# Migration vers PostgreSQL - Mener de gros volumes de données à bon port (5432)

21 juin 2022

Philippe Beaudoin
*Consultant*

DALIBO
L'expertise PostgreSQL

# La migration des données

- Dans un projet de migration vers PostgreSQL, la bascule des données
  - Peu consommatrice de charge de travail
  - Mais doit être : fiable, efficace, contrôlée, industrialisable
  - Interruption de service compatible avec les contraintes
- Migrer structures et contenus sont des sujets différents
  - 1 structure ↔ plusieurs contenus
    - Développement, test, production, production multiple,…
  - Gérer et versionner la référence de la structure d’une base

DALIBO

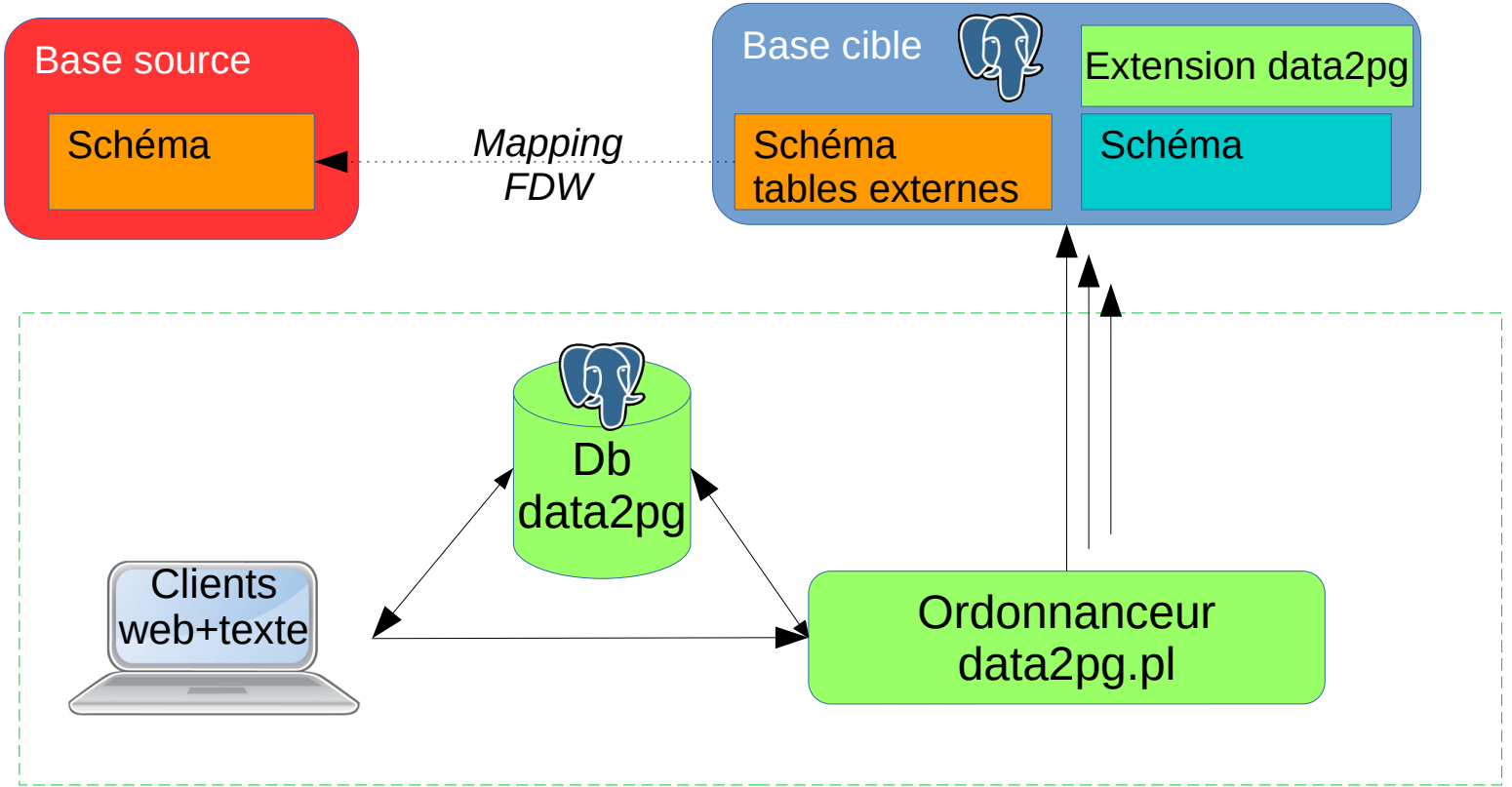
https://github.com/dalibo/data2pg

# Data2Pg - Objectifs

- Migrer efficacement des contenus de bases non PostgreSQL vers PostgreSQL
  - Via Foreign Data Wrapper
  - Tables + séquences
- Tracer, suivre et historiser les opérations
- Contrôler structures et contenus
- Reprise sur incident efficace
- Fonctions connexes
  - Analyser les données de la source
  - Comparer des bases source et cible pour tests des applications

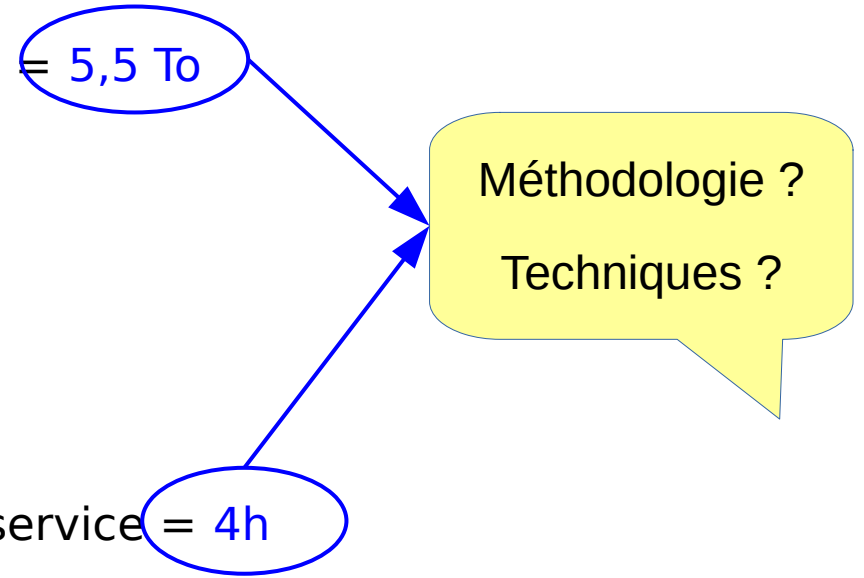
DALIBO

# Data2Pg - Architecture



# Cas réel

- Éditeur de progiciels
