

Disponibilité et Durabilité

Architectures et Réplifications

Dimitri Fontaine

dimitri@2ndQuadrant.fr

7 Juin 2012

1 Présentation

- Préambule
- Disponibilité, Durabilité
- Architectures et Réplifications

2 Architectures et Réplifications

- Augmentation du trafic
- Durabilité des données
- Disponibilité des données
- Disponibilité des services
- Sharding

3 Conclusion

- Rétrospective et avenir de la réplication PostgreSQL
- Questions

Dimitri Fontaine

2ndQuadrant France PostgreSQL Major Contributor

- pgloader, prefix, skytools, debian, ...
- **CREATE EXTENSION**
- *Command Triggers*
- *Bi-Directional Réplication*
- *Partitionnement*



Dimitri Fontaine

2ndQuadrant France PostgreSQL Major Contributor

- pgloader, prefix, skytools, debian, ...
- CREATE EXTENSION
- *Command Triggers*
- *Bi-Directional Réplication*
- *Partitionnement*



Dimitri Fontaine

2ndQuadrant France PostgreSQL Major Contributor

- pgloader, prefix, skytools, debian, ...
- CREATE EXTENSION
- *Command Triggers*
- *Bi-Directional Réplication*
- *Partitionnement*



Retour d'expérience, principalement avec Hi-Media

Hi-Media (200 millions de CA)

- 3 métiers autour de la monétisation web
- Allopass, HiPay: micro paiement sur internet
- Services Télécom
- Régie publicitaire

PostgreSQL au cœur de l'architecture technique

- Réponse aux contraintes
- Adaptation aux changements



Retour d'expérience, principalement avec Hi-Media

Hi-Media (200 millions de CA)

- 3 métiers autour de la monétisation web
- Allopass, HiPay: micro paiement sur internet
- Services Télécom
- Régie publicitaire

PostgreSQL au cœur de l'architecture technique

- Réponse aux contraintes
- Adaptation aux changements



Retour d'expérience, principalement avec Hi-Media

Hi-Media (200 millions de CA)

- 3 métiers autour de la monétisation web
- Allopass, HiPay: micro paiement sur internet
- Services Télécom
- Régie publicitaire

PostgreSQL au cœur de l'architecture technique

- Réponse aux contraintes
- Adaptation aux changements



Retour d'expérience, principalement avec Hi-Media

Hi-Media (200 millions de CA)

- 3 métiers autour de la monétisation web
- Allopass, HiPay: micro paiement sur internet
- Services Télécom
- Régie publicitaire

PostgreSQL au cœur de l'architecture technique

- Réponse aux contraintes
- Adaptation aux changements



Vos données sont votre métier

Comment assurer la *durabilité* et la *disponibilité* de vos données?

Besoin:

- Fiabilité
- Robustesse
- Performances
- Accompagnement du succès commercial
- Continuité et Innovation



Vos données sont votre métier

Comment assurer la *durabilité* et la *disponibilité* de vos données?

Besoin:

- Fiabilité
- Robustesse
- Performances
- Accompagnement du succès commercial
- Continuité et Innovation



Vos données sont votre métier

Comment assurer la *durabilité* et la *disponibilité* de vos données?

Besoin:

- Fiabilité
- Robustesse
- Performances
- Accompagnement du succès commercial
- Continuité et Innovation



1 Présentation

- Préambule
- Disponibilité, Durabilité
- Architectures et Répliquions

2 Architectures et Répliquions

- Augmentation du trafic
- Durabilité des données
- Disponibilité des données
- Disponibilité des services
- Sharding

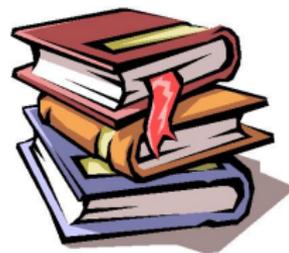
3 Conclusion

- Rétrospective et avenir de la répliquion PostgreSQL
- Questions

Glossaire

Thèmes abordés

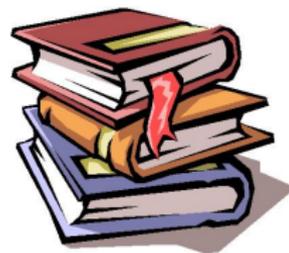
- Disponibilité
- Durabilité (ACID)
- Architectures
- Répliquions



