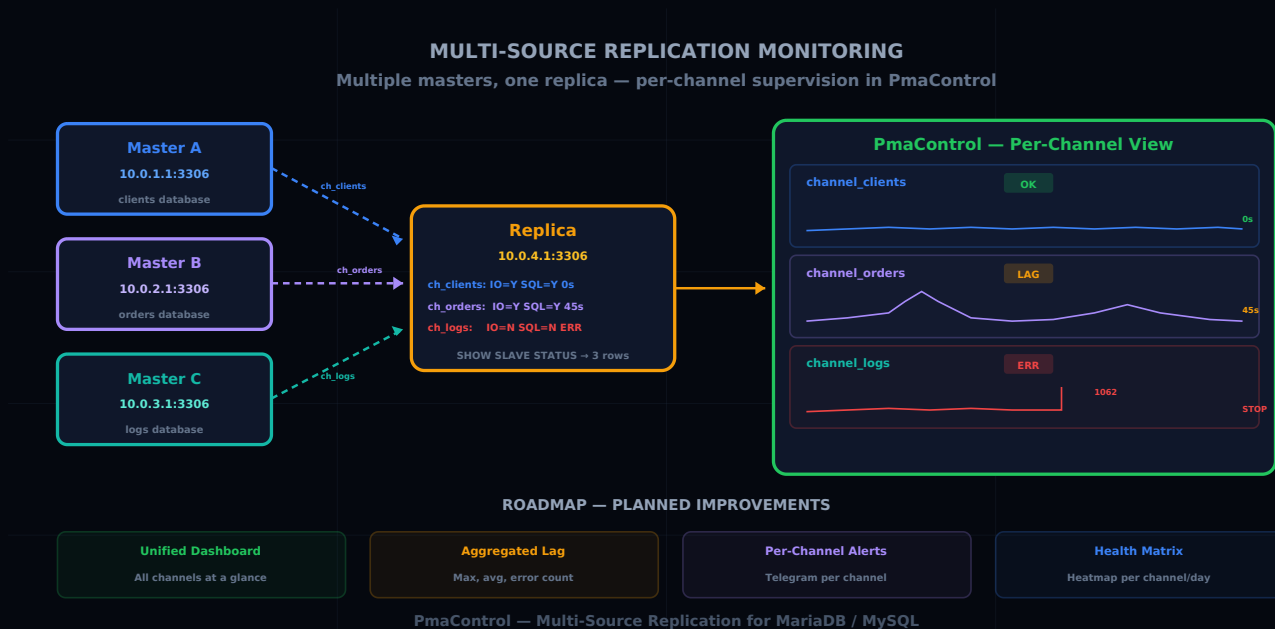


Nadzór replikacji multi-source z PmaControl

Aurélien LEQUOY · April 13, 2026

MARIADB MYSQL MULTI-SOURCE REPLICATION MONITORING PMACONTROL



Multi-source: wielu masterów, jedna replika

Replikacja multi-source pozwala pojedynczemu serwerowi repliki na otrzymywanie danych od **wielu masterów jednocześnie**. Każde połączenie z masterem to odrębny **kanał** (channel) z własnym wątkiem IO, własnym wątkiem SQL i własnym stanem replikacji.

To potężny wzorzec dla:

- **Konsolidacji**: agregowanie danych z wielu baz biznesowych na replice analitycznej
- **Migracji**: otrzymywanie danych ze starego i nowego systemu podczas przejścia
- **Data warehouse**: zasilanie hurtowni z wielu źródeł MariaDB / MySQL

W naszym poprzednim artykule o replikacji multi-source MySQL 8.4 opisywaliśmy konfigurację. Tutaj skupiamy się na nadzorze: jak PmaControl wykrywa, wyświetla i monitoruje te wiele kanałów.

Szybka konfiguracja (przypomnienie)

MariaDB

MariaDB natywnie wspiera multi-source od wersji 10.0:

```
-- Kanał 1: baza klientów
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';

-- Kanał 2: baza zamówień
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 używa klauzuli `FOR CHANNEL` :

```
-- Kanał 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 REPLICATION FOR CHANNEL 'channel_clients';

-- Kanał 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 REPLICATION FOR CHANNEL 'channel_orders';
```

Jak PmaControl wykrywa multi-source

Zapytanie wykrywające

Gdy Aspirateur zbiera dane replikacji, wykonuje:

