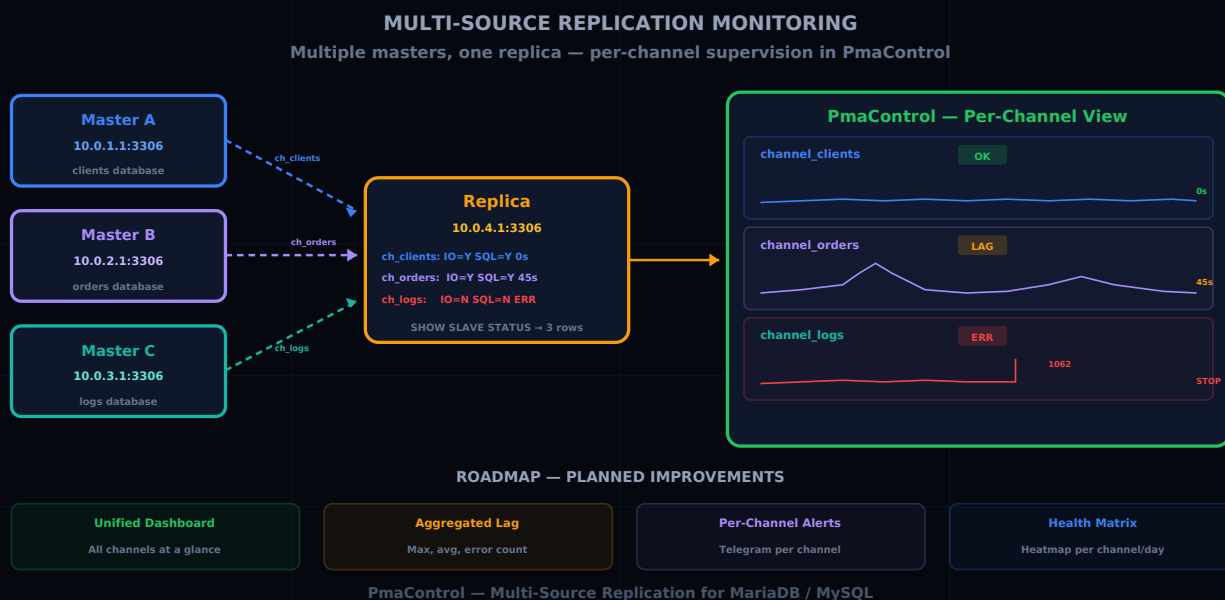


# Superviser la réplication multi-source avec PmaControl

Aurélien LEQUOY · 13 avril 2026

MARIADB MYSQL MULTI-SOURCE REPLICATION MONITORING PMACONTROL



## Multi-source : plusieurs masters, un seul réplica

La réplication multi-source permet à un unique serveur réplica de recevoir des données de **plusieurs masters simultanément**. Chaque connexion master est un **canal** (channel) distinct, avec son propre thread IO, son propre thread SQL, et son propre état de réplication.

C'est un pattern puissant pour :

- **Consolidation** : agréger les données de plusieurs bases métier sur un réplica analytique
- **Migration** : recevoir des données de l'ancien et du nouveau système pendant la transition
- **Data warehouse** : alimenter un entrepôt depuis plusieurs sources MariaDB / MySQL

Dans notre article précédent sur la réplication multi-source MySQL 8.4, nous avons couvert la configuration. Ici, nous nous concentrons sur la supervision : comment PmaControl détecte, affiche et surveille ces canaux multiples.

