

**anaska**

Alter Way GROUP

**FORUM PHP**  
Paris 2008



# Architectures haute disponibilité avec MySQL

Architectures haute disponibilité avec MySQL

Olivier DASINI - <http://dasini.net/blog>

Forum PHP 2008

# Votre conférencier

## **Olivier DASINI**

Consultant / Formateur certifié MySQL

Responsable du pôle bases de données d'ANASKA Alter Way

**[olivier.dasini@anaska.com](mailto:olivier.dasini@anaska.com)**

**<http://dasini.net/blog/>**

## Au menu...

- Introduction à la haute disponibilité
- MySQL replication
- MySQL cLuster
- Shared disk clustering
- Questions / Réponses

# Introduction à la haute disponibilité

- **Définition**

- Disponibilité =  $\text{uptime}/(\text{uptime}+\text{downtime})$ 
  - 90% => 36,5 jours/an || 99,9999% => 31 sec/an

- **Quel but**

- Minimiser le downtime
  - Redondance logicielle et/ou matérielle

- **Les termes**

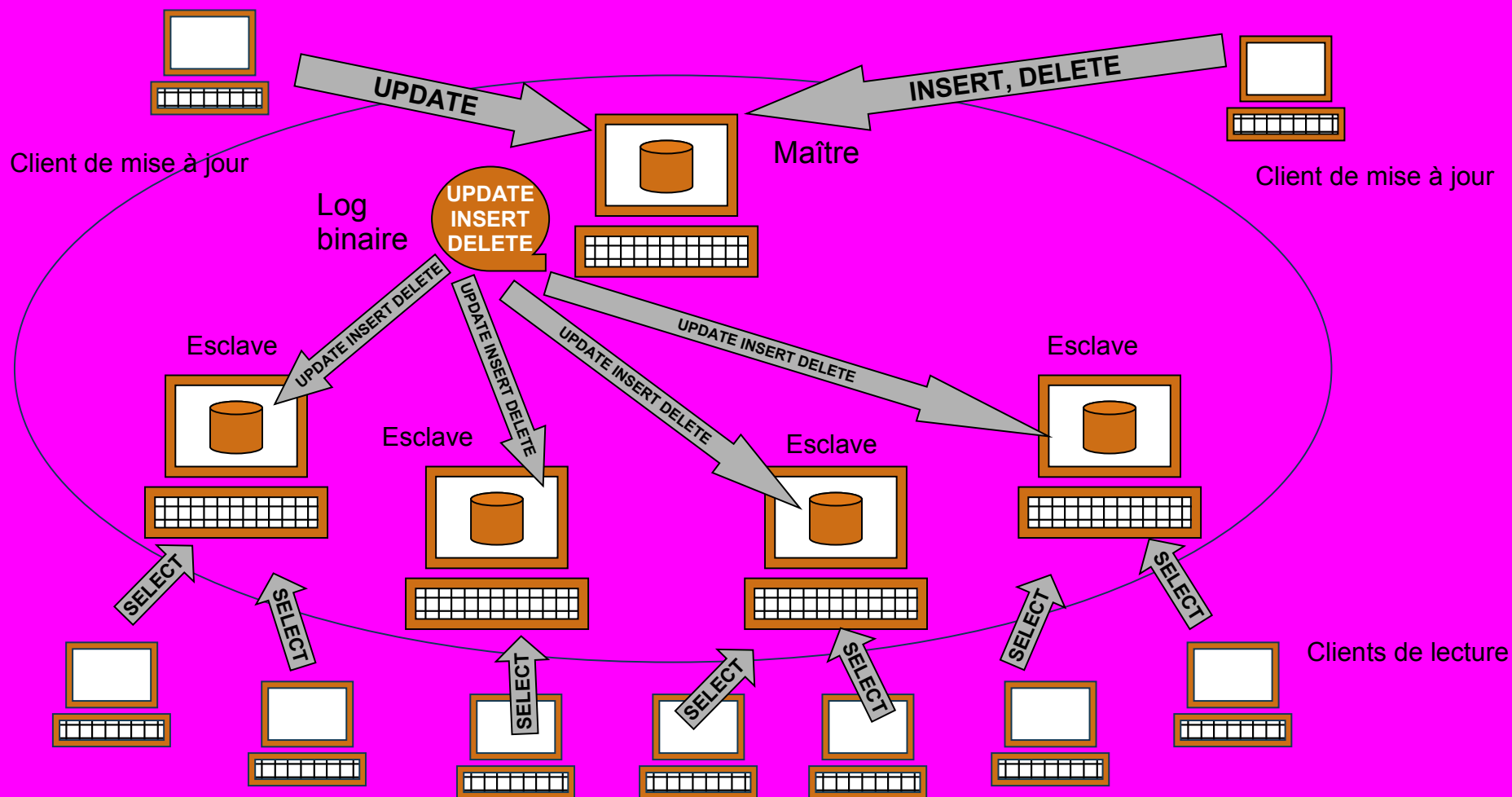
- Synchrone / Asynchrone
- Réplication / Cluster
- Shared disk / Shared nothing

