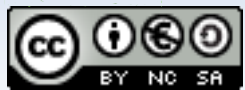
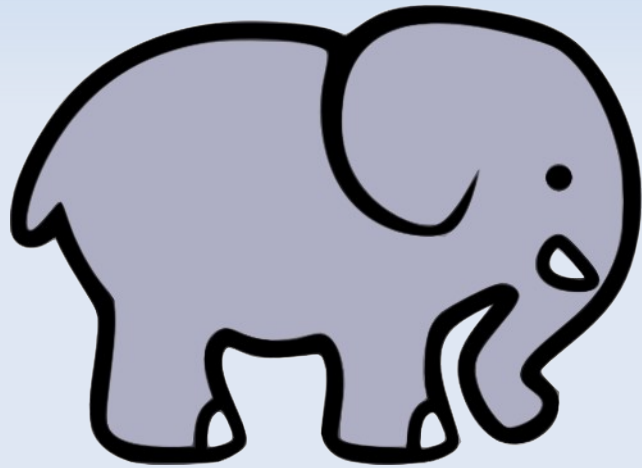
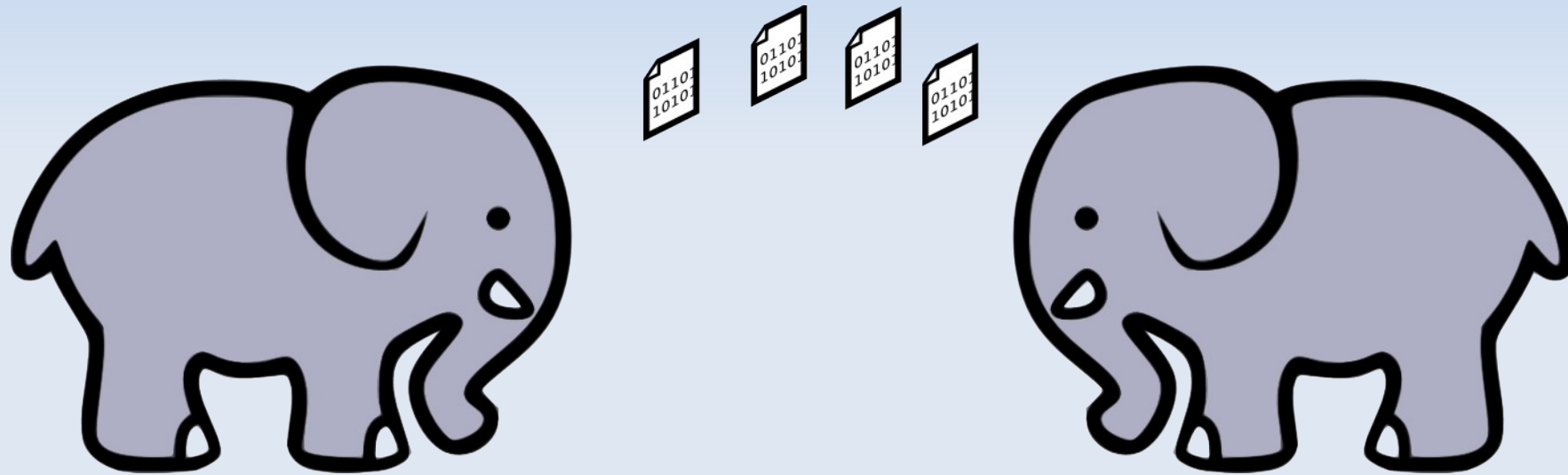


PostgreSQL : PITR et LogShipping



Jean-Christophe Arnu
PostgreSQLFr/CS

PostgreSQL : PITR et LogShipping



Jean-Christophe Arnu
PostgreSQLFr/CS

Contenu

- Rappel des concepts de bases de données
- Qu'est-ce que le PITR?
- Qu'est-ce que le Log Shipping?
- Génération des WAL et restauration (outils)



2008-07-04

PITR et log shipping



Comment fonctionne une base de données ?

