

Tungsten: une implémentation du futur clustering de PostgreSQL

Stephane Giron, Gilles Rayrat
Continuent.

Au menu...

- / **Introductions**
- / **Le clustering pour tous**
- / **Présentation de Tungsten**
- / **Intégration PostgreSQL**
- / **Questions et commentaires**

A propos de Continuent

/ **Notre Domaine:** Disponibilité continue des données

/ **Notre Solution**

- Continuent Tungsten (Réplication de bases Maître/Esclave)

/ **Notre Valeur Ajoutée:**

- S'assurer que les données sont disponibles où vous voulez, quand vous voulez
- Coût: moins de 20% du prix des solutions similaires

/ **Notre Expertise Technique**

- Réplication de bases
- Administration de clusters de bases de données
- Connectivité pour les applications

/ **Notre Partenaire**

- 2ndQuadrant et Simon Riggs (thanks, Simon)



Introduction
Le clustering pour tous
Présentation de Tungsten
Intégration PostgreSQL
Questions et commentaires

Définition

**Cluster: Un groupe de machines
connectées en réseau
travaillant conjointement pour
accomplir une tâche utile**

2005 - 2015: Changement Rapide de Techno

/ Plus de 95% des applis ne requièrent qu'une seule machine pour leur base

- Processeurs multi-core
- Mémoire à bas prix
- SSD (solid state devices)

/ Les infrastructures partagées dominant

- Virtualisation et clouds pour les “petits DBMS”
- Instances partagées pour les ISP/SaaS

/ Augmentation importante des utilisations non-OLTP

- Data stores simples et à coût réduit
- Applis Web, lectures intensives
- Webscale processing

2005 - 2015: Des besoins différents

/ Disponibilité

/ Protection des données

/ Utilisation des ressources

/ Performance

/ Intégration open source/commercial

/ Données distribuées géographiquement

2005-2015: La tendance

/ Réplication synchrone:

- Maître/maître (Postgres-R, Sequoia)
- Disque partagé (Oracle RAC)

/ Réplication asynchrone:

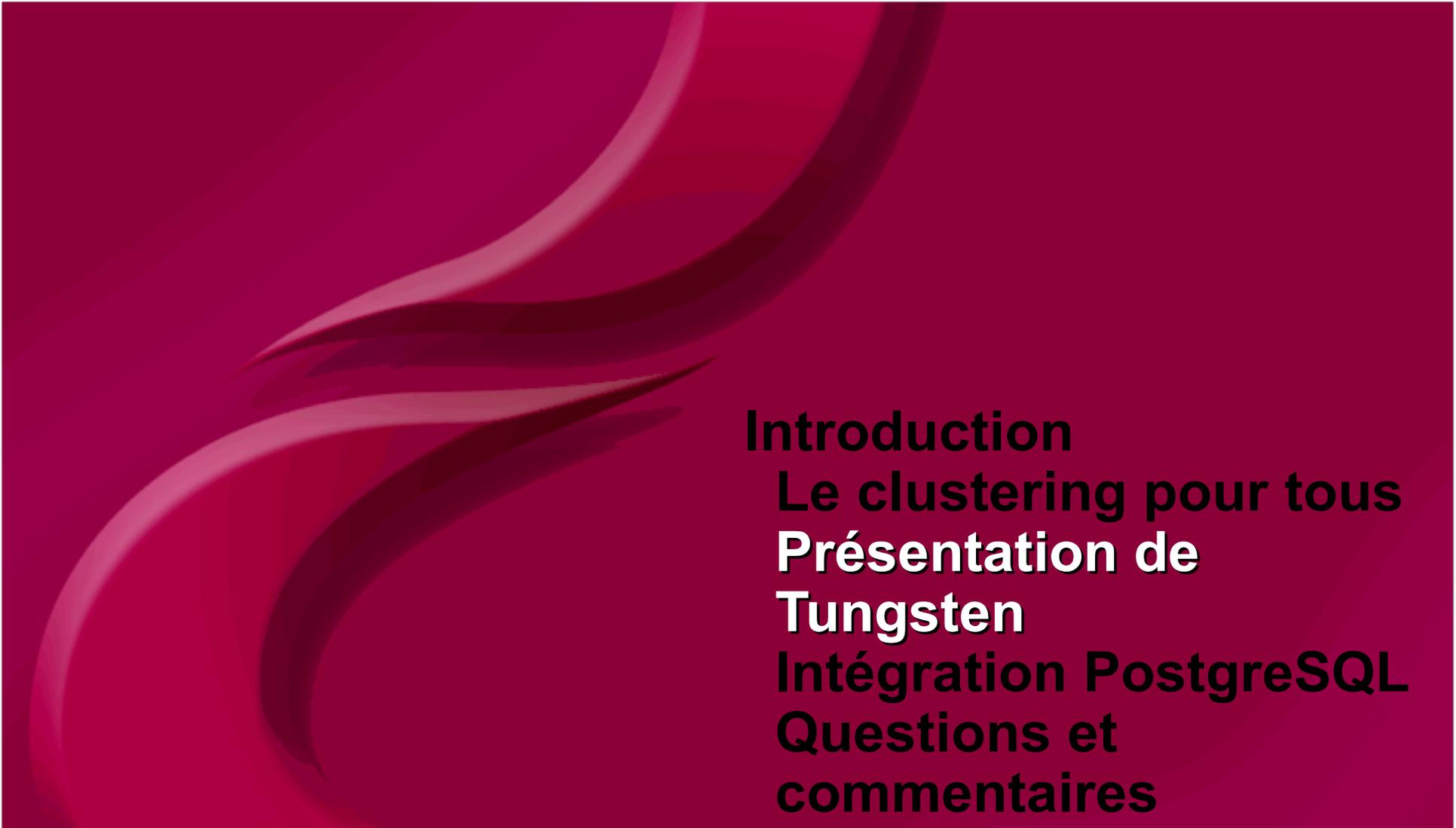
- Maître/esclave (MySQL)
- Consistence in-fine (SimpleDB, BigTable, Bucardo)