Glossaire

Thèmes abordés

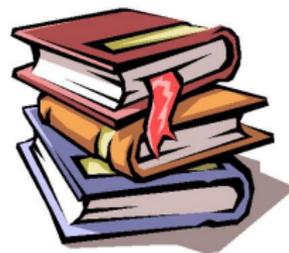
- Disponibilité
- Durabilité (ACID)
- Architectures
- Répliquions



Glossaire

Thèmes abordés

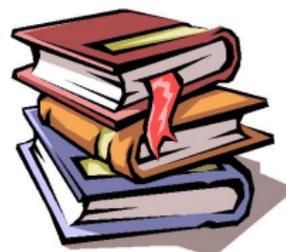
- Disponibilité
- Durabilité (ACID)
- Architectures
- Répliquions



Glossaire

Thèmes abordés

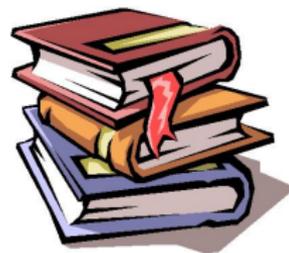
- Disponibilité *des services ou des données?*
- Durabilité (ACID)
- Architectures
- Réplifications



Glossaire

Thèmes abordés

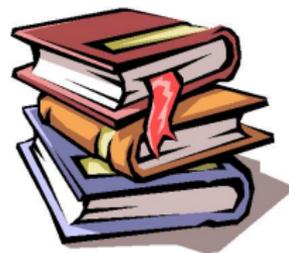
- Disponibilité *des services ou des données?*
- Durabilité (ACID)
- Architectures
- Réplifications



Glossaire

Thèmes abordés

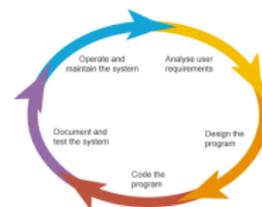
- Disponibilité *des services ou des données?*
- Durabilité (ACID)
- Architectures
- Répliquions



Une approche par le besoin

Les besoins évoluent, les solutions s'adaptent

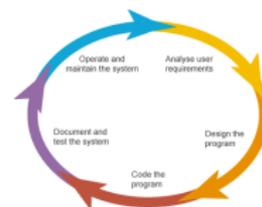
- Projet simple, besoins simples
- Le projet évolue et les besoins aussi
- Haute Disponibilité *des données*
- Haute Disponibilité *des services*
- Répartition de charge *en lecture*
- Répartition de charge *en écriture*



Une approche par le besoin

Les besoins évoluent, les solutions s'adaptent

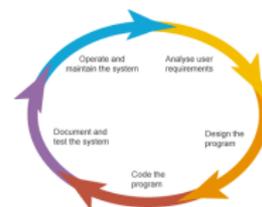
- Projet simple, besoins simples
- Le projet évolue et les besoins aussi
- Haute Disponibilité *des données*
- Haute Disponibilité *des services*
- Répartition de charge *en lecture*
- Répartition de charge *en écriture*



Une approche par le besoin

Les besoins évoluent, les solutions s'adaptent

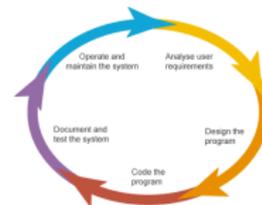
- Projet simple, besoins simples
- Le projet évolue et les besoins aussi
- Haute Disponibilité *des données*
- Haute Disponibilité *des services*
- Répartition de charge *en lecture*
- Répartition de charge *en écriture*



Il faut bien commencer quelque part

Cycle de vie d'un projet

Prenons comme exemple un projet relativement simple, une application web dont les besoins évoluent avec le succès grandissant.