## Configuration rapide (rappel)

### MariaDB

MariaDB supporte nativement le multi-source depuis la version 10.0 :

```
-- Canal 1 : base clients
CHANGE MASTER 'channel_clients' TO
  MASTER_HOST = '10.0.1.1',
  MASTER_USER = 'repl',
  MASTER_PASSWORD = 'secret',
  MASTER_USE_GTID = slave_pos;
START SLAVE 'channel_clients';

-- Canal 2 : base commandes
CHANGE MASTER 'channel_orders' TO
  MASTER_HOST = '10.0.2.1',
  MASTER_USER = 'repl',
  MASTER_PASSWORD = 'secret',
  MASTER_USE_GTID = slave_pos;
START SLAVE 'channel_orders';
```

### MySQL 8.0+

MySQL utilise la clause `FOR CHANNEL` :

```
-- Canal 1
CHANGE REPLICATION SOURCE TO
  SOURCE_HOST = '10.0.1.1',
  SOURCE_USER = 'repl',
  SOURCE_PASSWORD = 'secret',
  SOURCE_AUTO_POSITION = 1
  FOR CHANNEL 'channel_clients';
START REPLICA FOR CHANNEL 'channel_clients';

-- Canal 2
CHANGE REPLICATION SOURCE TO
  SOURCE_HOST = '10.0.2.1',
  SOURCE_USER = 'repl',
  SOURCE_PASSWORD = 'secret',
```

```
SOURCE_AUTO_POSITION = 1
FOR CHANNEL 'channel_orders';
START REPLICA FOR CHANNEL 'channel_orders';
```

## Comment PmaControl détecte le multi-source

---

### La requête de détection

Quand l'Aspirateur collecte les données de réplication, il exécute :

```
SHOW SLAVE STATUS;
-- ou en MySQL 8.0+ :
SHOW REPLICA STATUS;
```

En réplication classique (un seul master), cette requête retourne **une seule ligne**. En multi-source, elle retourne **une ligne par canal**.

PmaControl détecte automatiquement le multi-source quand `SHOW SLAVE STATUS` retourne plus d'une ligne. Aucune configuration spéciale n'est nécessaire côté PmaControl.

### Stockage interne

Chaque canal est stocké comme une entrée de réplication indépendante dans les tables time series. Le nom du canal est conservé et utilisé comme discriminant :

```
ts_value_general_json:
  server_id = 42
  variable  = slave_status
  channel   = 'channel_clients'
  value     = { "Slave_IO_Running": "Yes", "Seconds_Behind_Master": 3, ... }

server_id = 42
variable  = slave_status
channel   = 'channel_orders'
value     = { "Slave_IO_Running": "Yes", "Seconds_Behind_Master": 0, ... }
```

## La page slave en multi-source

---

Quand PmaControl détecte plusieurs canaux sur un serveur, la page slave affiche **chaque canal indépendamment**. Concrètement :

## Un bloc par canal

Chaque canal a son propre bloc avec :

- **Nom du canal** affiché en en-tête (ex: `channel_clients` )
- **État IO/SQL** : les indicateurs `Slave_IO_Running` et `Slave_SQL_Running` propres au canal
- **Lag** : `Seconds_Behind_Master` spécifique au canal
- **GTID** : l'état GTID propre au canal
- **Dernière erreur** : `Last_SQL_Error` du canal

## Un graphique de lag par canal

Chaque canal a son propre graphique Chart.js avec l'historique de lag sur 5 jours. Les graphiques sont empilés verticalement sur la page.

Cela permet de comparer visuellement : si `channel_clients` a un lag stable à 0 secondes mais que `channel_orders` montre des pics récurrents, le problème est clairement localisé sur le master des commandes.

## Health strip par canal

Chaque canal a sa propre bande de santé. Un canal en vert et l'autre en rouge donne immédiatement l'information : le problème est sur un seul canal, pas sur le réplica lui-même.