- Des transactions
- Des accès disques (parfois beaucoup)
- Des écritures (parfois souvent)
- Des performances



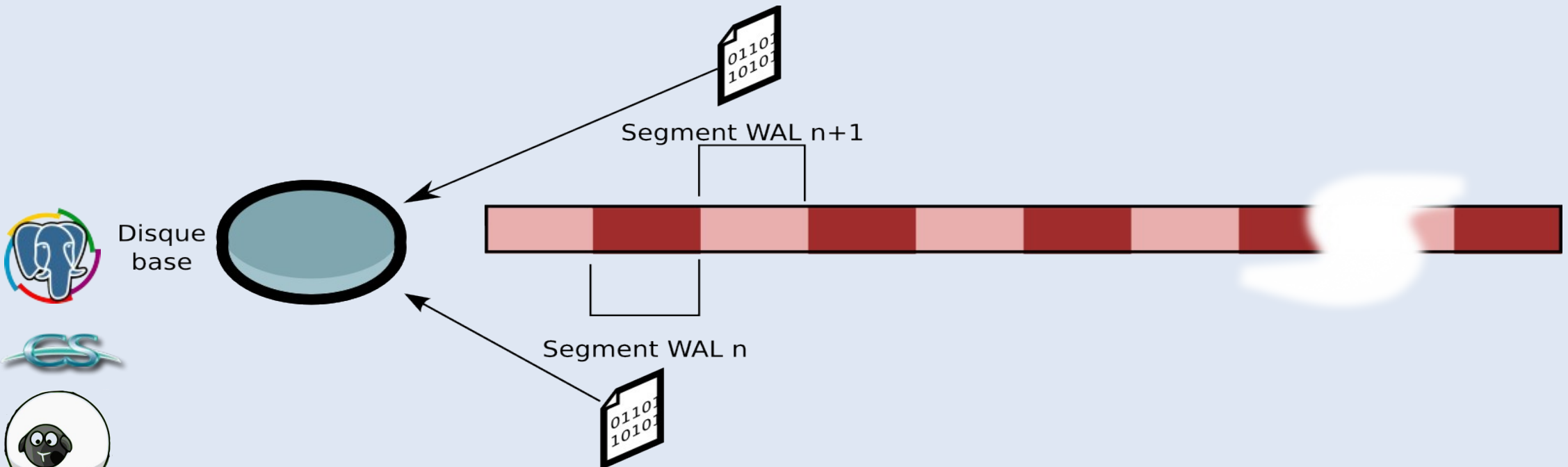
2008-07-04

PITR et log shipping



Améliorer les performances : WAL

- Écrire dans plusieurs fichiers prend du temps
- Mise à jour d'un journal
- Écriture en différé dans les fichiers des tables



Disque base



2008-07-04

PITR et log shipping



WAL et segments

- Un segment WAL = 16Mo
- 3 segments par défaut (`checkpoint_segments`)
- Temps entre deux fichiers WAL : 5 minutes ou `checkpoint_timeout` ou l'appel à `CHECKPOINT`
- Les WAL sont recyclés après écriture de leur contenu dans les fichiers de données de la base