/ Admin simplifié:

/ Utilisation efficace de ressources:

- Partitions et modèles multitenants
- Migration vers des performances +, ou -
- Virtualisation

/ Protection de données:



Introduction
Le clustering pour tous
Présentation de
Tungsten
Intégration PostgreSQL
Questions et
commentaires

Tungsten ?

- / **Tungsten est une implémentation de clustering maître/esclave pour :**
 - Protéger les données
 - Maintenir une haute disponibilité
 - Améliorer l'utilisation des ressources
 - Augmenter la performance
- / **Installation et configuration en quelques minutes**
- / **Sauvegarde/restauration intégrés, contrôle de l'intégrité des données**
- / **Failover efficace**
- / **Administration distribuée et par "règles"**
- / **Pas ou peu de changements applicatifs**
- / **Modularité maximale**
- / **Pas de prérequis hardware**

Tungsten Open Source Foundation

/ Tungsten Replicator

- Réplication multi BDD et multi plateformes maître/esclave
- Extensible à d'autres types de réplication

/ Tungsten Connector

- Proxy MySQL/PostgreSQL <> JDBC ultra-rapide

/ Tungsten SQL Router

- Wrapper JDBC pour failover, load-balancing et partitioning transparent haute-performance (ne requiert aucun proxy)

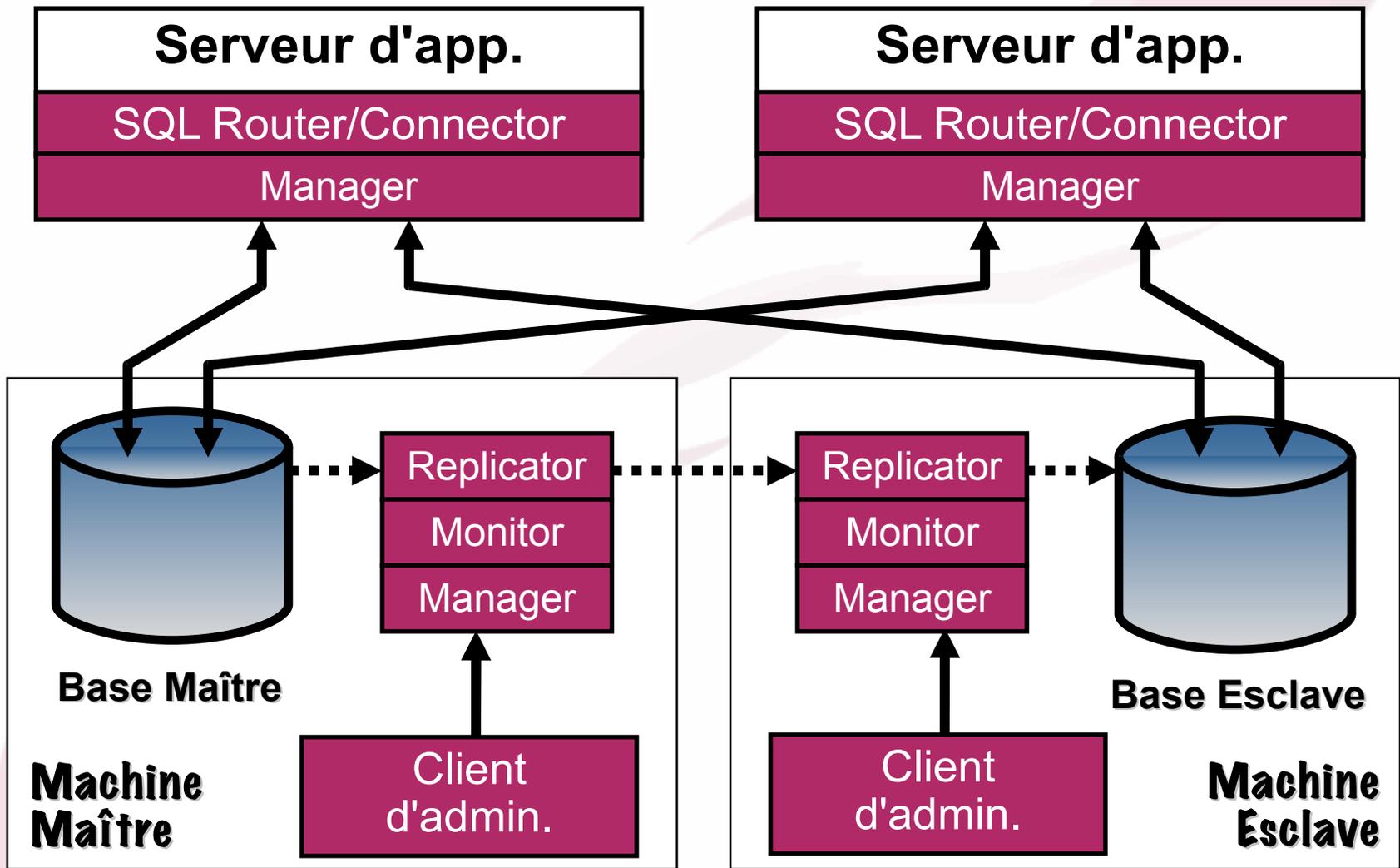
/ Tungsten Manager

- Administration distribuée avec config. automatique à base de règles, pas de SPOF

