



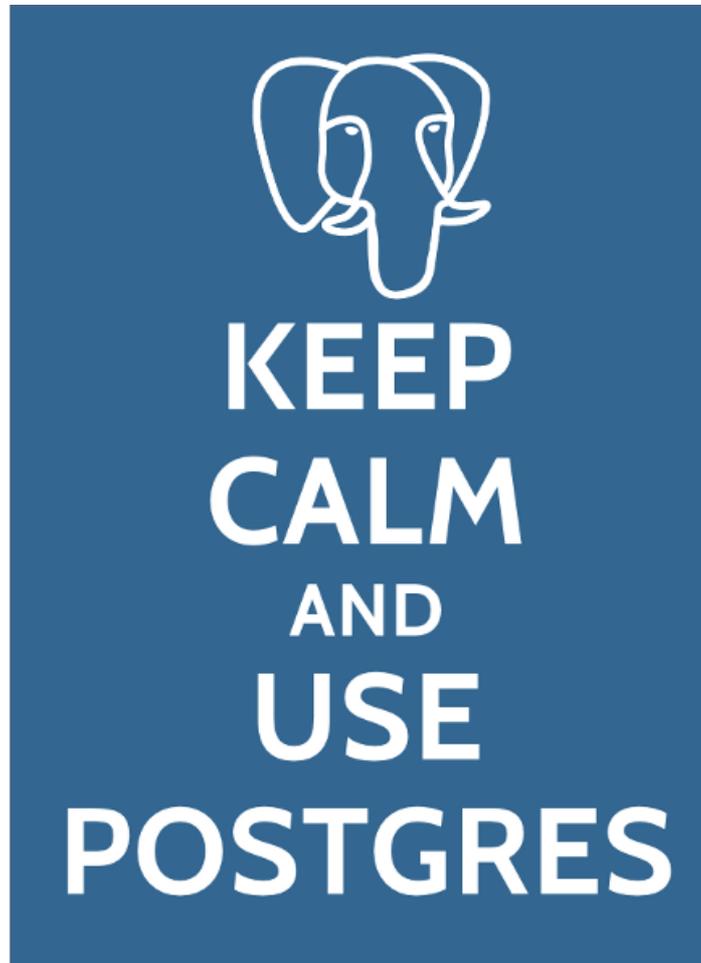
## **Nouveautés de PostgreSQL 9.4**



## Table des matières

Nouveautés de PostgreSQL 9.4.....	4
1 Bonjour !.....	4
1.1 Licence Creative Commons CC-BY-NC-SA.....	5
2 PostgreSQL 9.4 arrive à grand pas !.....	5
3 Au menu.....	6
4 Rappels.....	6
5 versions majeures.....	7
6 Versions Mineures.....	7
7 Versions Actuelles.....	7
8 Recap 9.0 / 2010.....	8
9 Recap 9.1 / 2011.....	8
10 Recap 9.2 / 2012.....	8
11 Recap 9.3 / 2013.....	9
12 Axe 1 - Performances.....	9
13 Index GIN.....	9
14 Index GIN : exemple.....	10
15 prewarm.....	10
16 Prewarm : exemple.....	10
17 EXPLAIN : plus de détails.....	11
18 auto_explain : tracer les triggers.....	11
19 Axe 2 : SQL.....	11
20 FILTER.....	12
21 FILTER : Avant.....	12
22 FILTER : après.....	12
23 WITHIN GROUP.....	13
24 WITHIN GROUP : mediane.....	13
25 WITHIN GROUP : mode.....	13
26 Vues matérialisées.....	14
27 Triggers sur tables externes.....	14
28 WITH ORDINALITY.....	14
29 pl/pgsql stacktrace.....	15
30 NoSQL : JSONB.....	15
31 hstore.....	16
32 Axe 3 : Administration.....	16
33 alter system.....	16
34 alter system.....	17
35 options des tablespaces.....	17
36 move tablespaces.....	17
37 PITR : pg_base_backup.....	18
38 PITR : recovery_target.....	18
39 vacuum.....	18
40 Replication : time delay.....	19
41 Monitoring.....	19
42 Et aussi.....	19
43 Regressions.....	20

44	Encore un peu de patience !.....	20
45	Réplication logique.....	20
46	Parallelisme.....	21
47	UPSERT / MERGE.....	21
48	Bilan.....	21
49	pour aller plus loin.....	22



# Nouveautés de PostgreSQL 9.4

## 1 Bonjour !



- Damien Clochard
- Directeur des Opérations de DALIBO
- Administrateur de [www.postgresql.fr](http://www.postgresql.fr)
- Fondateur de PostgreSQL Magazine
- Twitter : @daamien

## 1.1 Licence Creative Commons CC-BY-NC-SA



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

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

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

Vous êtes libres 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.