- Base source Oracle
- 20 milliards de lignes
- 4 To de données + 1,5 To d'index = 5,5 To
- 1800 tables et séquences
- Moyens matériels représentatifs
  - 2 serveurs Linux 20 cœurs
  - Baie de disques efficace
  - Réseau 1 Gbps
- Durée maximum de coupure du service = 4h

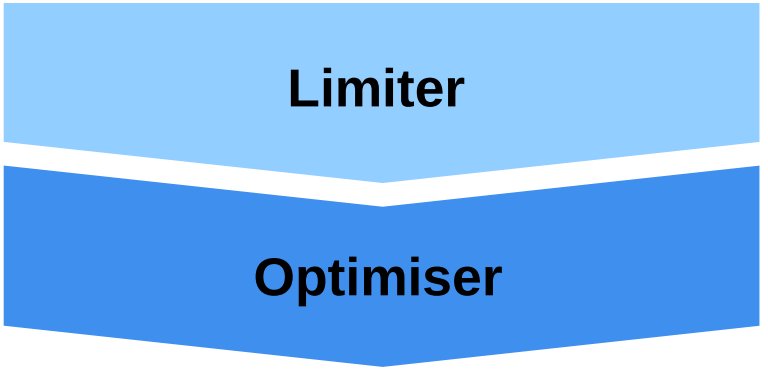


# Limiter

# Limiter

- Supprimer les tables obsolètes
- Passer les traitements de purge et archivage avant la migration
- Garder vides les tables intermédiaires de gros traitements

DALIBO


Limiter
Optimiser

# Optimiser le processus

- La copie directe par FDW optimise les mouvements de données
- Charger les tables sans index
  - 35 % de gain mesuré sur une base
- Tirer profit au maximum des caches
  - Placer juste après le chargement de la table : la recréation des index, les ANALYZE, les contrôles de contenus
- Éviter les contrôles des Foreign Keys quand les données sur la source sont censées être cohérentes
  - Techniques spécifiques (intégrées dans Data2Pg)
- Avoir des données pleinement opérationnelles juste après le chargement
  - Garder l'auto-vacuum mais ANALYZE explicites
  - Charger les données triées quand un index CLUSTER existe pour éviter des réorganisations post migration

