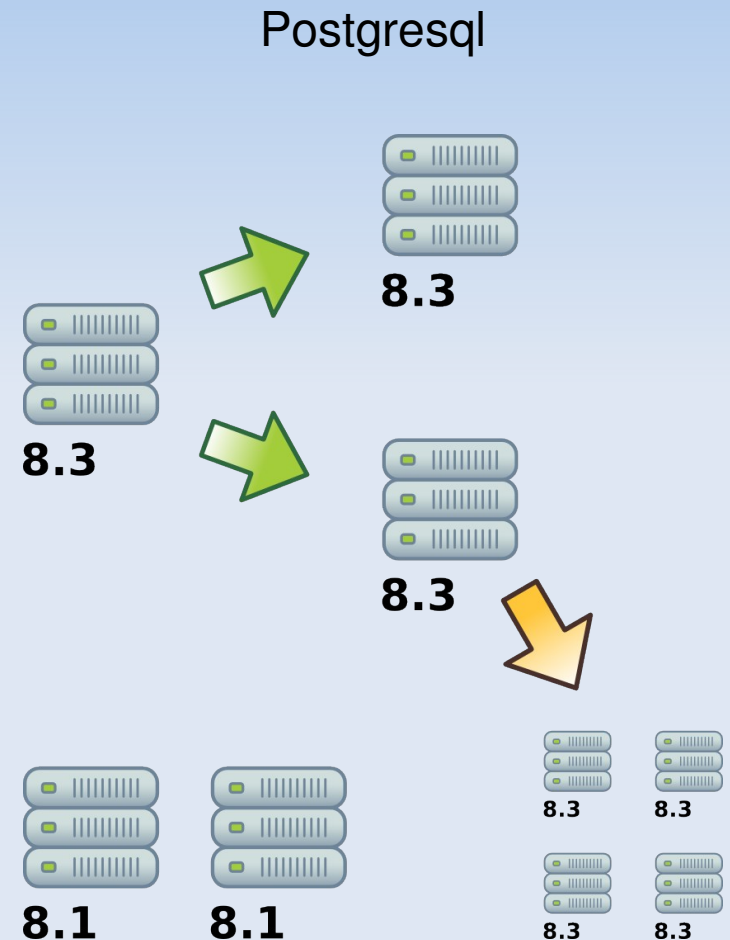


Replication avec Londiste

- Directeur technique de Fotolia, banque d'images libres de droits
- Base de données principale:
 - Environ 150 GB de données
 - Dans 150 tables
 - La plus grosse table: 22 millions d'enregistrements, 12 GB
- Choix de londiste
 - Écrit en python, donc compréhensible et maintenable
 - Utilisation future d'autres parties de skytools, comme pgq, logutriga, cube_dispatcher, ...
- ... par rapport à slony
 - Marre d'avoir des fichiers de config à recommencer de comprendre à chaque fois
 - Envie de tester de nouvelles solutions



Migration 8.1 vers 8.3, préparation

- Lecture des release notes de la 8.3
- Installation du serveur 8.3
- Mise aux normes de l'applicatif pour la 8.3 (cast essentiellement)
- Choix de londiste
- Installation de londiste sur les deux serveurs, du provider sur la 8.1 et du subscriber sur la 8.3
- Ajout des tables et des séquences au provider



8.1



8.3



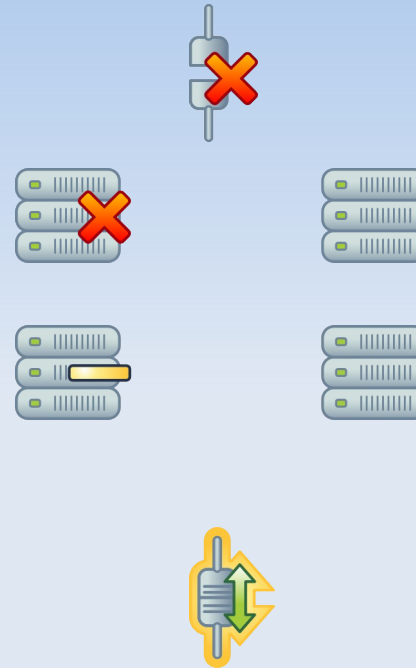
Migration 8.1 vers 8.3, synchro et tests

- Ajout des tables et des séquences au subscriber
- Ne pas oublier de créer les index sur les tables (peut se faire après la copie)
- Lancement de la synchronisation
- Aucun problème rencontré
- Copie totale en moins de 24 heures, création des index incluse
- La synchronisation est effective
- Tests de requêtes sur la 8.3, en lecture seule évidemment



Migration 8.1 vers 8.3, fin

- Mise en maintenance du site, page d'attente affichée aux visiteurs
- Coupure de la 8.1, coupure réplication
- Changement applicatif pour la connexion à la base de données principale
- Tests privés
- Réouverture du site



Conclusion et questions

- La coupure du site aura alors duré moins d'une heure
- L'essentiel dans les tests applicatifs après le changement de connexion
- Où me joindre:
 - ludovic@fotolia.com
 - luddic@gmail.com
 - Skype: ludoviclevesque
 - Physiquement en Seine-et-Marne
 - Ou bien dans un resto/bar sur Paris...