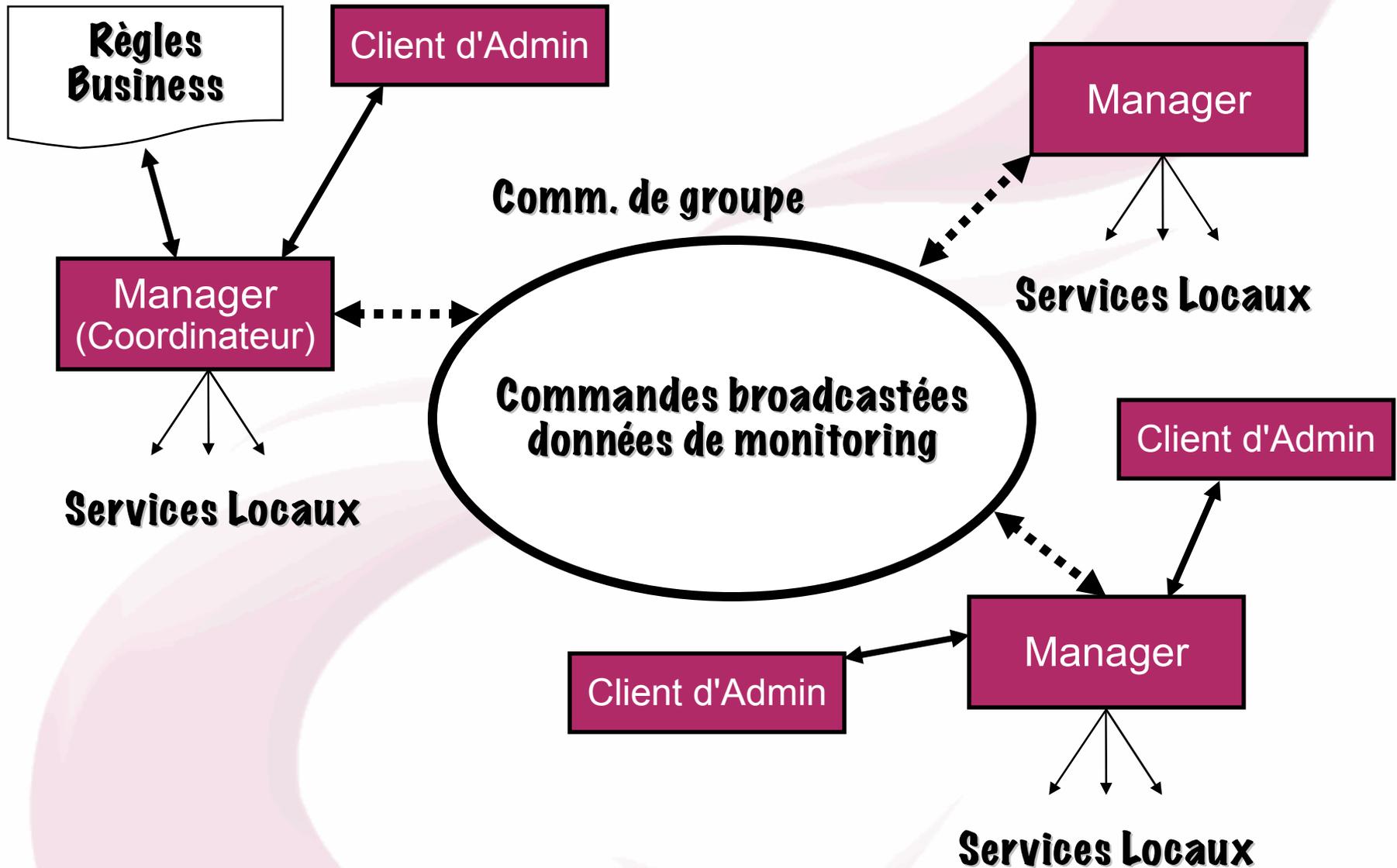
/ Tungsten Monitor

- Mesure la latence et détecte l'état des ressources