1 Présentation

- Préambule
- Disponibilité, Durabilité
- Architectures et Répliquions

2 Architectures et Répliquions

- Augmentation du trafic
- Durabilité des données
- Disponibilité des données
- Disponibilité des services
- Sharding

3 Conclusion

- Rétrospective et avenir de la répliquion PostgreSQL
- Questions

Séparation des tâches

Disponibilité des services

- partie frontale *stateless*
- attention à `max_connections`
- pas de connections persistentes!
- `pgbouncer`



Séparation des tâches

Disponibilité des services

- partie frontale *stateless*
- attention à `max_connections`
- pas de connections persistentes!
- `pgbouncer`



Séparation des tâches

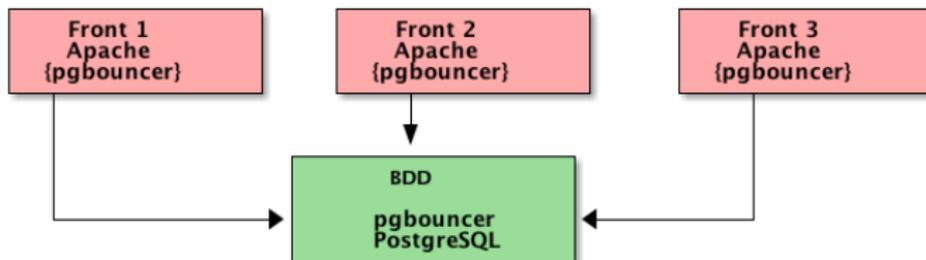
Disponibilité des services

- partie frontale *stateless*
- attention à `max_connections`
- pas de connections persistentes!
- `pgbouncer`