# MySQL Replication

# MySQL Replication

# MySQL Replication

## Vue d'ensemble de la réplication

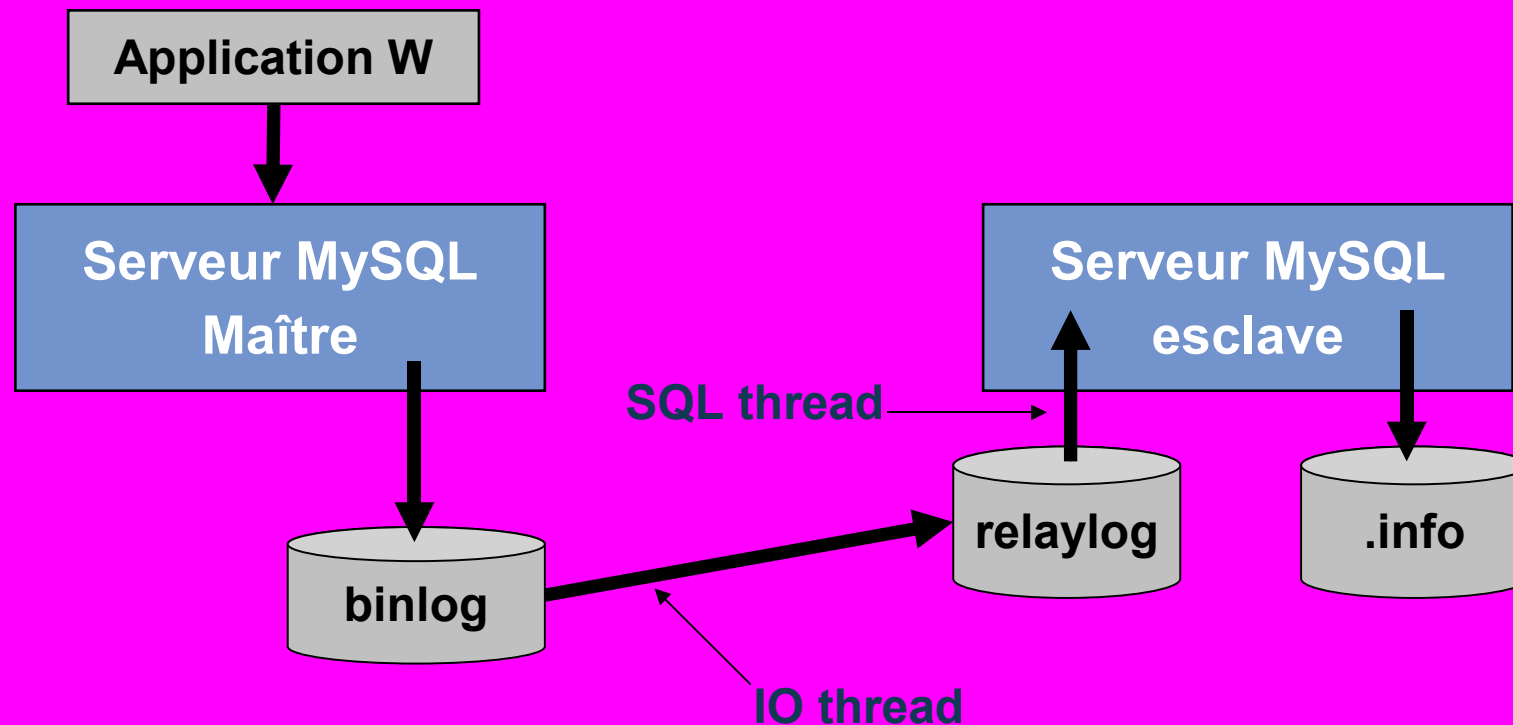


# MySQL Replication

- Log binaire
- Row based / Statement based
- SET binlog\_format = row | statement | mixed