## Actions par canal

Les actions (START, STOP, SKIP) sont disponibles **par canal** :

```
-- Arrêter un seul canal
STOP SLAVE 'channel_orders';

-- Sauter une erreur sur un canal
SET GLOBAL sql_slave_skip_counter = 1;
START SLAVE 'channel_orders';

-- MariaDB : stopper un canal spécifique
STOP SLAVE 'channel_clients';
```

## REPLICATE\_REWRITE\_DB par canal

En multi-source, il est courant d'utiliser `REPLICATE_REWRITE_DB` pour remapper les noms de bases :

```
CHANGE REPLICATION SOURCE TO
  SOURCE_HOST = '10.0.1.1',
  ...
  FOR CHANNEL 'channel_clients';

-- Remapper 'clients' vers 'dw_clients' sur le réplica
CHANGE REPLICATION FILTER
  REPLICATE_REWRITE_DB = ((clients, dw_clients))
  FOR CHANNEL 'channel_clients';
```

PmaControl détecte et affiche ces filtres dans les détails du canal. C'est important pour le debugging : si une table n'apparaît pas sur le réplica, vérifier les filtres de réécriture est le premier réflexe.

## Limitations actuelles

PmaControl gère bien le multi-source au niveau de chaque canal individuel. Cependant, certaines fonctionnalités manquent pour une supervision vraiment unifiée.

### Pas de vue "tous les canaux sains"

Actuellement, il n'existe pas de page consolidée qui montre l'état de santé de tous les canaux d'un réplica en un seul coup d'oeil. Il faut scroller la page slave pour vérifier chaque canal individuellement.

Ce qui manque : un tableau récapitulatif en haut de page avec un indicateur vert/rouge par canal, comme :

```
channel_clients  □ I0: Yes  SQL: Yes  Lag: 0s
channel_orders   ▲□ I0: Yes  SQL: Yes  Lag: 45s
channel_logs     □ I0: No   SQL: No   Error: 1062
```

### Pas de lag agrégé

PmaControl surveille le lag de chaque canal séparément, mais ne calcule pas de métrique agrégée. Par exemple :

- Le lag maximum parmi tous les canaux
- Le lag moyen
- Le nombre de canaux en erreur sur le total

Ces métriques agrégées seraient utiles pour les alertes : "au moins un canal a plus de 60 secondes de lag" est plus pertinent que des alertes canal par canal quand on a 10 canaux.

## Pas de graphe de dépendances entre canaux

En multi-source, les canaux sont indépendants. Mais dans certaines architectures, il y a des dépendances logiques : le canal A doit être appliqué avant le canal B (par exemple, les tables de référence avant les tables transactionnelles).

PmaControl ne modélise pas ces dépendances. Le DBA doit gérer manuellement l'ordre d'application si nécessaire.

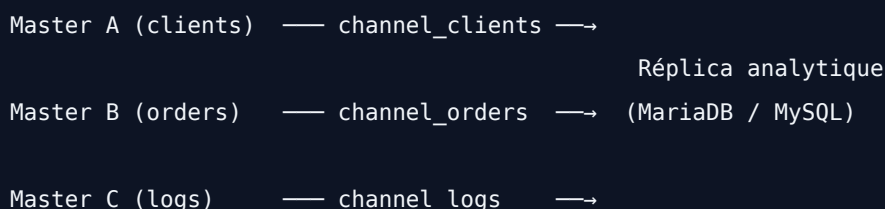
## Pas d'alerting par canal

Les alertes actuelles sont au niveau du serveur, pas du canal. Si le canal `channel_clients` s'arrête mais que `channel_orders` fonctionne, l'alerte indique simplement "réplication en erreur sur le serveur X" sans préciser le canal.

## Cas pratiques

### Consolidation analytique

Architecture typique :



Le réplica consolide les trois bases. PmaControl surveille les trois canaux. Si le master C (logs) est lent, le lag du canal `channel_logs` augmente sans impacter les deux autres.

Dans PmaControl, vous verrez trois blocs sur la page slave du réplica, avec trois graphiques de lag indépendants. Le canal `channel_logs` aura sa propre bande de santé en ambre ou rouge, tandis que les deux autres restent en vert.

## Migration par canal

Pendant une migration, vous pouvez avoir temporairement :

```
Ancien master — channel_legacy —>
                                     Nouveau réplica
Nouveau master — channel_new —>
```

Le canal `channel_legacy` sera supprimé une fois la migration terminée. Pendant la transition, PmaControl surveille les deux canaux et vous permet de vérifier que le nouveau canal a bien rattrapé avant de couper l'ancien.