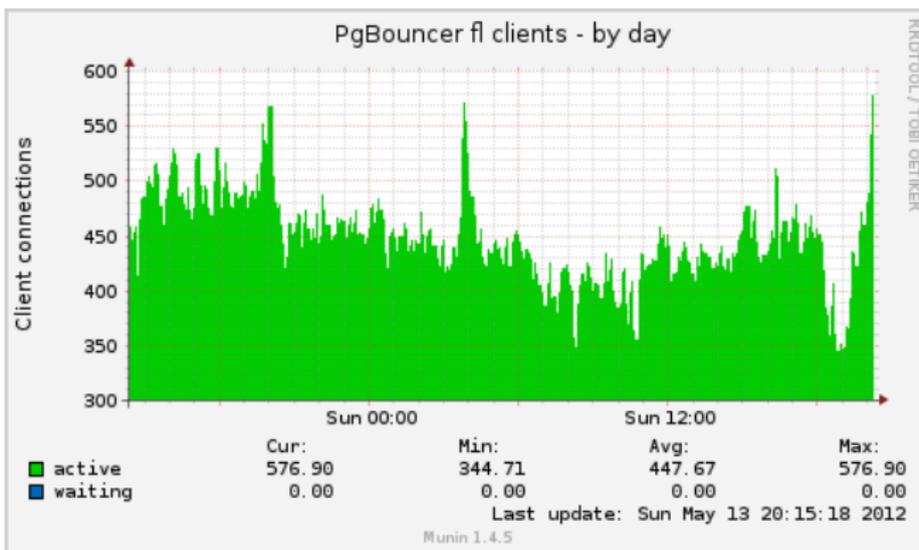
Séparation des tâches

Utilisation de serveurs dédiés et d'un concentrateur de connexions



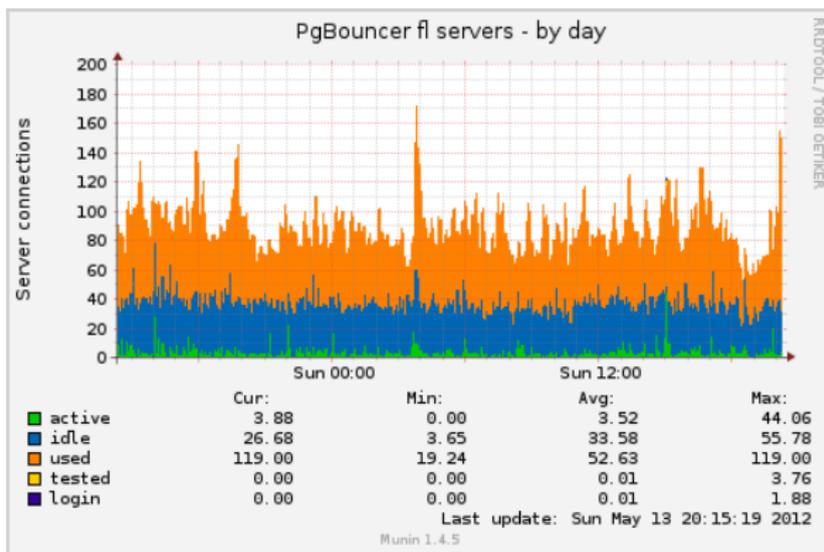
pgbouncer

pgbouncer réutilise des connexions client **et** serveur.



pgbouncer

pgbouncer réutilise des connexions client et serveur.



1 Présentation

- Préambule
- Disponibilité, Durabilité
- Architectures et Répliquions

2 Architectures et Répliquions

- Augmentation du trafic
- Durabilité des données
- Disponibilité des données
- Disponibilité des services
- Sharding

3 Conclusion

- Rétrospective et avenir de la répliquion PostgreSQL
- Questions

Le plan de sauvegardes

Le plan de sauvegardes est crucial

- `pg_dump -Fc nocturne`
- `pg_dumpall -globals-only`
- Rétention des données
- Nocturne, 7 jours
- Hebdomadaire, 7 semaines
- Mensuelle, 12 mois
- Annuelle, 30 ans



Le plan de sauvegardes

Le plan de sauvegardes est crucial

- `pg_dump -Fc nocturne`
- `pg_dumpall -globals-only`
- Rétention des données
- Nocturne, 7 jours
- Hebdomadaire, 7 semaines
- Mensuelle, 12 mois
- Annuelle, 30 ans



Le plan de sauvegardes

Le plan de sauvegardes est crucial

- `pg_dump -Fc nocturne`
- `pg_dumpall -globals-only`
- Rétention des données
- Nocturne, 7 jours
- Hebdomadaire, 7 semaines
- Mensuelle, 12 mois
- Annuelle, 30 ans



Plan de Reprise de l'Activité

`pg_dump`, `pg_restore`

- protège contre les *erreurs et omissions*
- temps de restauration conséquent
- Indispensable à la **durabilité** des données
- La **disponibilité** est-elle suffisante?



Plan de Reprise de l'Activité

`pg_dump`, `pg_restore`

- protège contre les *erreurs et omissions*
- temps de restauration conséquent
- Indispensable à la **durabilité** des données
- La **disponibilité** est-elle suffisante?



1 Présentation

- Préambule
- Disponibilité, Durabilité
- Architectures et Répliquions

2 Architectures et Répliquions

- Augmentation du trafic
- Durabilité des données
- Disponibilité des données
- Disponibilité des services
- Sharding

3 Conclusion

- Rétrospective et avenir de la répliquion PostgreSQL
- Questions

Plan de Reprise de l'Activité

Sauvegardes physiques et retour à un point dans le passé

- **Point In Time Recovery**, 8.1
- *warm standby*, 8.2
- Archivage et *crash recovery*
- `archive_command`
- `restore_command`
- `walmgr.py`, WAL-E



Plan de Reprise de l'Activité

Sauvegardes physiques et retour à un point dans le passé

- **Point In Time Recovery**, 8.1
- *warm standby*, 8.2
- Archivage et *crash recovery*
- `archive_command`
- `restore_command`
- `walmgr.py`, WAL-E



Plan de Reprise de l'Activité

Sauvegardes physiques et retour à un point dans le passé

- **Point In Time Recovery**, 8.1
- *warm standby*, 8.2
- Archivage et *crash recovery*
- `archive_command`
- `restore_command`
- `walmgr.py`, WAL-E