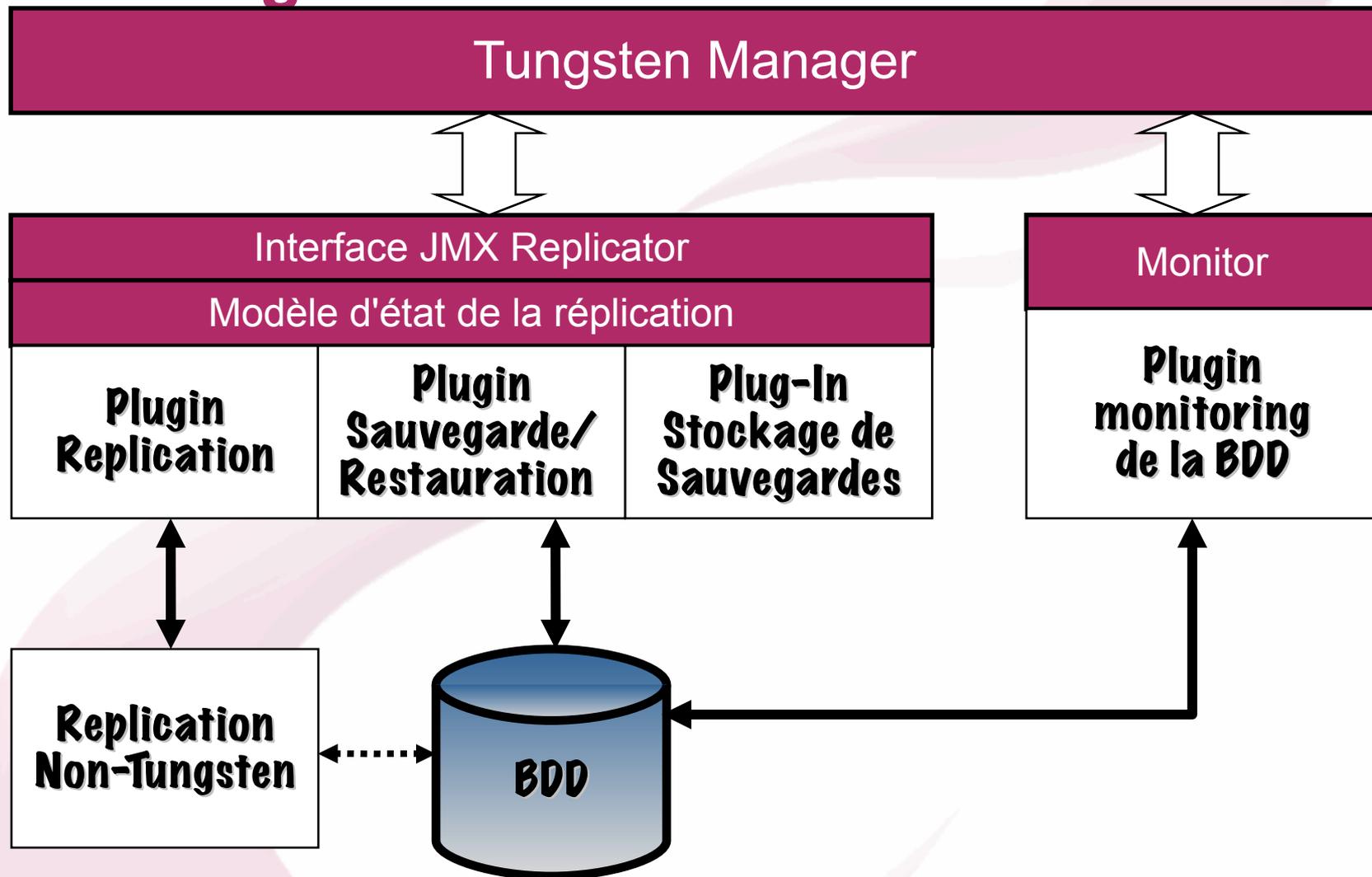
Tungsten En Action



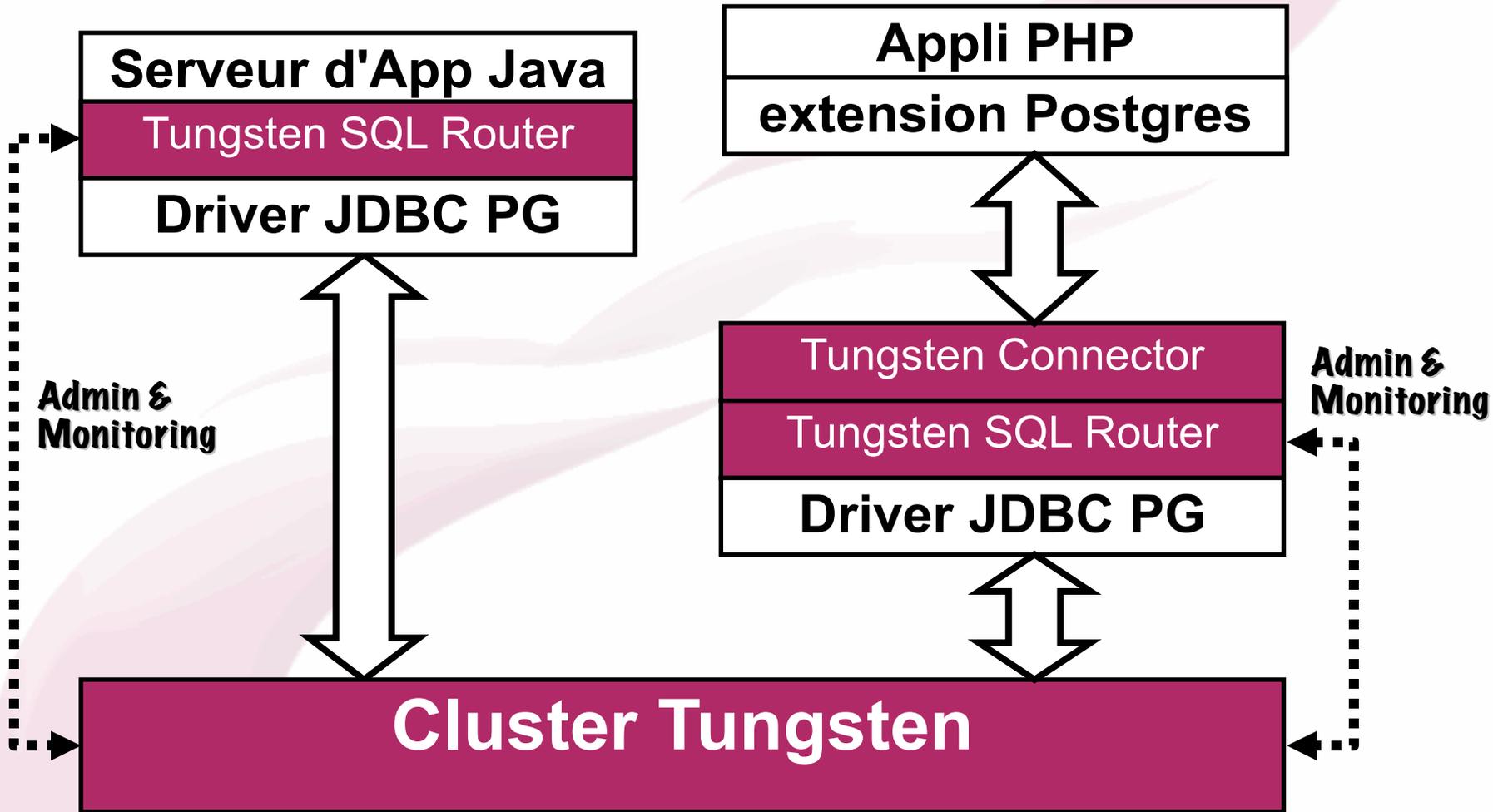
Administration distribuée à base de règles