# MySQL Replication

## Détail de la réplication



# MySQL Replication

- IO thread
  - Récupère les informations du log binaire du maître et les copie dans le relay log
- SQL thread
  - Joue les requêtes du relay log sur l'esclave

# MySQL Replication

## Configuration du maître

- Activer log binaire
- Définir server-id
- Utilisateur avec droit REPLICATION SLAVE
- Faire une sauvegarde synchronisée avec les logs binaires
  - `mysqldump --master-data`

# MySQL Replication

## Configuration de l'esclave

- Définir server-id
- Restaurer la sauvegarde
- CHANGE MASTER TO  
MASTER\_HOST='adresse\_maitre',  
MASTER\_USER='utilisateur\_replication',  
MASTER\_PORT= port,  
MASTER\_LOG\_FILE='log\_binaire\_maitre',  
MASTER\_LOG\_POS=position\_ds\_log\_binaire\_maitre;
- SLAVE START;

# MySQL Replication - Avantages

- Simplicité
- Natif dans MySQL
- Stable
- Possibilité d'utiliser SSL
- Possibilité de mélanger différents moteurs

# MySQL Replication - Inconvénients

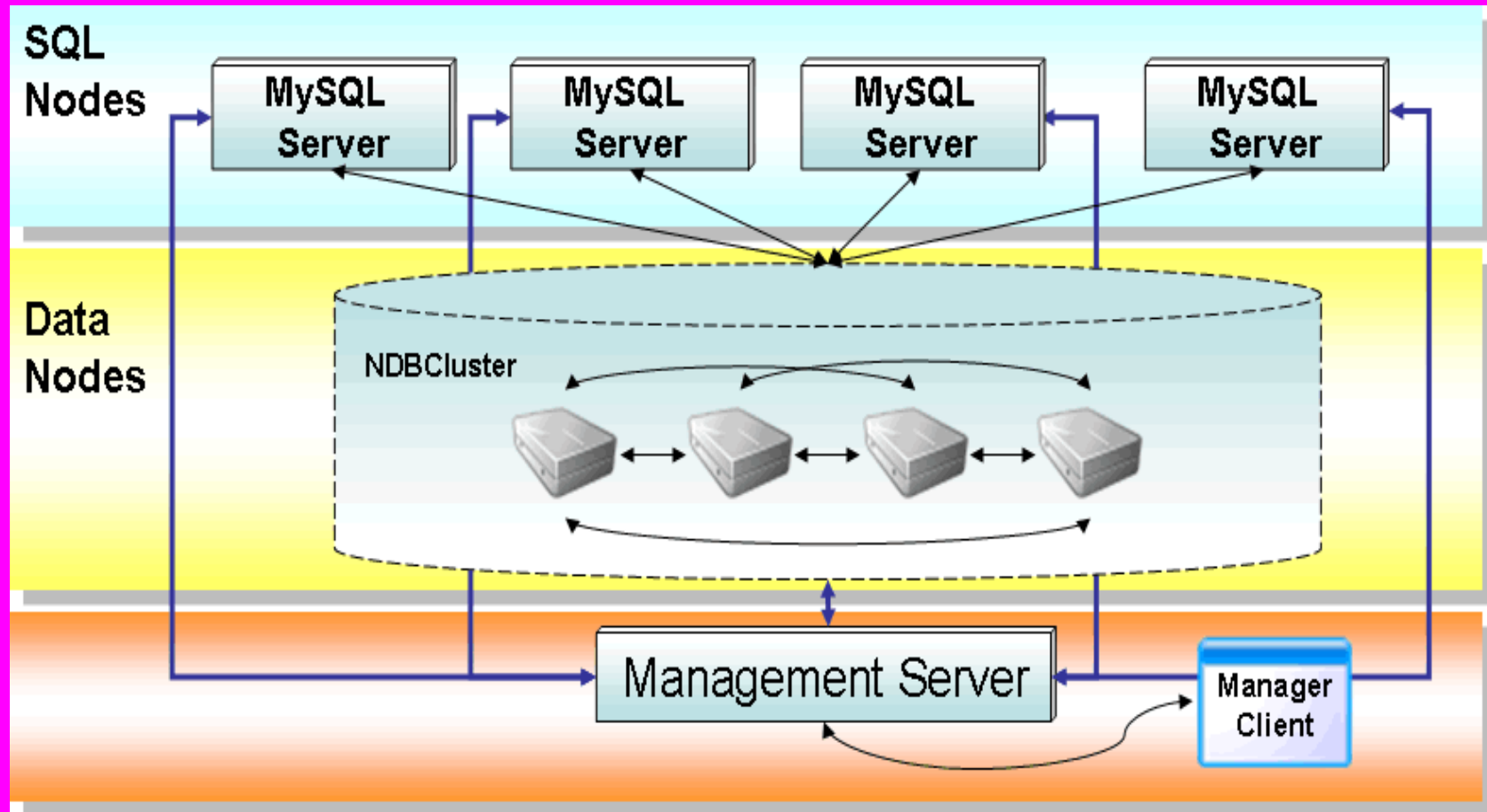
- Asynchrone
- Pas de basculement automatique
- Pas « scalable » en écriture
- Le maître est le goulot d'étranglement
- Le maître est SPOF
- Load balancing non géré par MySQL