Plan de Reprise de l'Activité

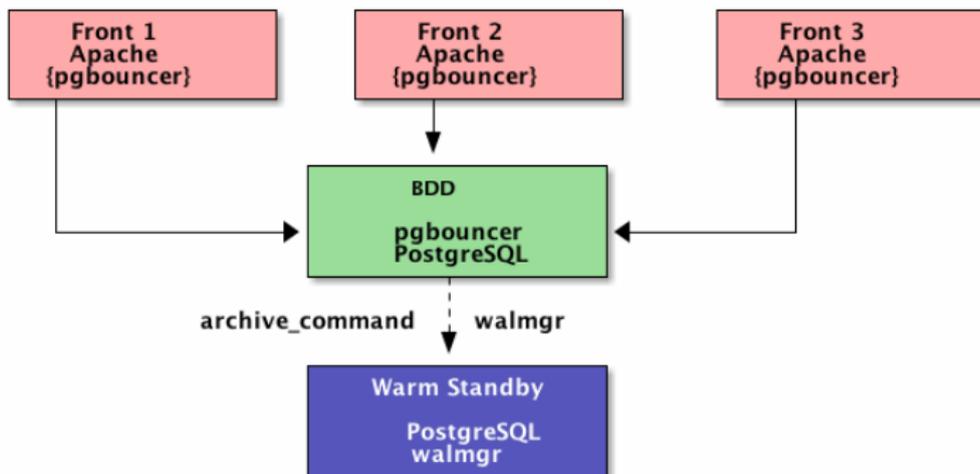
Sauvegardes physiques et retour à un point dans le passé

- **Point In Time Recovery**, 8.1
- *warm standby*, 8.2
- Archivage et *crash recovery*
- `archive_command`
- `restore_command`
- `walmgr.py`, WAL-E



Warm Standby

Mise en place du Warm Standby



1 Présentation

- Préambule
- Disponibilité, Durabilité
- Architectures et Répliquions

2 Architectures et Répliquions

- Augmentation du trafic
- Durabilité des données
- Disponibilité des données
- Disponibilité des services
- Sharding

3 Conclusion

- Rétrospective et avenir de la répliquion PostgreSQL
- Questions

Séparation des composants

Il est parfois possible de distinguer backoffice et production

- Réplication croisée
- Slony, Londiste, Bucardo
- Traitements spécifiques, *batches*
- Contexte *hors-ligne*
- Traitements *Transactionnels*
- Skytools fourni **PGQ**



Séparation des composants

Il est parfois possible de distinguer backoffice et production

- Réplication croisée
- Slony, **Londiste**, Bucardo
- Traitements spécifiques, *batches*
- Contexte *hors-ligne*
- Traitements *Transactionnels*
- Skytools fourni **PGQ**



Séparation des composants

Il est parfois possible de distinguer backoffice et production

- Réplication croisée
- Slony, **Londiste**, Bucardo
- Traitements spécifiques, *batches*
- Contexte *hors-ligne*
- Traitements *Transactionnels*
- Skytools fourni **PGQ**



Séparation des composants

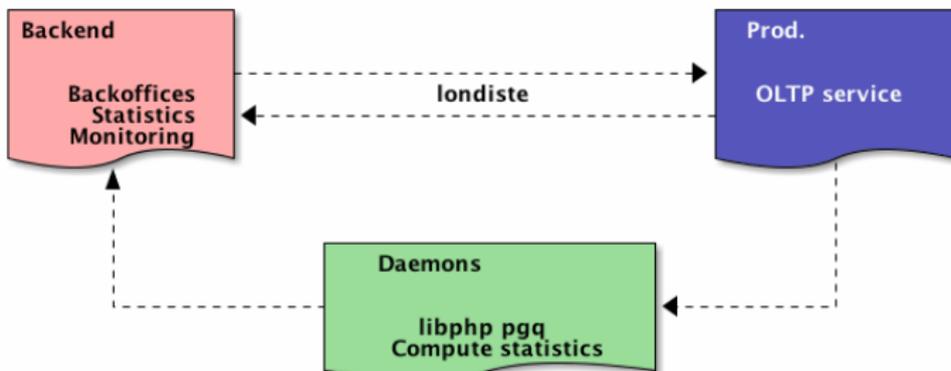
Il est parfois possible de distinguer backoffice et production