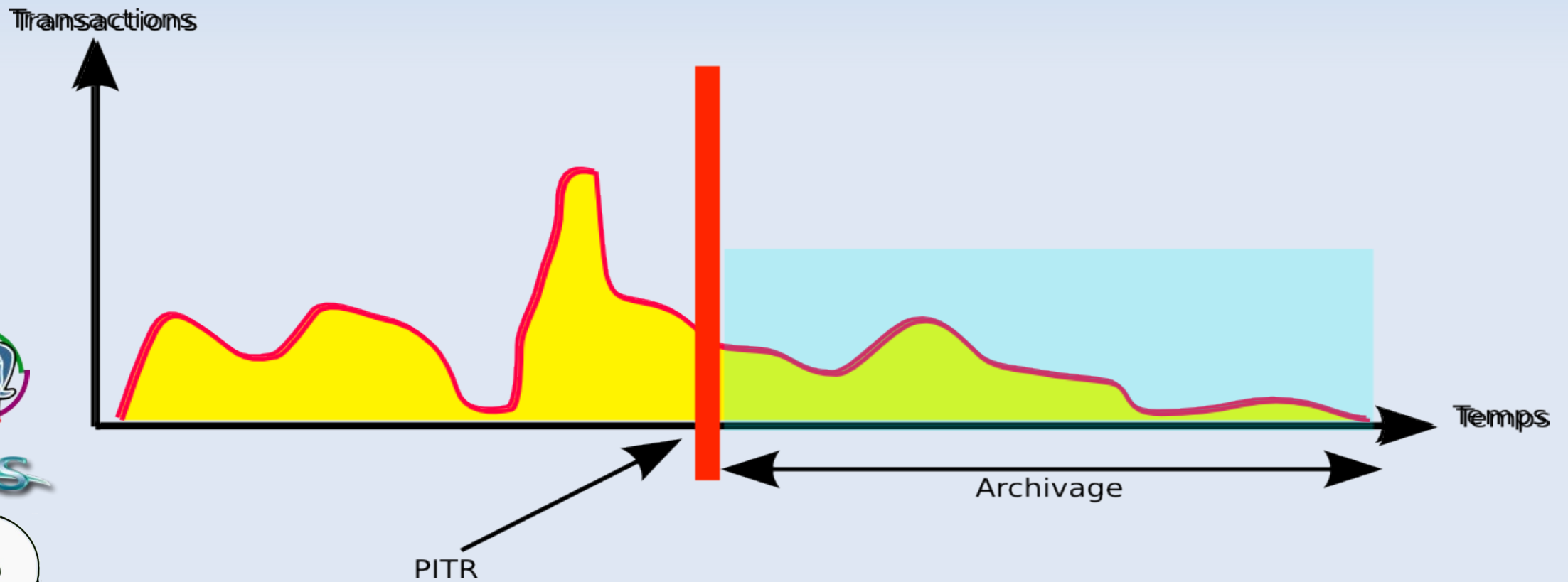


2008-07-04

PITR et log shipping



Le PITR



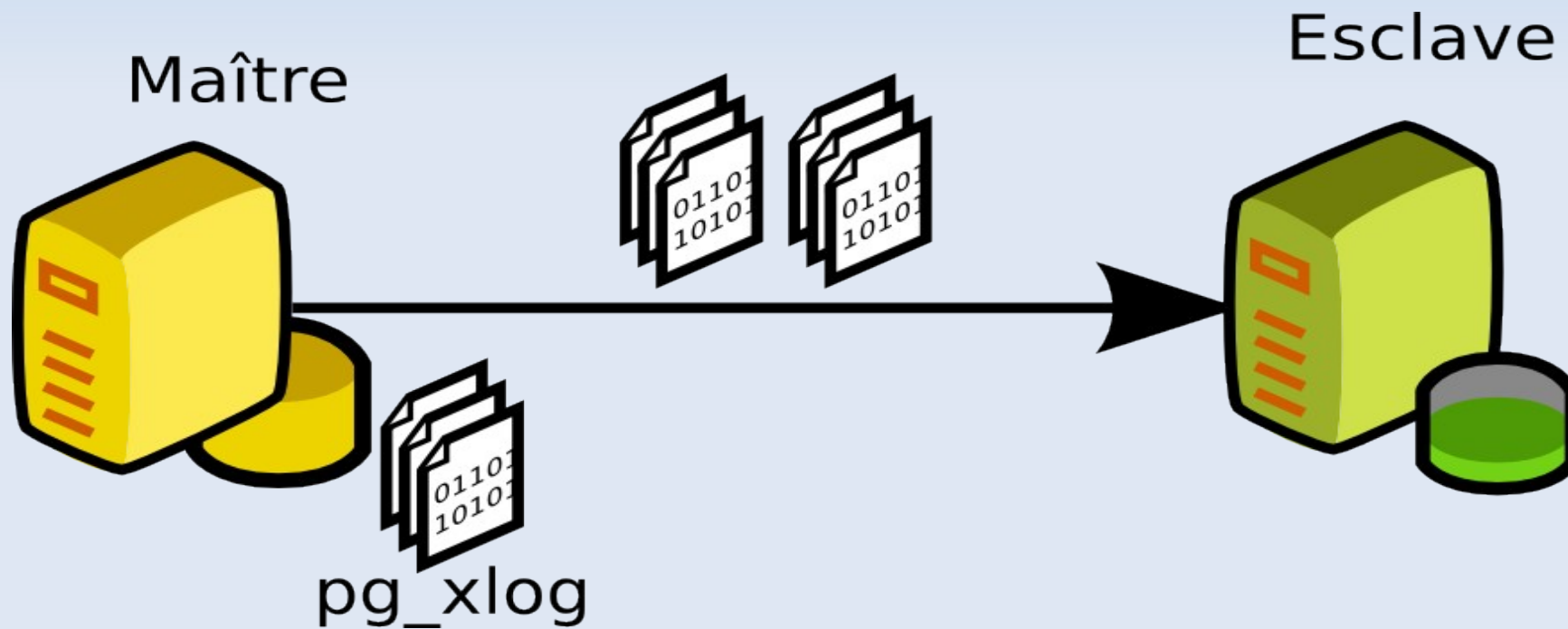
2008-07-04

PITR et log shipping



Le LogShipping

- Grande distribution (presque) gratuite de fichiers de journaux transactionnels



2008-07-04

PITR et log shipping



Configurer l'archivage du WAL

- Tout se trouve dans le fichier postgresql.conf
 - Debian /etc/postgresql/<version>/<cluster>/
 - Standard /var/lib/pgsql/data

- Quelques options :

```
archive_mode = on          # On active l'archivage
archive_command = '...'   # Commande d'archivage
archive_timeout = 60      # Secondes
```



2008-07-04

PITR et log shipping



Le script d'archivage

- `archive_command` doit être personnalisé
- `%p` fichier journal à copier à partir du `pg_xlog`
- `%f` nom du fichier (pour la destination)
- A la base :
 - `cp -i %p /vers/archive/%f`
 - `rsync -a %p login@serveur:/vers/archive/%f`
- Doit retourner 0 en cas de succès et autre chose en cas d'erreur



2008-07-04

PITR et log shipping



Gestion de l'archivage

- Ecriture de l'archive en fonction de la sollicitation en insert, update, delete
- Si pas d'activité pour terminer le log, après `archive_timeout` secondes, passage à un autre xlog
- Si indisponibilité du répertoire cible (disque plein, disque indisponible, répertoire absent, ...)
 - Test toutes les secondes (x3)
 - Puis si toujours des erreurs test toutes les minutes



Surveillance

- => Attention à l'augmentation des logs (`pg_xlog` et `pg_xlog/archive_status`)
- Surveiller ces zones (nagios, ...)



2008-07-04

PITR et log shipping



Utiliser le PITR

- Sans PITR, il faut arrêter la base pour archiver les fichiers de la base!
- Avec le PITR, on définit un point de repère de la base, on fait un backup, sans arrêter la base!
 - Fonction de préparation au backup à chaud :
`pg_start_backup('nom_archive')`
 - Fonction de fin de backup à chaud : `pg_stop_backup()`