À chaque réutilisation ou distribution de cette création, vous devez faire apparaître clairement au public les conditions contractuelles de sa mise à disposition. La meilleure manière de les indiquer est un lien vers cette page web.

Chacune de ces conditions peut être levée si vous obtenez l'autorisation du titulaire des droits sur cette œuvre.

Rien dans ce contrat ne diminue ou ne restreint le droit moral de l'auteur ou des auteurs.

Le texte complet de la licence est disponible à cette adresse:

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/legalcode>

## 2 PostgreSQL 9.4 arrive à grand pas !



- Développement commencé en mai 2013
- Environ 1 an de travail
- Actuellement en phase *beta*
  - → Testez-la par vous mêmes !
- Sortie prévue pour l'automne 2014

## 3 Au menu



- Recap des épisodes précédents...
- Rappel sur le cycle des versions
- Axe 1 : Plus de performances
- Axe 2 : Nouveautés SQL
- Axe 3 : Administration simplifiée
- ...

## 4 Rappels



- Postgres est le SGBD le plus dynamique du marché
- Le cycle des versions est stable et simple
- 1 version majeure par an
- 4-5 versions mineures par an
- Chaque version majeure est supportée 5 ans

## 5 versions majeures



- Désignée par les 2 premiers chiffres : 8.4 / 9.3 / etc.
- Nouveautés + Améliorations des perfs
- Attention ! Risques d'incompatibilités
- MAJ → dump/restore ou pg\_upgrade
- Plusieurs stratégies possibles :
  - le grand saut : 1 MAJ tous les 5 ans
  - au fil de l'eau : 1 MAJ tous les ans

## 6 Versions Mineures



- Désignée par 3 nombres : 8.4.16 / 8.4.17 / etc.
- Correctifs de bugs et de failles de sécurité
- Aucun risque d'incompatibilité
- MAJ → arrêt/redémarrage
- 1 seule stratégie :
  - mettre à jour dès que possible !

## 7 Versions Actuelles



- 8.4(.21) jusqu'en juillet 2014
- 9.0(.17) jusqu'en septembre 2015
- 9.1(.13) jusqu'en septembre 2016
- 9.2(.8) jusqu'en septembre 2017
- 9.3(.4) jusqu'en septembre 2018
- 9.4(beta) jusqu'en septembre 2019

## 8 Recap 9.0 / 2010



- Hot Standby
- Streaming Replication
- Version 64 bit pour Windows

## 9 Recap 9.1 / 2011



- Réplication synchrone
- Tables non-journalisées (Unlogged Tables)
- Extensions + PGXN
- Tables externes (Foreign Data Wrappers)

## 10 Recap 9.2 / 2012



- Performances
- JSON
- index-only scans

## 11 Recap 9.3 / 2013



- Postgres foreign data wrapper
- Vues matérialisées (presque)
- Checksums

## 12 Axe 1 - Performances



- Index GIN
- “Préchauffage”
- Améliorations de la commande EXPLAIN
- module auto\_explain

## 13 Index GIN



- Generalized Inverted iNdex
- Recherche : plein texte, tableau, clef-valeur, json, xml
- Nouveau Format
  - index entre 2 à 10 fois plus petits

<http://hlinnaka.iki.fi/presentations/NordicPGDay2014-GIN.pdf>

## 14 Index GIN : exemple



→ Tests réalisés par Heikki Linnakangas (VMWare)

- Table : 346 MB
- Index B-tree : 214 MB
- Index GIN (9.3) : 58 MB
- Index GIN (9.4) : 11 MB

## 15 prewarm



- “Préchauffez” votre cache
- Au redémarrage, le cache est vide
  - les perfs sont moins bonnes
- Idée : Charger le cache Postgres
  - avec les données d'une ou plusieurs tables
  - utile pour les serveurs standby
- Possible avant ...
  - `find $PGDATA/base -type f -exec cat {} +`
  - `pgfincore`

## 16 Prewarm : exemple

```
# CREATE EXTENSION pg_prewarm
CREATE EXTENSION

# SELECT pg_prewarm('messages');
pg_prewarm
-----
      127855
(1 ROW)
```

## 17 EXPLAIN : plus de détails



- Affiche les colonnes de regroupements sur les noeuds Agg et Group
- Similaire à ce qui est affiché sur les noeuds Sort
- Temps consommé par le planificateur

## 18 auto\_explain : tracer les triggers



- Extension qui trace les plans des requêtes lentes
- en mode `log_analyze` :
  - EXPLAIN ANALYZE sur les requêtes
  - Jamais en production !
- nouvelle option : `log_triggers`
  - → infos sur l'exécution des triggers

## 19 Axe 2 : SQL



- FILTER
- WITHIN GROUP
- Vues Matérialisées
- WITH ORDINALITY
- pl/psql stacktrace
- et un peu de NoSQL !