DALIBO

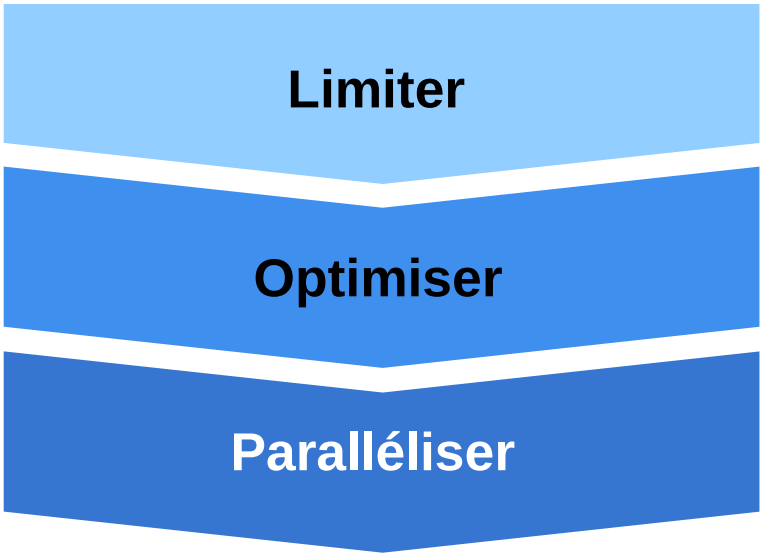
# Optimiser PostgreSQL

- Quelques paramètres positionnables au niveau session
  - `synchronous_commit = off`
  - `maintenance_work_mem`
  - `max_parallel_maintenance_workers`
- Un jeu de paramètres spécifiques de l'instance pour la durée de la migration ?
  - Que si pas d'autres bases en production dans l'instance
  - `wal_level = minimal`
    - => reconstruire les réplicas post-bascule
  - `fsync = off`
    - => revenir en « configuration production » avant l'ouverture du service

  - Un fichier `migration.conf` appelé par `include_if_exists` dans postgresql.conf