pg_start_backup(...)

- Peut prendre un peu de temps pour se terminer
- Effectue un CHECKPOINT pour vider le cache sur disque.
- Crée
`$PGDATA/<version>/<cluster>/backup_label`



2008-07-04

PITR et log shipping



Sauvegarde des fichiers

- Permet de faire une sauvegarde (tar/rsync) de
 - Sous Debian `$PGDATA/<version>/<cluster>`
 - Sous autre systèmes `$PGDATA`
- Sauvegarde à chaud, les clients peuvent toujours modifier/insérer des données et pas d'influence sur la validité de la sauvegarde



pg_stop_backup()

- Est immédiat.
- Crée
`$PGDATA/<version>/<cluster>/backup_label`
`.old`
- On peut, par la suite, se baser sur les fichiers de transactions archivés pour faire de la sauvegarde en continu



2008-07-04

PITR et log shipping



16

Le LogShipping

- On conserve le principe de PITR
- On combine avec un transfert des journaux de transactions d'un serveur à l'autre
- On rejoue ces journaux sur l'esclave
- L'esclave est indisponible (*warm-standby*) jusqu'à une erreur d'exécution de `restore_command` (`<>0`) dans le fichier `$PG_DATA/recovery.conf`



2008-07-04

PITR et log shipping



17

La solution « legacy » pour recovery.conf

- Utiliser cp pour copier les fichiers dans \$PG_DATA/pg_xlog
- pg_standby permet de disposer d'un outil restore_command intelligent (sur l'esclave)
 - Debug ou journal (-d)
 - Copie ou lien (-c ou -l)
 - Chemin vers le répertoire des fichiers archives
 - Temps d'attente entre deux tentatives (-s <temps en s>)
 - Nombre de tentatives (-r <nombre>)



2008-07-04

PITR et log shipping



18

Solution intégrée walmgr

- Walmgr est un outil Skype
- Système gérant l'ensemble des fonctions de PITR, d'archivage et de log shipping
- Effectue l'ensemble des tâches décrites précédemment au travers d'un seul script python
- S'installe sur maître **et** esclave
- Nécessite l'écriture de fichiers de configuration
- Basé sur rsync, ssh, ...