# MySQL Replication - Avancée

- Réplication circulaire
- Changement de structure online
- Mise à jour online
- Mélanger différents moteurs de stockage
- Différents réglages serveur
  - delay\_key\_write (MyISAM)
  - innodb\_flush\_log\_at\_trx\_commit (InnoDB)
  - sync\_binlog (log binaire)
- Différentes structures d'index

# MySQL Cluster

# MySQL Cluster



# MySQL Cluster – Les nœuds

- Noeuds sql
  - Serveur MySQL(en général) avec ndbcluster activé
  - Autres: ndb\_restore...
  - Reçoivent les requêtes et renvoient les résultats à l'application
- Noeuds de données
  - Processus ndbd
  - Stockage des données et index
- Noeuds de gestion
  - Processus ndb\_mgmd
  - Configuration (config.ini)
  - Gestion (arrêt, redémarrage des nœuds, sauvegarde...)

# MySQL Cluster

- Partitionnement
- Réplication synchrone
- Transactions
- Shared nothing
- No SPOF

# MySQL Cluster – Configuration

- MySQL Cluster (avec le moteur ndb cluster SE ou CGE)
- Créer le fichier de configuration (config.ini)
- Démarrer le noeud de gestion
- Démarrer les noeuds de données
- Démarrer les noeuds sql