- Réplication croisée
- Slony, **Londiste**, Bucardo
- Traitements spécifiques, *batches*
- Contexte *hors-ligne*
- Traitements *Transactionnels*
- Skytools fourni **PGQ**



Séparation des composants

Mise en place de Londiste et PGQ



Queueing avec PGQ

PGQ pour les traitements *hors ligne*

- Écrit principalement en PLpgSQL
- API clientes en python
- et PHP
- en cours de portage vers Java
- Skytools3: Cooperative Worker
- Fiable, robuste, facile à superviser



Queueing avec PGQ

PGQ pour les traitements *hors ligne*

- Écrit principalement en PLpgSQL
- API clientes en python
- et PHP
- en cours de portage vers Java
- Skytools3: Cooperative Worker
- Fiable, robuste, facile à superviser



Queueing avec PGQ

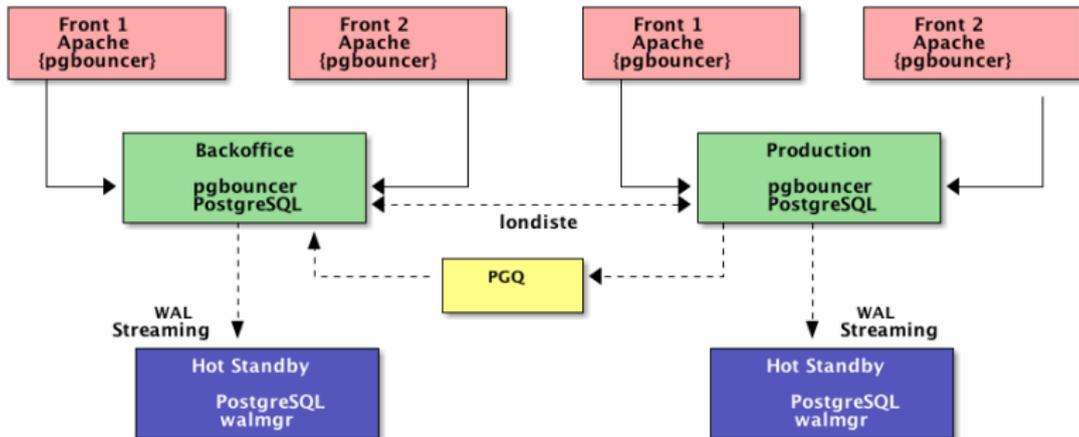
PGQ pour les traitements *hors ligne*

- Écrit principalement en PLpgSQL
- API clientes en python
- et PHP
- en cours de portage vers Java
- Skytools3: **Cooperative Worker**
- Fiable, robuste, facile à superviser



Plan de Continuité de l'Activité

Mise en place de Hot Standby



1 Présentation

- Préambule
- Disponibilité, Durabilité
- Architectures et Réplifications

2 Architectures et Réplifications

- Augmentation du trafic
- Durabilité des données
- Disponibilité des données
- Disponibilité des services
- Sharding

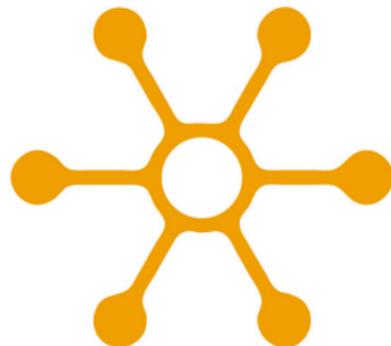
3 Conclusion

- Rétrospective et avenir de la réplication PostgreSQL
- Questions

Augmenter la capacité en écriture

PL/proxy

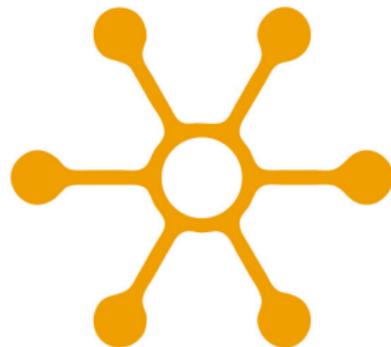
- *Scale-up* ou *Scale-out*?
- *Remote Procedure Call*
- *Sharding*
- Base de données répartie
- Transactions autonomes
- Procédures Stockées