2008-07-04

PITR et log shipping



19

Log shipping avec walmgr

- Envoyer les xlogs
 - Utiliser sync pour un envoi one shot
 - Utiliser syncdaemon pour un envoi au fil de l'eau



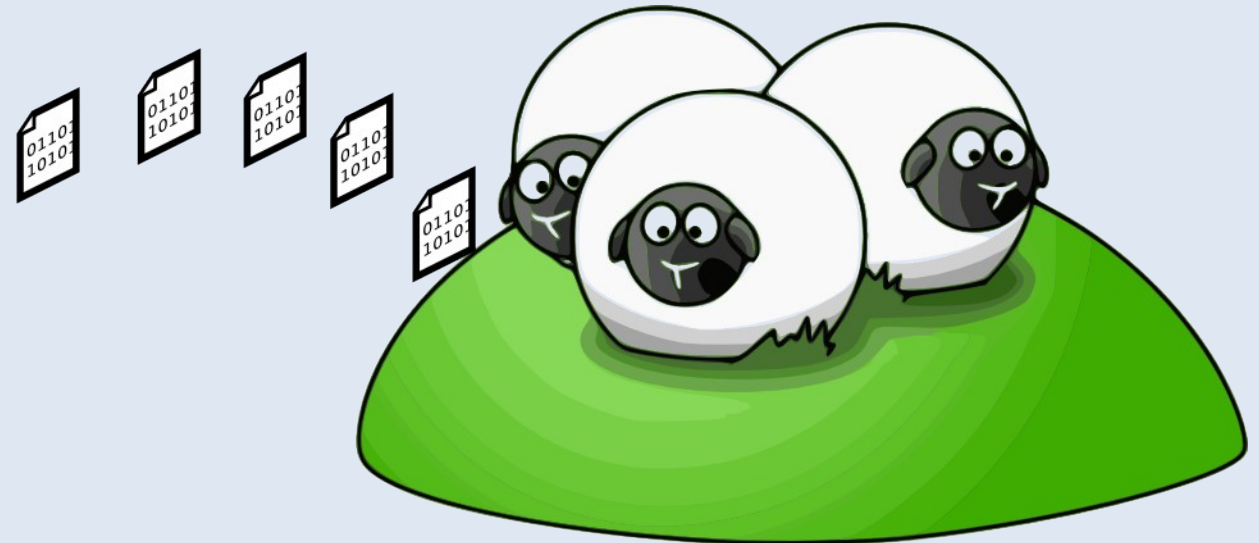
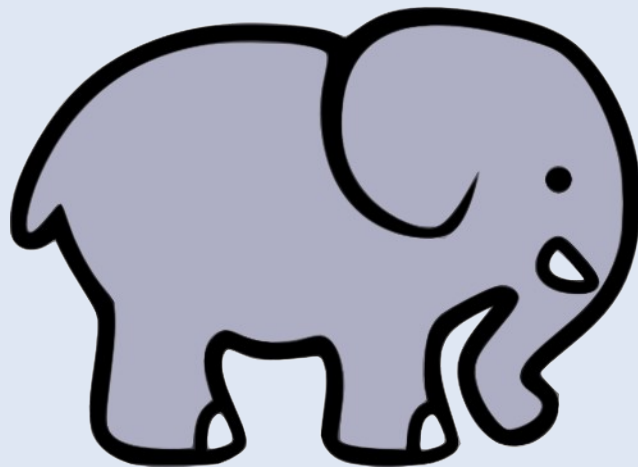
2008-07-04

PITR et log shipping



PITR et LogShipping

- Et « les moutons seront bien gardés » ;-)
- Le PITR et le log shipping constituent une solution de réplication robuste et fiable
- Des mécanismes simples à mettre en œuvre avec différents outils



2008-07-04

PITR et log shipping

Merci

- PostgreSQLFr <http://www.postgresqlfr.org>
- PostgreSQL <http://www.postgresql.org>
- Merci
- Questions ?



2008-07-04

PITR et log shipping

22