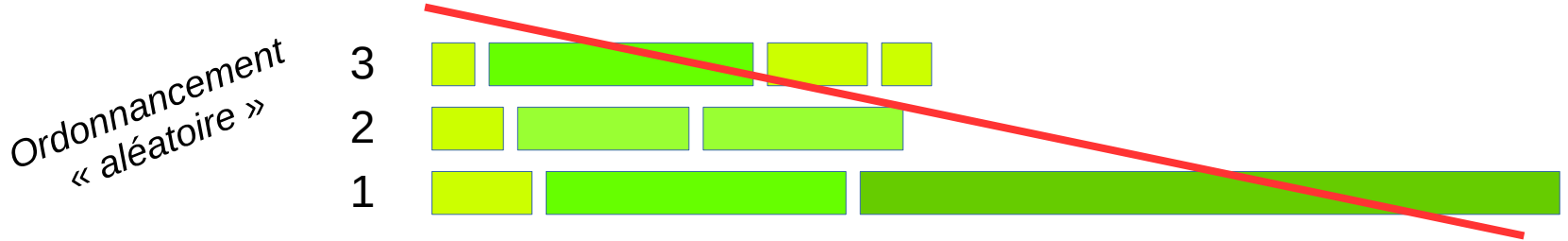
# Cas réel

- Après optimisations : 176h51mn => 7,33 jours


Limiter
Optimiser
Paralléliser

# Paralléliser les copies de tables

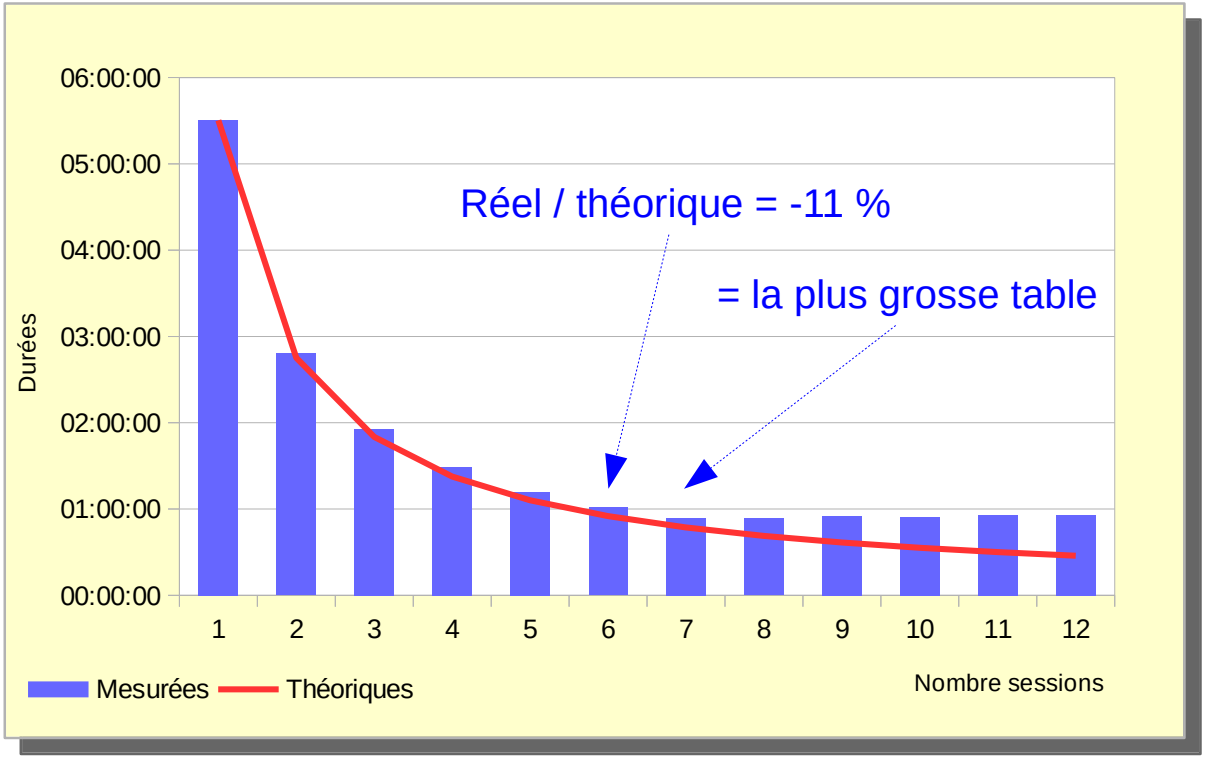
- Répartir les copies de tables et séquences sur plusieurs connexions




- Par durée décroissante, pour une utilisation optimum des sessions
  - Taille des tables dans les statistiques sur la base source
  - Ou référence de temps d'une exécution précédente

# Efficacité du parallélisme

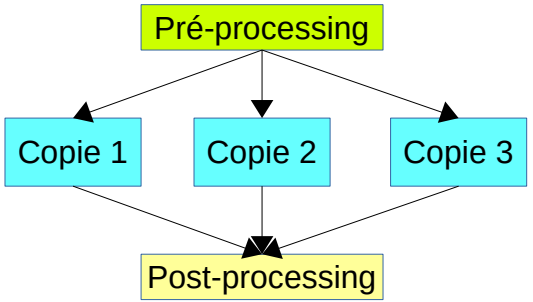
- Mesures sur 3 % du volume




- Les limites :
  - Les sessions doivent se partager les ressources : puissances CPU, débit réseau, débit I/O
  - La durée de la plus grosse table

# Paralléliser le traitement d'une table

- Pour les plus grosses tables, copier la table en plusieurs morceaux
  - Une clause WHERE pour chaque morceau
  - Attention : filtrer sur la source (push-down)




- Quelle clé de répartition ?
  - Données numériques => fonction modulo
  - Données alphanumériques ou temporelles => range
    - Comment identifier les bornes ?