Monitoring & maintenance

- Nagios plugin

- Basé sur `pgq.get_consumer_info()`;

```
(postgres@fo-db4:5432) [fotolia] # select consumer_name, EXTRACT(second from lag) from pgq.get_consumer_info();
consumer_name      | date_part
-----+-----
londiste_db4_db3_replication | 1.60188
londiste_db4_db5_replication | 0.597607
(2 rows)
```

- Vérification de la synchro

- Fait un `COUNT()` sur les deux serveurs après avoir synchronisé les données
 - `/usr/bin/python ./bin/londiste.py etc/londiste-db4-db5.ini compare $TABLE`

```
/usr/bin/python ./bin/londiste.py etc/londiste-db4-db5.ini compare content
2008-09-17 00:26:36,032 29693 INFO Locking public.content
2008-09-17 00:26:36,033 29693 INFO Syncing public.content
2008-09-17 00:26:38,039 29693 INFO Counting public.content
2008-09-17 00:26:41,515 29693 INFO srcdb: res = 9386928
2008-09-17 00:26:43,507 29693 INFO dstdb: res = 9386928
```

Installation de londiste

- Récupération des sources
 - <https://developer.skype.com/SkypeGarage/DbProjects/SkyTools/Londiste>
- Compilation, création de .deb si voulues
- Installation
- Création des fichiers de config
 - Pour pgqadm
 - Pour chaque réplication
- Installation et lancement du ticker
 - `/usr/bin/python ./bin/pgqadm.py install`
 - `/usr/bin/python ./bin/pgqadm.py -d etc/ticker.ini ticker`
- Installation et lancement de londiste
 - `/usr/bin/python ./bin/londiste.py etc/londiste-db4-db5.ini provider install`
 - `/usr/bin/python ./bin/londiste.py etc/londiste-db4-db5.ini subscriber install`
 - `/usr/bin/python ./bin/londiste.py etc/londiste-db4-db5.ini replay -d`
- Création des tables, londiste ne le faisant pas tout seul
- Ajout des tables et des séquences à la réplication

Ajout d'une replication

- Ajout des subscribers
 - `/usr/bin/python ./bin/londiste.py etc/londiste-db4-db5.ini subscriber install`
- Ajout des tables souhaitées
 - Création de la table sur le subscriber
 - Ajout au provider (seulement si elle n'est pas encore déclarée comme répliquée)
 - `/usr/bin/python ./bin/londiste.py etc/londiste-db4-db5.ini provider add $TABLE`
 - Ajout au subscriber
 - `/usr/bin/python ./bin/londiste.py etc/londiste-db4-db5.ini subscriber add $TABLE`
 - On attend la copie des données
- Commandes utiles
 - Liste les tables disponibles sur le provider non répliqué sur le subscriber
 - `/usr/bin/python ./bin/londiste.py etc/londiste-db4-db5.ini subscriber missing`
 - Check la structure des tables du subscriber par rapport au provider
 - `/usr/bin/python ./bin/londiste.py etc/londiste-db4-db5.ini subscriber check`

etc/ticker.ini

[pgqadm]

job_name = pgqadm_fotolia_db4

db = dbname=fotolia user=postgres host=127.0.0.1 port=5432

how often to run maintenance [seconds]

maint_delay = 600

how often to check for activity [seconds]

loop_delay = 0.1

logfile = /usr/local/skytools-2.1.7/log/%(job_name)s.log

pidfile = /usr/local/skytools-2.1.7/pid/%(job_name)s.pid

etc/londiste-db4-db5.ini

[londiste]

job_name = londiste_db4_db5_replication

provider_db = dbname=fotolia user=postgres port=5432 host=fo-db4

subscriber_db = dbname=fotolia user=postgres port=5433 host=fo-db5

it will be used as sql ident so no dots/spaces

pgq_queue_name = londiste.db4_replika

#logfile = /usr/local/skytools-2.1.7/log/%(job_name)s.log

pidfile = /usr/local/skytools-2.1.7/pid/%(job_name)s.pid

etc/londiste-db4-db3.ini

[londiste]

job_name = londiste_db4_db3_replication

provider_db = dbname=fotolia user=postgres port=5432 host=fo-db4

subscriber_db = dbname=fotolia user=postgres port=5433 host=fo-db3

it will be used as sql ident so no dots/spaces

pgq_queue_name = londiste.db4_replika

#logfile = /usr/local/skytools-2.1.7/log/%(job_name)s.log

pidfile = /usr/local/skytools-2.1.7/pid/%(job_name)s.pid