## 20 FILTER



- Issu du standard SQL
- Filtrer des agrégats (sum / count)
- Avant 9.4 : possible mais compliqué

## 21 FILTER : Avant

```
SELECT
  SUM (
    CASE WHEN m.etat = 'lu'
    THEN 1
    ELSE 0 END),
  SUM (
    CASE WHEN m.etat = 'supprimé'
    THEN 1
    ELSE 0 END)
FROM messages m
WHERE ....
```

## 22 FILTER : après

```
SELECT
  COUNT(*) FILTER (WHERE m.etat = 'lu')
  COUNT(*) FILTER (WHERE m.etat = 'supprimé')
FROM messages m
WHERE ....
```

## 23 WITHIN GROUP



- Opérations sur des “ensembles ordonnés”
- Cas d'écoles
  - Calculer une médiane
  - Trouver la valeur la plus répandue

## 24 WITHIN GROUP : mediane

```
SELECT
  percentile_cont(0.5)
  WITHIN GROUP (ORDER BY salaire)
FROM employes;
```

```
percentile_disc
-----
                1712
(1 ROW)
```

```
SELECT
  percentile_cont(array[0.05,0.95])
  WITHIN GROUP (ORDER BY salaire)
FROM employes;
```

## 25 WITHIN GROUP : mode

```
SELECT
  mode() WITHIN GROUP (ORDER BY couleur_yeux)
FROM employes;
```

```
mode
-----
marron
(1 ROW)
```

## 26 Vues matérialisées



- Présentes depuis la version 9.2 !
- Oui mais...
  - un verrou exclusif est posé à chaque REFRESH
  - impact négatif sur les perfs
  - et donc : intérêt limité
- REFRESH MATERIALIZED VIEW CONCURRENTLY

## 27 Triggers sur tables externes



- SQL/MED : Foreign Data Wrappers
- Connecter Postgres à des stockages “distants”
- Accessibles en écriture depuis version 9.3
- 9.4 : triggers sur les tables externes
- Utile pour tracer l'activité

<http://michael.otacoo.com/postgresql-2/postgres-9-4-feature-highlight-trigger-foreign-tables/>

## 28 WITH ORDINALITY

- équivalent de row\_number() mais pour les fonctions
- très utile avec unnest()

```
SELECT *
FROM unnest(array['H', 'T', 'V'])
WITH ORDINALITY;
```

unnest	ordinality
H	1
T	2

v | 3

## 29 pl/pgsql stacktrace

```
CREATE OR REPLACE FUNCTION public.foo() RETURNS integer AS
DECLARE
stack text;
BEGIN
GET DIAGNOSTICS stack = PG_CONTEXT;
RAISE NOTICE E'--- Call Stack ---\n%', stack;
RETURN 1;
END;
$$ LANGUAGE plpgsql;
```

## 30 NoSQL : JSONB



- format JSON : flexible, puissant... et lent
- nouveau format JSONB
  - stockage binaire
  - profite des optimisations GIN
  - plus rapide que mongodb sur certains tests

## 31 hstore



- Stockage clef/valeur
- Créé 2003 / Essor en 2011
- Avec PostgreSQL 9.4 :
  - `nested-hstore` : fonctionnalité refusée
  - opérateurs de conversions

## 32 Axe 3 : Administration



- ALTER SYSTEM
- Tablespaces
- SAuvagardes ( Base backup / PITR )
- Vacuum
- Réplication
- Monitoring

## 33 alter system

Jusqu'ici on pouvait faire :

```
SET work_mem = 512MB;  
ALTER USER paul SET work_mem = 512MB;
```

Mais avec une portée limitée :

- utilisateur
- une base
- une session

<http://www.depsz.com/2014/01/09/waiting-for-9-4-add-alter-system-command-to-edit-the->

[server-configuration-file/](#)