Open Replicator – pour une réplication non-Tungsten

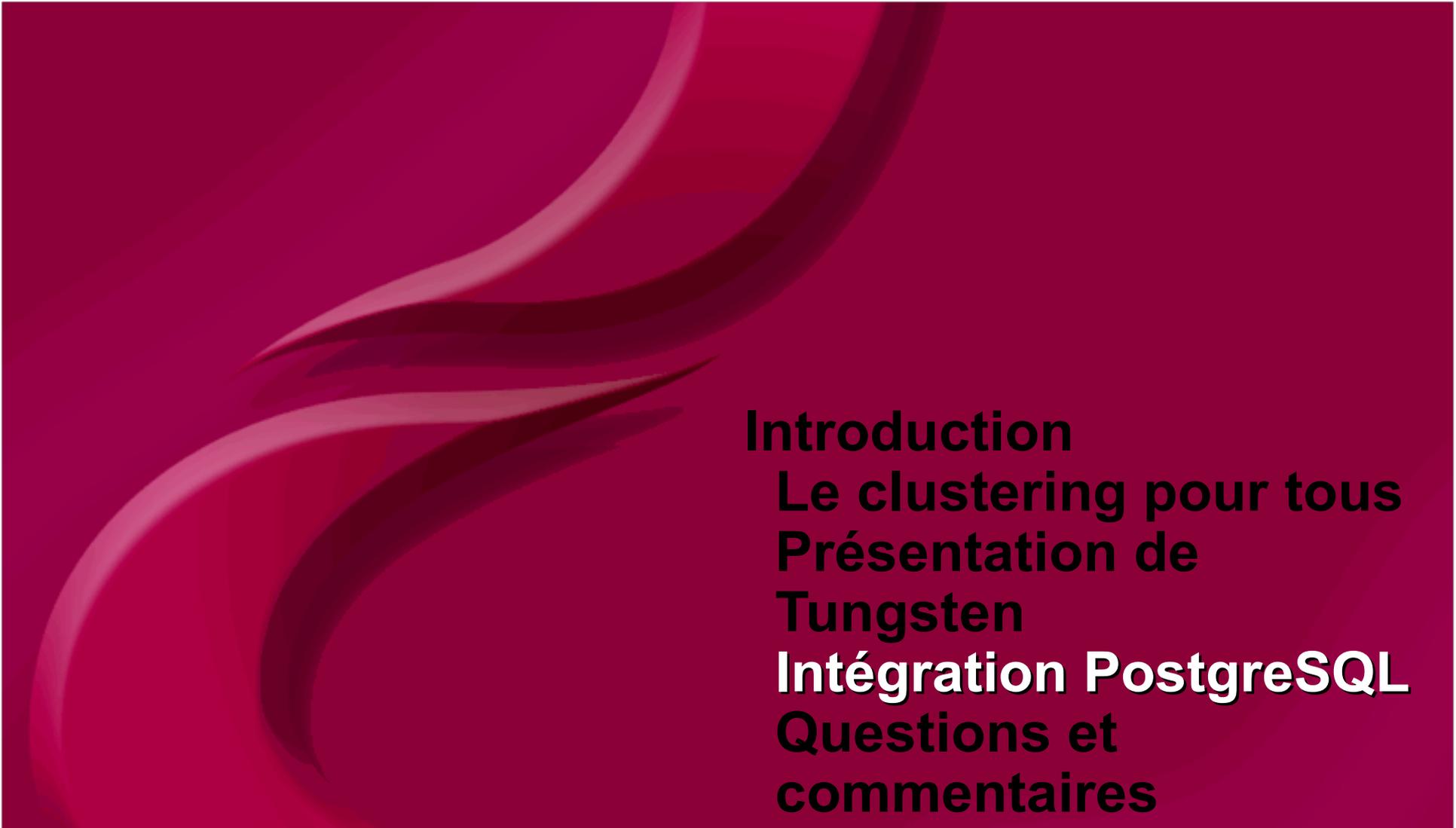


Routage SQL



Ce qu'il faut retenir...