# MySQL Cluster – config.ini

**[ndbd default]**

NoOfReplicas= 2

DataDir= /var/lib/mysql-cluster

**[ndb\_mgmd]**

Hostname= adresse\_serveur\_gestion

DataDir= /var/lib/mysql-cluster

**[ndbd]**

HostName= adresse\_noeud\_donnée\_1

**[ndbd]**

HostName= adresse\_noeud\_donnée\_2

**[mysqld]**

HostName= adresse\_noeud\_sql\_1

**[mysqld]**

# MySQL Cluster – Avantages

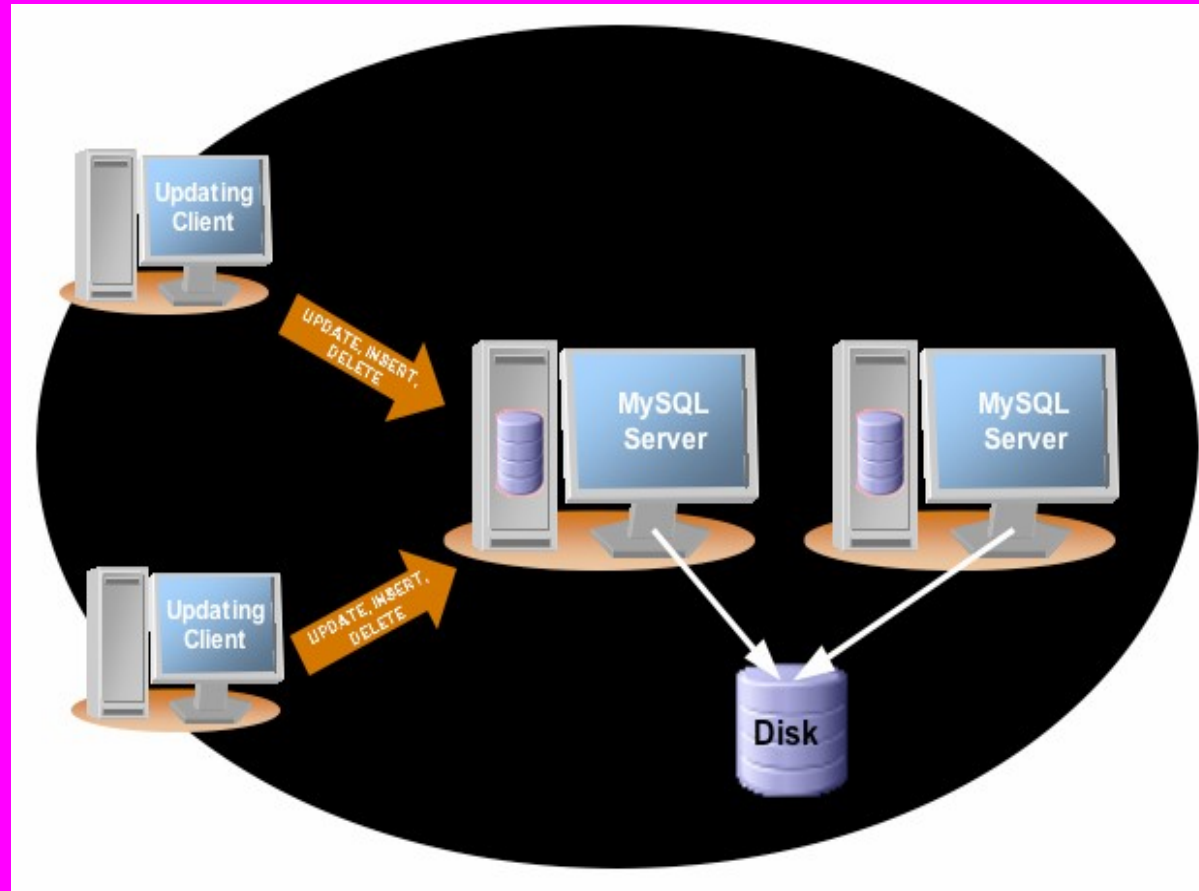
- Haute disponibilité
  - Basculement automatique
  - Sauvegarde à chaud
  - Mise à jour à chaud
  - Ajout de nœud sql à chaud
- Faible coût
- « Scalable » lecture & écriture
- Shared Nothing
- No SPOF

# MySQL Cluster – Inconvénients

- Requêtes complexes
  - Performances moindres avec plusieurs sous-requêtes, jointures...
- Systèmes d'exploitation
  - Pas de support MS Windows
- MySQL 5.0.x
  - Données & index nécessairement en RAM
- MySQL 5.1.x
  - Index nécessairement en RAM
- Spécificités du moteur NDB
  - Pas de fulltext, intégrité référentielle...

# Shared Disk Clustering

# Shared Disk Clustering



# Shared Disk Clustering

- Plusieurs machines partagent un même espace de stockage
  - SAN (storage area network)
- Connexion haut débit vers le système de stockage
  - Fibre channel, iSCSI, ...
- Un système est nécessaire pour vérifier l'état des noeuds et gérer le basculement
  - Heartbeat, Linux HA,...

# Shared Disk Clustering Active-Passive

- Un nœud est accessible (actif)
- L'autre en attente (passif)
- Le basculement est géré par un programme externe
- Vérification des tables avant démarrage du passif
  - MyISAM: `myisamchk, --myisam-recover`

# Shared Disk Clustering Active-Active

- Tous les nœuds sont actif en même temps
- Quelques limitations avec MySQL
  - Seulement les tables MyISAM
  - Problèmes liés aux caches(requêtes, index,...)
  - ...

# Shared Disk Clustering – Avantages

- Haute disponibilité
- Assurance que les données sont les mêmes quelque soit le serveur
- Supporte de grands volumes de données

# Shared Disk Clustering – Inconvénients

- Coût
- SPOF
  - Le système de stockage (en théorie)

# Questions ?