Augmenter la capacité en écriture

PL/proxy

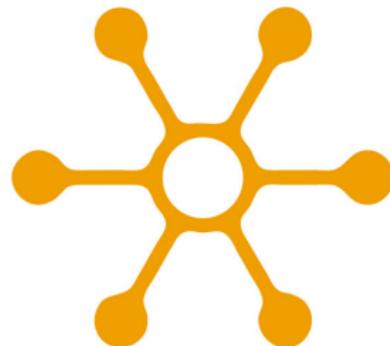
- *Scale-up* ou *Scale-out*?
- *Remote Procedure Call*
- *Sharding*
- Base de données répartie
- Transactions autonomes
- **Procédures Stockées**



Augmenter la capacité en écriture

PL/proxy

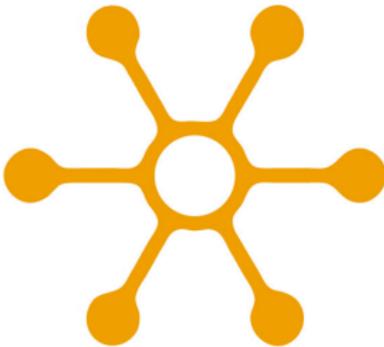
- *Scale-up* ou *Scale-out*?
- *Remote Procedure Call*
- *Sharding*
- Base de données répartie
- Transactions autonomes
- Procédures Stockées



Augmenter la capacité en écriture

PL/proxy

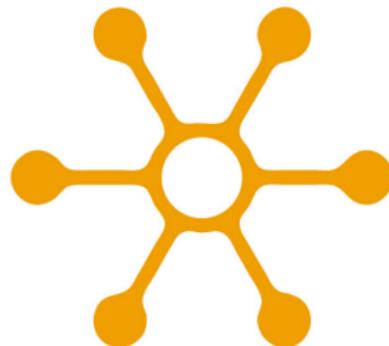
- *Scale-up* ou *Scale-out*?
- *Remote Procedure Call*
- *Sharding*
- Base de données répartie
- Transactions autonomes
- Procédures Stockées



Augmenter la capacité en écriture

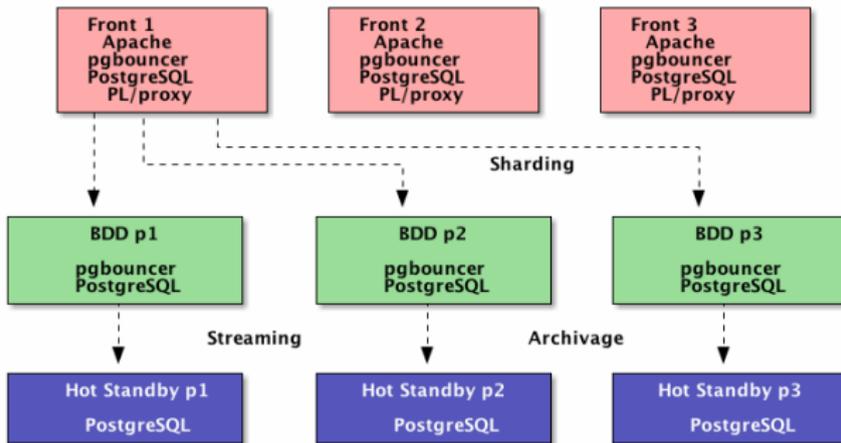
PL/proxy

- *Scale-up* ou *Scale-out*?
- *Remote Procedure Call*
- *Sharding*
- Base de données répartie
- Transactions autonomes
- **Procédures Stockées**