- / **Installation en 15 minutes**
- / **Commandes uniques pour :**
 - Vérifier le status du cluster
 - Sauvegarder une base
 - Restaurer une base
 - Vérifier les données entre réplicas
 - Confirmer le status de la replication
 - Switcher les rôles des serveurs pour maintenance
 - Assurer le failover d'un serveur vers le replica le plus à jour
- / **Découverte automatique des nouveaux réplicas**
- / **Failover automatique en cas de crash d'un serveur**
- / **Procedures de provisioning simples**

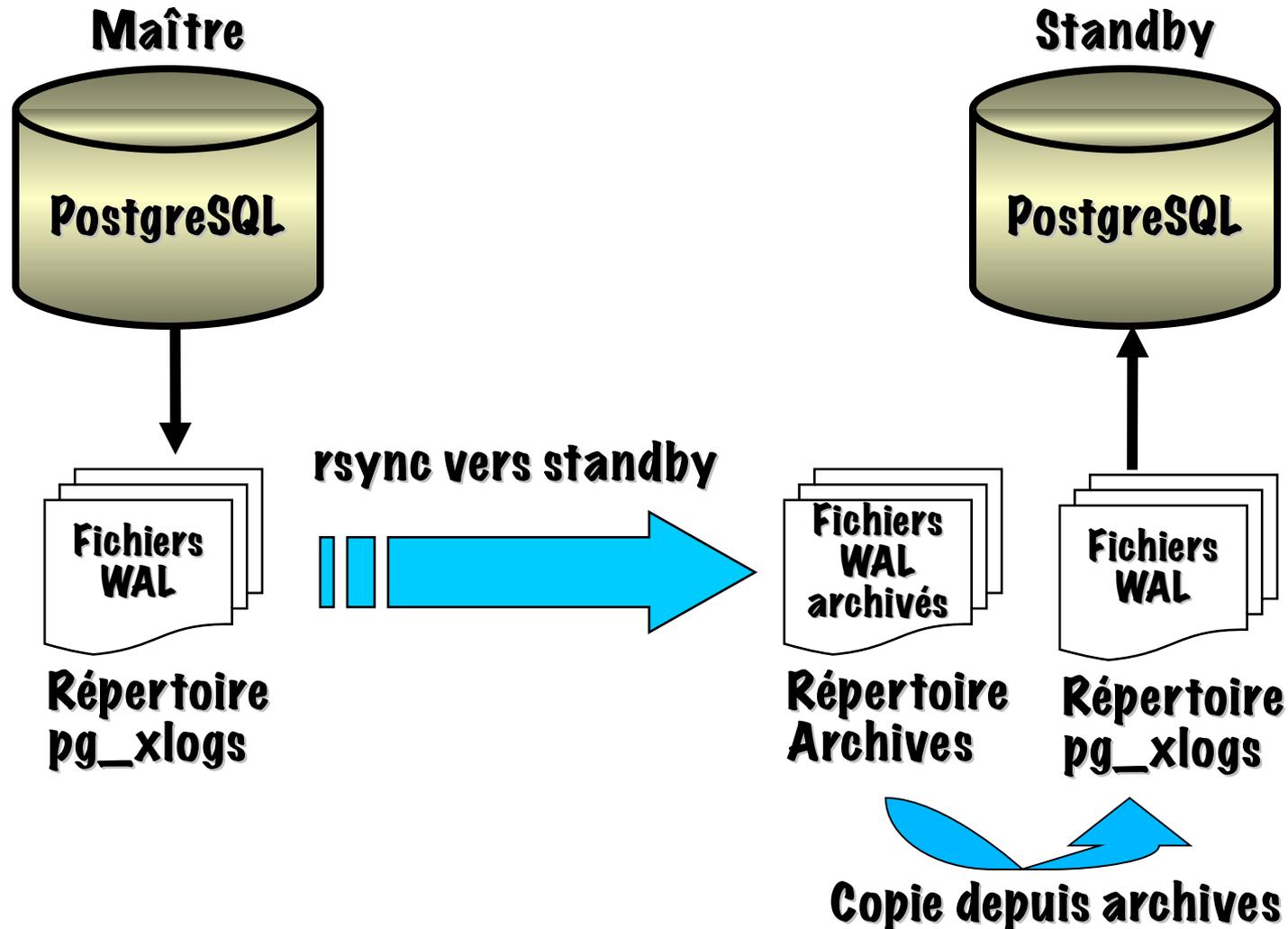


Introduction
Le clustering pour tous
Présentation de
Tungsten
Intégration PostgreSQL
Questions et
commentaires

Tungsten et PostgreSQL

- / **Problème : Nous ne pouvons pas (encore) lire les logs PostgreSQL**
- / **Solution : Utiliser le Warm Standby / PITR pour répliquer les données vers des bases en standby**
 - Bonne disponibilité (basique) / failover rapide
 - Une fois que le hot standby fonctionne, on est presque rendus !
 - Couvre mal les problèmes de maintenance
- / **Solution : Utiliser Londiste pour répliquer vers des “réplicas actifs”**
 - Couvre les problèmes de maintenance et de passage à l'échelle en lecture

Implémentation Warm Standby



Configuration du Warm Standby (Ancienne méthode)

/ Configurer le postgresql.conf du maître (et rebooter)

```
archive_mode = on
archive_command = 'rsync -cz $1 ${STANDBY}:${PGHOME}/archive/$2
%p %f'
archive_timeout = 60
```

/ Configurer le recovery.conf du standby

```
restore_command = 'pg_standby -c -d -k 96 -r 1 -s 30 -w 0 -t
${PGDATA}/trigger.dat ${PGHOME}/archive %f %p %r'
```