## Filtrage par base

Avec `REPLICATE_DO_DB` ou `REPLICATE_REWRITE_DB`, chaque canal peut être filtré pour ne répliquer que certaines bases :

```
CHANGE REPLICATION FILTER
  REPLICATE_DO_DB = (clients, clients_archive)
  FOR CHANNEL 'channel_clients';

CHANGE REPLICATION FILTER
  REPLICATE_DO_DB = (orders, order_items)
  FOR CHANNEL 'channel_orders';
```

PmaControl affiche les filtres actifs pour chaque canal, ce qui facilite le diagnostic quand une table attendue n'apparaît pas sur le réplica.

## Roadmap

---

Les améliorations prévues pour le support multi-source dans PmaControl :

### Dashboard multi-source unifié

Un tableau de bord dédié au multi-source avec :

- Vue matricielle : un réplica en ligne, ses canaux en colonnes
- Indicateur de santé global (vert si tous les canaux OK, rouge si au moins un KO)
- Lag agrégé (max, moyenne, nombre d'erreurs)

## Matrice de santé des canaux

Une visualisation compacte type "heatmap" :

	Lun	Mar	Mer	Jeu	Ven	Sam	Dim
ch_clients	🟢	🟢	🟢	🟢	🟢	🟢	🟢
ch_orders	🟢	🟢	🟢	🟢	🟢	🟢	🟢
ch_logs	🟢	🟢	🟢	🟢	🟢	🟢	🟢

## Alerting par canal

Alertes Telegram spécifiques par canal :

```

△ Replication Warning
Server: replica-analytics (10.0.3.1:3306)
Channel: channel_orders
Lag: 120s (threshold: 60s)
Status: IO=Yes, SQL=Yes

```

## Détection automatique des dépendances

Analyse des clés étrangères cross-canal pour détecter les dépendances potentielles et alerter si un canal dépendant prend du retard.

## Bonnes pratiques multi-source

### 1. Nommer les canaux clairement

Utilisez des noms descriptifs : `channel_clients` , `channel_orders` , pas `ch1` , `ch2` . PmaControl affiche les noms tels quels — des noms clairs facilitent le diagnostic.

### 2. Un canal = une base (ou un groupe logique)

Évitez de mélanger des bases sans rapport dans un même canal. Si un canal s'arrête sur une erreur, tout ce qu'il contient est bloqué.

### 3. Monitorer le relay log space

Avec plusieurs canaux, l'espace disque du relay log peut exploser. Chaque canal a son propre relay log. Surveillez `Relay_Log_Space` par canal dans PmaControl.

### 4. Tester les arrêts de canal individuels

Assurez-vous que l'arrêt d'un canal n'impacte pas les autres. PmaControl vous permet de STOP/START chaque canal indépendamment — utilisez cette capacité pour valider l'isolation.

### 5. Documenter l'architecture

Le multi-source ajoute de la complexité. Documentez quel canal porte quelle base, depuis quel master, avec quels filtres. PmaControl affiche ces informations, mais une documentation externe reste utile pour l'onboarding.

## Conclusion

---

La réplication multi-source est un outil puissant pour la consolidation de données MariaDB / MySQL. PmaControl détecte automatiquement les canaux multiples et les surveille individuellement avec des graphiques de lag, des bandes de santé et des actions correctives par canal.

Les limitations actuelles — absence de vue unifiée, pas d>alerting par canal, pas de lag agrégé — sont identifiées et font partie de la roadmap. En attendant, la supervision canal par canal couvre les besoins essentiels : savoir quel canal a du lag, quel canal est en erreur, et pouvoir agir dessus individuellement.

Pour les DBA qui gèrent des architectures multi-source en production, PmaControl reste l'outil le plus adapté — même imparfait — car aucun concurrent n'offre cette visibilité canal par canal out of the box.