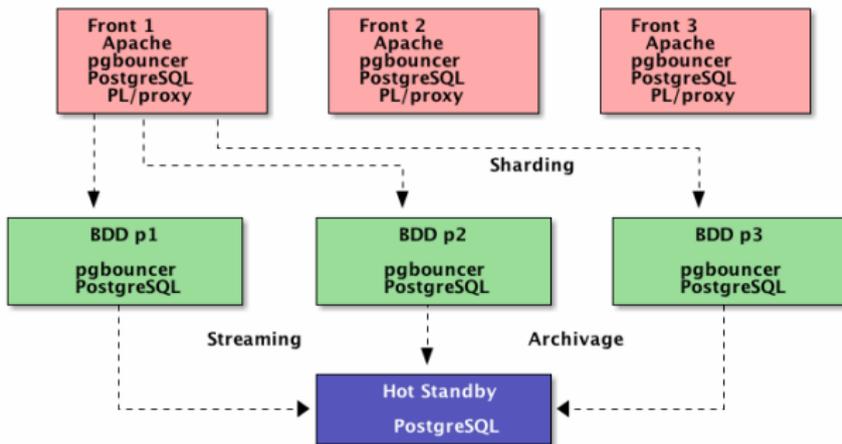
Augmenter la capacité en écriture

Mise en place de p1proxy



Augmenter la capacité en écriture

Mise en place de p1proxy



1 Présentation

- Préambule
- Disponibilité, Durabilité
- Architectures et Répliquions

2 Architectures et Répliquions

- Augmentation du trafic
- Durabilité des données
- Disponibilité des données
- Disponibilité des services
- Sharding

3 Conclusion

- Rétrospective et avenir de la répliquion PostgreSQL
- Questions

Haute Disponibilité Multi Sites

Rétrospective et avenir de la réplication dans PostgreSQL

- **8.1, PITR**
- 8.2, Warm Standby
- 8.3, pg_standby
- 9.0, Hot Standby
- 9.1, Replication Synchrone
- 9.2, Replication en Cascade
- 9.3, Bi-Directional Replication



Haute Disponibilité Multi Sites

Rétrospective et avenir de la réplication dans PostgreSQL

- 8.1, PITR
- 8.2, Warm Standby
- 8.3, `pg_standby`
- 9.0, Hot Standby
- 9.1, Replication Synchrone
- 9.2, Replication en Cascade
- 9.3, **Bi-Directional Replication**



Haute Disponibilité Multi Sites

Rétrospective et avenir de la réplication dans PostgreSQL

- 8.1, PITR
- 8.2, Warm Standby
- 8.3, pg_standby
- 9.0, Hot Standby
- 9.1, Replication Synchrone
- 9.2, Replication en Cascade
- 9.3, Bi-Directional Replication



Haute Disponibilité Multi Sites

Rétrospective et avenir de la réplication dans PostgreSQL

- 8.1, PITR
- 8.2, Warm Standby
- 8.3, pg_standby
- 9.0, Hot Standby
- 9.1, Replication Synchrone
- 9.2, Replication en Cascade
- 9.3, Bi-Directional Replication



Haute Disponibilité Multi Sites

Rétrospective et avenir de la réplication dans PostgreSQL

- 8.1, PITR
- 8.2, Warm Standby
- 8.3, `pg_standby`
- 9.0, Hot Standby
- 9.1, Replication Synchrone
- 9.2, Replication en Cascade
- 9.3, **Bi-Directional Replication**



Haute Disponibilité Multi Sites

Rétrospective et avenir de la répliation dans PostgreSQL

- 8.1, PITR
- 8.2, Warm Standby
- 8.3, pg_standby
- 9.0, Hot Standby
- 9.1, Replication Synchrone
- 9.2, Replication en Cascade
- 9.3, Bi-Directional Replication



Haute Disponibilité Multi Sites

Rétrospective et avenir de la réplication dans PostgreSQL

- 8.1, PITR
- 8.2, Warm Standby
- 8.3, pg_standby
- 9.0, Hot Standby
- 9.1, Replication Synchrone
- 9.2, Replication en Cascade
- 9.3, **Bi-Directional Replication**



Questions?

Retrouvez-nous sur le stand et dans les couloirs!