/ Provisionner le standby

```
psql# select pg_switch_xlog();
psql# select pg_xlogfile_name(pg_start_backup('base_backup'));
rsync -cva --inplace --exclude=*pg_xlog* ${PGHOME}/
${STANDBY}:${PGHOME}/archive
psql# select pg_xlogfile_name(pg_stop_backup());
```

/ Démarrer le standby, la recovery démarre

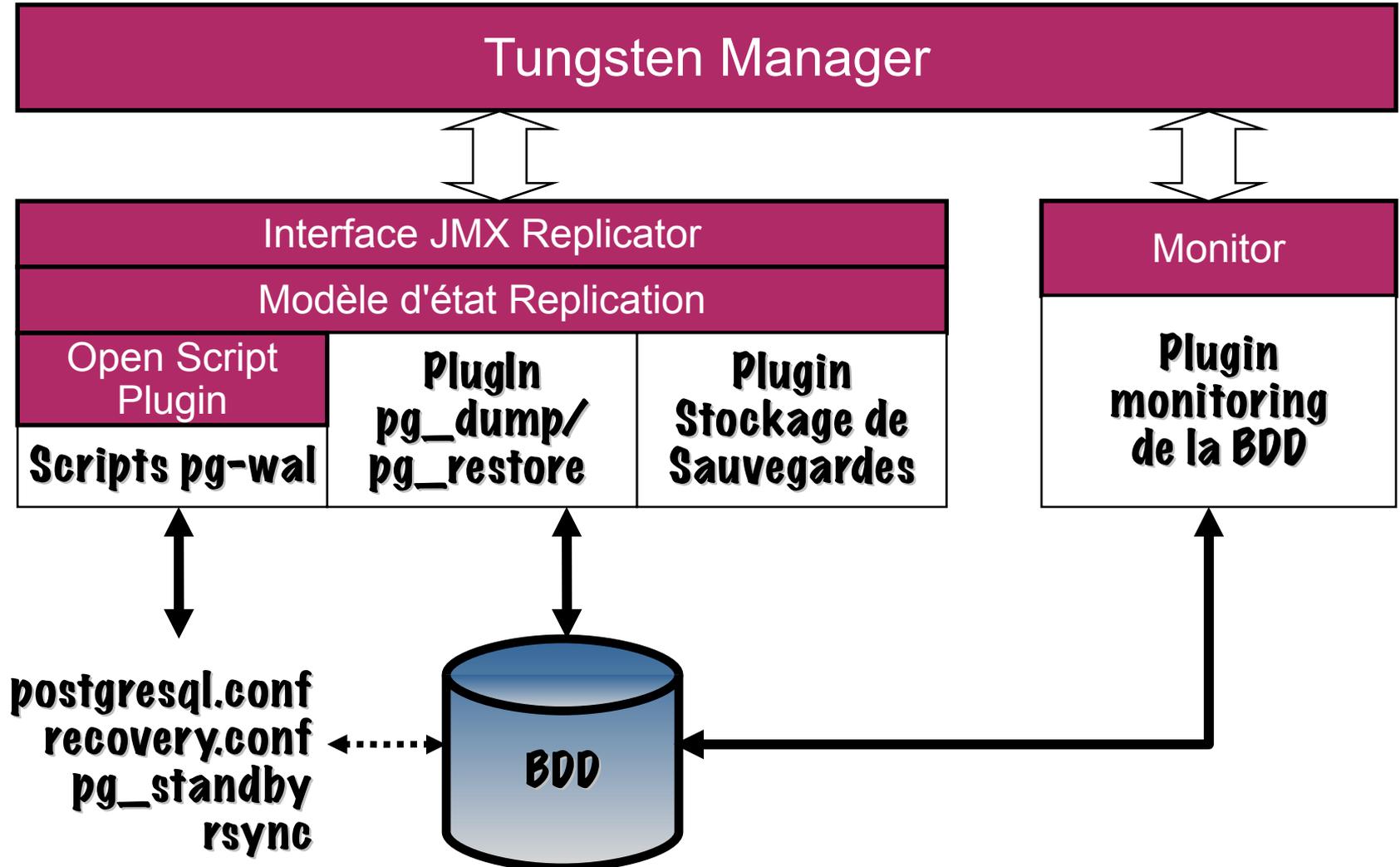
/ touch \${PGDATA}/trigger.dat pour lancer un failover

Warm Standby : ses inconvénients

- / Parfait pour la haute disponibilité, pas pour le passage à l'échelle**
- / Perte de données possibles en cas de failover non planifié !**
- / La restauration du maître implique un re-provisioning**
- / Installation et administration beaucoup plus difficile qu'il n'y paraît**
- / Le monitoring devient un élément critique**
- / Impossible d'ouvrir le standby avant un failover**
- / Nécessité de vérifier que tous les logs ont été lus avant de faire un failover**

Reste, malgré tous ces inconvénients, une excellente solution !!!