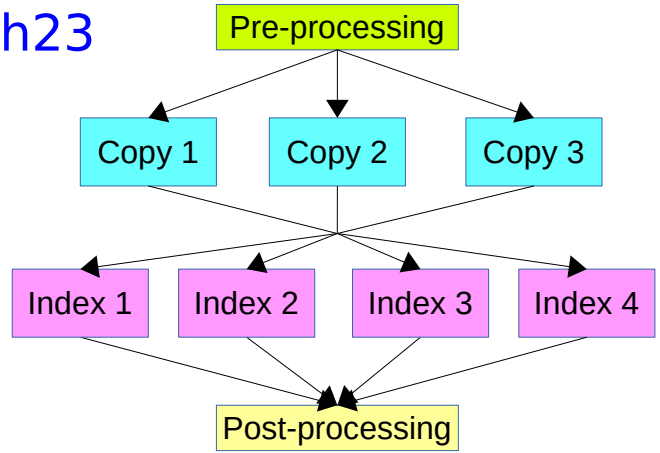
```
SELECT percentile_disc(0.5) WITHIN GROUP (ORDER BY <col>) FROM <table>
```

    - Beaucoup plus rapide mais approximatif : les stats PostgreSQL

```
SELECT histogram_bounds FROM pg_stats WHERE schemaname = <schema>
  AND tablename = <table> AND attname = <colonne>
```

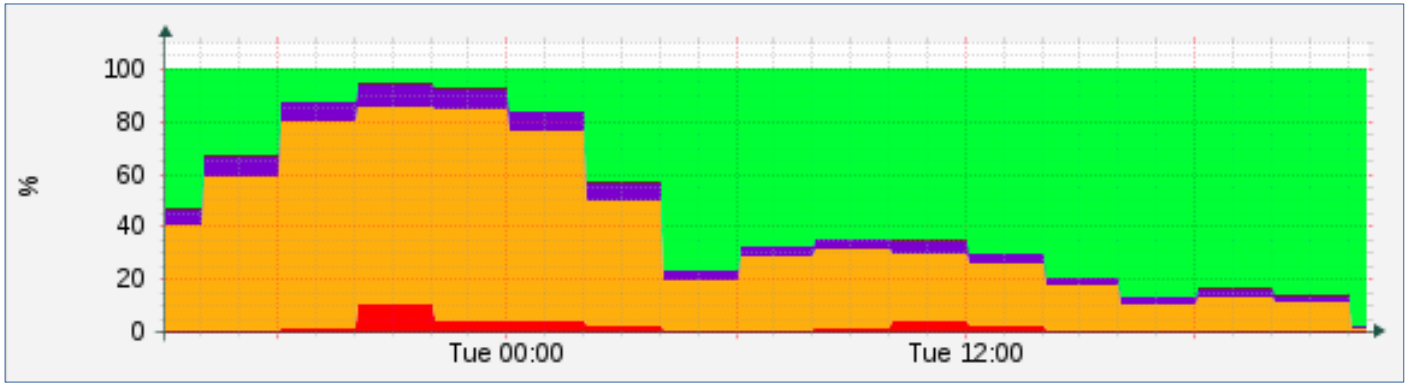
# Cas réel

- Avant parallélisation (rappel) : 176h51
- Avec parallélisation « simple » : 31h08
  - 21 sessions, dont 2 pour les plus petites tables
    - Pour que l'auto-vacuum démarre rapidement
- Avec parallélisation « optimisée » : 19h23
  - Découpage des 5 plus grosses tables en 2 ou 3 morceaux
  - Parallélisation de la reconstruction des index des 3 plus grosses tables