## 34 alter system

Avec PostgreSQL 9.4

```
ALTER SYSTEM paul SET work_mem = 512MB;  
SELECT pg_reload_conf();
```

- Contrôle si la valeur est cohérente
- Certains paramètres nécessitent un redémarrage

## 35 options des tablespaces

```
CREATE TABLESPACE ssd  
LOCATION '/ssd'  
WITH (random_page_cost = 1.1);
```

## 36 move tablespaces

```
ALTER TABLESPACE pg_default  
MOVE INDEXES TO ssd;
```

```
ALTER TABLESPACE ssd  
MOVE ALL TO san;
```

<http://www.depesz.com/2014/01/23/waiting-for-9-4-tablespace-changes/>

## 37 PITR : pg\_base\_backup



- Outil de sauvegarde niveau fichier
- Très utile pour le PITR
- Avec la version 9.4
  - Gestion plus souple des tablespaces
  - option `-max-rate` pour soulager le système

[http://www.depesz.com/2014/03/05/waiting-for-9-4-allow-base\\_backup-to-be-throttled/](http://www.depesz.com/2014/03/05/waiting-for-9-4-allow-base_backup-to-be-throttled/)

[http://www.depesz.com/2014/03/04/waiting-for-9-4-pg\\_basebackup-add-support-for-relocating-tablespaces/](http://www.depesz.com/2014/03/04/waiting-for-9-4-pg_basebackup-add-support-for-relocating-tablespaces/)

## 38 PITR : recovery\_target



- option PITR (`recovery.conf`)
- Jusqu'ici plusieurs strategies
  - par défaut : rejouer tous les WALs disponibles
  - `time` : on rejoue jusqu'à une date
  - `xid` : on rejoue jusqu'à une transaction
  - `name`: on rejoute jusqu'à un point de restauration
- nouvelle option : `immediate`
  - on rejoue le strict nécessaire

## 39 vacuum



- `autovacuum_work_mem`
- `vacuumdb -analyze-in-stages`

## 40 Replication : time delay



- `min_recovery_apply_delay = 2h`
- Système “idiot proof”
- Possible avec le PITR mais plus rapide

## 41 Monitoring



- `pg_stat_statements`
- Affiche le `queryid`
- `pg_stat_archiver`
- Surveiller l'export des WAL

## 42 Et aussi...



- `COPY FORCE NULL`
- suppression de message d'erreur inutiles
- options `WITH CHECK` pour la mise à jour des vues
- et surtout ....
- Support des années à 5 chiffres

## 43 Regressions



- Très peu cette année !
- Authentification Kerberos 5 remplacée par GSSAPI

## 44 Encore un peu de patience !



- parallélisme
- réplication logique
- UPSERT

<http://www.craigkerstiens.com/2014/02/15/PostgreSQL-9.4-What-I-Wanted/>

## 45 Réplication logique



- “Logical Decoding”
- Extraction / Reconstruction des ordres SQL
- Réplication logique
  - plus fine et plus souple que le Hot Standby
  - possible entre différentes version
- A terme :
  - réplication d'un sous-ensemble des tables
  - upgrade majeur sans coupure de service

## 46 Parallelisme



- actuellement : 1 CPU par requête
- Dynamic Background Workers
- Dynamic Shared Memory
- Briques de base pour le traitement parallèle des requêtes

## 47 UPSERT / MERGE



- “UPDATE or INSERT”
- Très compliqué à faire correctement
- INSERT INTO ON DUPLICATE KEY LOCK FOR UPDATE

## 48 Bilan



- Pas de “killer feature”
- Mais des petits choses utiles dans tous les domaines
  - Perf
  - SQL / NoSQL
  - Admin
- Un avenir radieux

## 49 pour aller plus loin



- Wiki : Page “What's new in PostgreSQL 9.4 ?”
- Blog de Michael Paquier
- Blog de “Depesz”
- planet.postgresql.org

- <http://www.craigkerstiens.com/2014/02/02/Examining-PostgreSQL-9.4/>
- [https://wiki.postgresql.org/wiki/What's\\_new\\_in\\_PostgreSQL\\_9.4](https://wiki.postgresql.org/wiki/What's_new_in_PostgreSQL_9.4)
- <http://sql-info.de/de/postgresql/postgresql-94/articles-reports-postgresql-94-new-features.html>
- <http://www.hagander.net/talks/postgresql94.pdf>
- <http://fr.slideshare.net/EnterpriseDB/a-peek-in-the-elephants-trunk>