Implémentation du Warm Standby dans Tungsten



Ce qu'il faut retenir

/ **Configuration simple du warm standby**

/ **Commandes uniques pour :**

- Vérifier le status du cluster, y compris les stats de réplication
- Sauvegarder une base
- Restaurer une base
- Provisionner une base
- Vérifier l'intégrité des données entre réplicas
- *Confirmer le status de la réplication*
- *Switcher les rôles des serveurs pour maintenance*
- *Assurer le failover d'un serveur vers le replica le plus à jour*

/ **Découverte automatique des nouveaux réplicas**

/ **Failover automatisé**

Futurs axes de travail

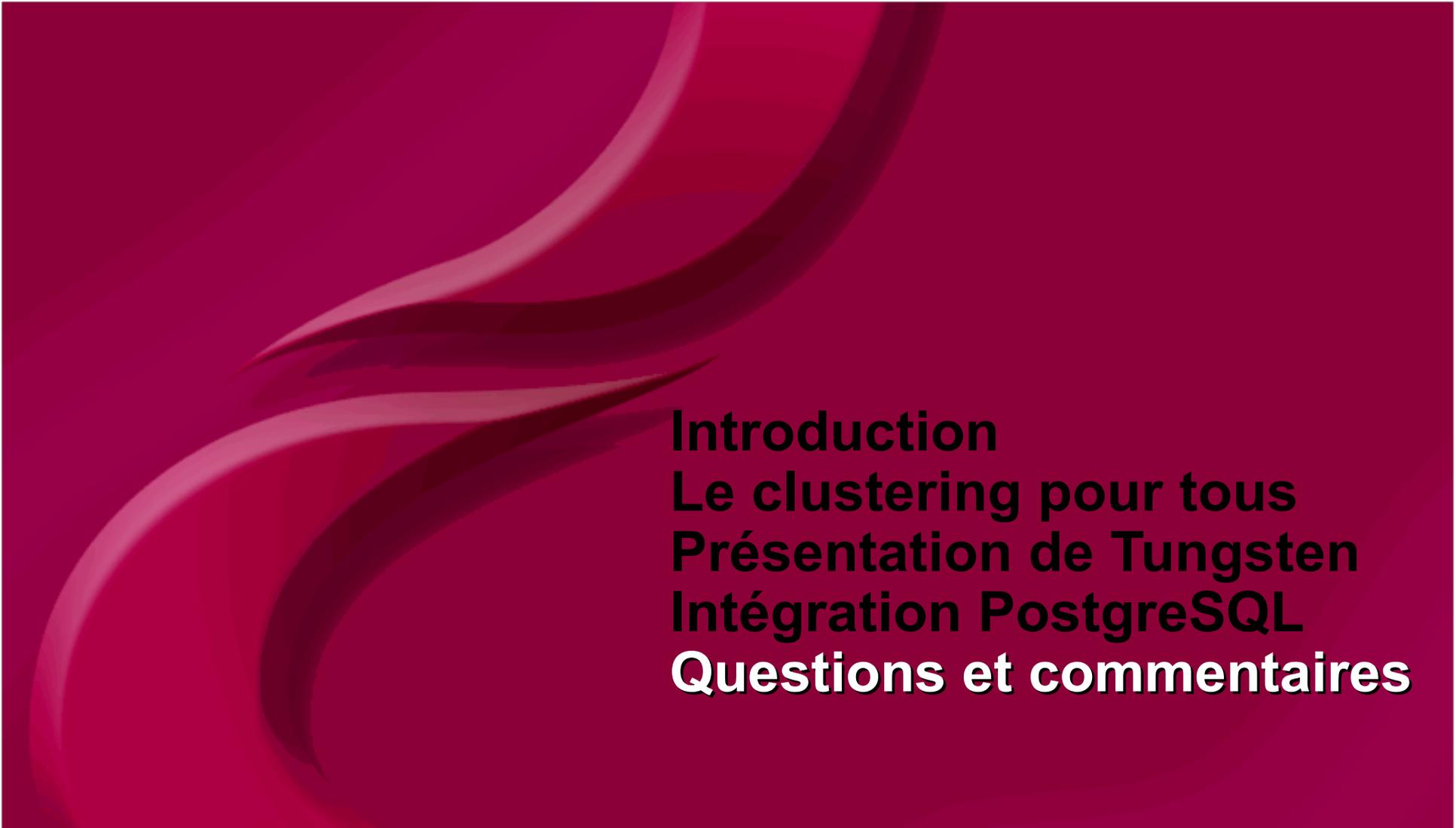
/ Terminer les commandes d'admin du warm standby

- Configuration
- Backup des noeuds esclaves
- Monitoring
- Notifications en cas de crash
- Log Streaming et PITR

/ Implementer Londiste pour disposer de réplicas actifs

/ Lecture directe des logs PostgreSQL

D'autres fonctionnalités comme les IP flottantes



Introduction
Le clustering pour tous
Présentation de Tungsten
Intégration PostgreSQL
Questions et commentaires

En résumé

/ Le clustering change...

- Les nouvelles technologies et les nouveaux besoins changent la donne du clustering

/ Tungsten anticipe

- Les clusters Continuent répondent aux nouveaux besoins mieux que les autres approches

/ Essayez le !

- Tungsten est disponible au téléchargement, n'hésitez pas à nous envoyer votre feedback

Contacts

Maison mère / USA

560 S. Winchester Blvd., Suite 500
San Jose, CA 95128
Tel (866) 998-3642
Fax (408) 668-1009

Europe/Asie

Lars Sonckin kaari 16
02600 Espoo, Finlande
Tel +358 50 517 9059
Fax +358 9 863 0060

e-mails: [stephane.giron](mailto:stephane.giron@continuent.com) [gilles.rayrat](mailto:gilles.rayrat@continuent.com) [robert.hodges](mailto:robert.hodges@continuent.com) (at) continuent.com

Site Continuent :

<http://www.continuent.com>

Site 2ndQuadrant :

<http://www.2ndquadrant.com>