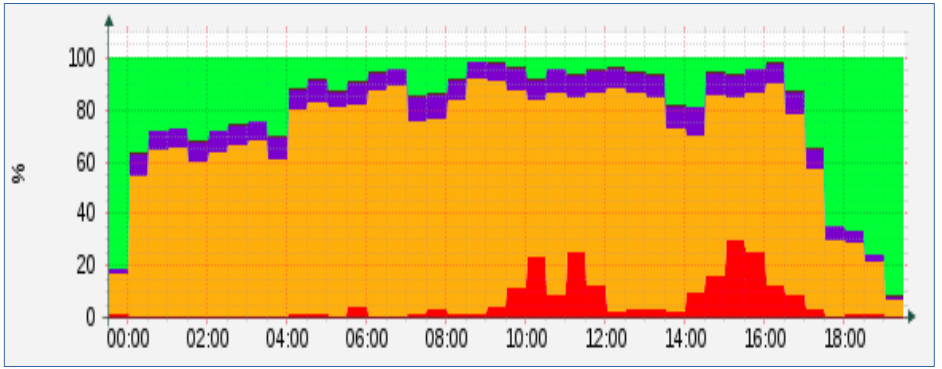


# Cas réel : utilisation CPU VM PostgreSQL

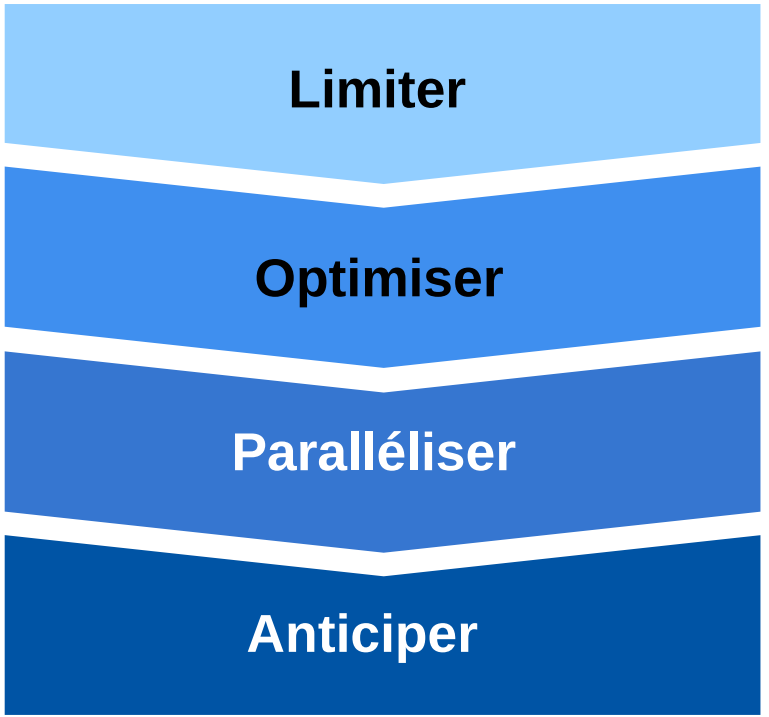
Parallélisation « simple »




Parallélisation avec découpage de tables




idle
system
user
IO wait


Limiter
Optimiser
Paralléliser
Anticiper

# Anticiper

- Migrer quelques tables avant les autres, base source ouverte en mise à jour
  - Des tables qui ne seront pas mises à jour avant la migration
    - Ex : tables d'archive
  - Des tables dont on peut identifier les changements par leur contenu
    - Ex : un historique avec que des INSERT et une date d'insertion
    - Copie incrémentale, avec une clause WHERE
  - Des tables possédant déjà un trigger traçant les mises à jour
    - Copie incrémentale, avec une requête spécifique exploitant la trace
  - Autre cas
    - Création d'un trigger sur la base source pour tracer les changements et copie incrémentale
      - Invasif et impact performances sur la source

DALIBO

# Cas réel

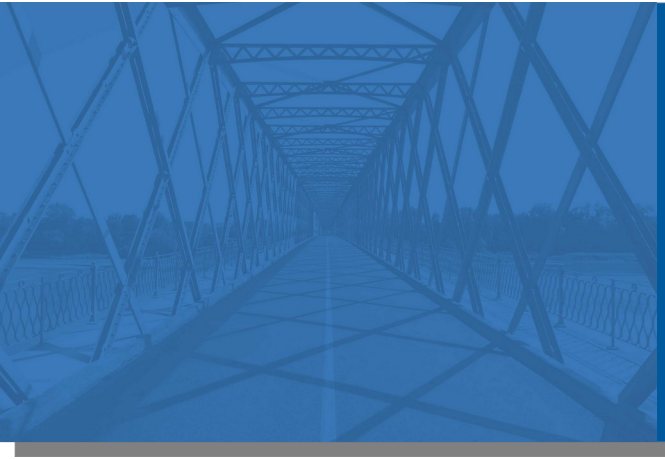
- Nombreuses grosses tables d'historiques avec colonnes d'horodatage des insertions, modifications, suppressions
- 1 grosse table en insertion seule avec colonne date d'insertion
- Copie initiale 19h23
  - 20 sessions
  - 76 tables avec découpage des 5 plus grosses
  - Création des index des tables copiées
- Copie différentielle 3h43
  - 13 sessions
  - Découpage de 2 autres grosses tables
  - Contrôle des nombres de lignes sur les tables anticipées et/ou découpées


DALIBO

DALIBO
L'expertise PostgreSQL

Philippe Beaudoin

philippe.beaudoin@dalibo.com
07 69 14 67 21



# Merci pour votre attention