```
SHOW SLAVE STATUS;  
-- lub w MySQL 8.0+:  
SHOW REPLICA STATUS;
```

W klasycznej replikacji (jeden master) to zapytanie zwraca **jeden wiersz**. W multi-source zwraca **jeden wiersz na kanał**.

PmaControl automatycznie wykrywa multi-source, gdy `SHOW SLAVE STATUS` zwraca więcej niż jeden wiersz. Żadna specjalna konfiguracja po stronie PmaControl nie jest wymagana.

Przechowywanie wewnętrzne

Każdy kanał jest przechowywany jako niezależny wpis replikacji w tabelach time series. Nazwa kanału jest zachowywana i używana jako wyróżnik:

```
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, ... }
```

Strona slave w multi-source

Gdy PmaControl wykrywa wiele kanałów na serwerze, strona slave wyświetla **każdy kanał niezależnie**. Konkretnie:

Blok na kanał

Każdy kanał ma własny blok z:

- **Nazwą kanału** wyświetlaną w nagłówku (np. `channel_clients`)
- **Stanem IO/SQL**: wskaźniki `Slave_IO_Running` i `Slave_SQL_Running` właściwe dla kanału
- **Opóźnieniem**: `Seconds_Behind_Master` specyficznym dla kanału
- **GTID**: stanem GTID właściwym dla kanału
- **Ostatnim błędem**: `Last_SQL_Error` kanału

Wykres opóźnień na kanał

Każdy kanał ma własny wykres Chart.js z historią opóźnień za 5 dni. Wykresy są ułożone pionowo na stronie.

Pozwala to na wizualne porównanie: jeśli `channel_clients` ma stabilne opóźnienie 0 sekund, ale `channel_orders` pokazuje powtarzające się skoki, problem jest wyraźnie zlokalizowany na masterze zamówień.

Health strip na kanał

Każdy kanał ma własny pasek zdrowia. Jeden kanał na zielono, a drugi na czerwono, natychmiast daje informację: problem dotyczy jednego kanału, nie samej repliki.

Akcje per kanał

Akcje (START, STOP, SKIP) są dostępne **per kanał**:

```
-- Zatrzymaj jeden kanał
STOP SLAVE 'channel_orders';

-- Pomiń błąd na kanale
SET GLOBAL sql_slave_skip_counter = 1;
START SLAVE 'channel_orders';

-- MariaDB: zatrzymaj konkretny kanał
STOP SLAVE 'channel_clients';
```

REPLICATE_REWRITE_DB per kanał

W multi-source często stosuje się `REPLICATE_REWRITE_DB` do przemapowania nazw baz danych:

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

-- Przemapuj 'clients' na 'dw_clients' na replice
CHANGE REPLICATION FILTER
REPLICATE_REWRITE_DB = ((clients, dw_clients))
FOR CHANNEL 'channel_clients';
```

PmaControl wykrywa i wyświetla te filtry w szczegółach kanału. Jest to ważne dla debugowania: jeśli tabela nie pojawia się na replice, sprawdzenie filtrów przepisywania jest pierwszym odruchem.

Obecne ograniczenia

PmaControl dobrze radzi sobie z multi-source na poziomie każdego indywidualnego kanału. Jednak pewne funkcjonalności brakują do naprawę zunifikowanego nadzoru.

Brak widoku "wszystkie kanały zdrowe"

Obecnie nie istnieje skonsolidowana strona pokazująca stan zdrowia wszystkich kanałów repliki jednym rzutem oka. Trzeba przewijać stronę slave, aby sprawdzić każdy kanał indywidualnie.

Czego brakuje: tabeli podsumowującej na górze strony ze wskaźnikiem zielono/czerwono per kanał, na przykład:

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

Brak zagregowanego opóźnienia

PmaControl monitoruje opóźnienie każdego kanału osobno, ale nie oblicza metryki zagregowanej. Na przykład:

- Maksymalne opóźnienie wśród wszystkich kanałów
- Średnie opóźnienie
- Liczba kanałów z błędami na tle całości

Te zagregowane metryki byłyby przydatne dla alertów: "przynajmniej jeden kanał ma ponad 60 sekund opóźnienia" jest bardziej trafne niż alerty kanał po kanale, gdy mamy 10 kanałów.

Brak grafu zależności między kanałami

W multi-source kanały są niezależne. Ale w pewnych architekturach istnieją zależności logiczne: kanał A musi być zastosowany przed kanałem B (na przykład tabele referencyjne przed tabelami transakcyjnymi).

PmaControl nie modeluje tych zależności. DBA musi ręcznie zarządzać kolejnością stosowania, jeśli jest to konieczne.

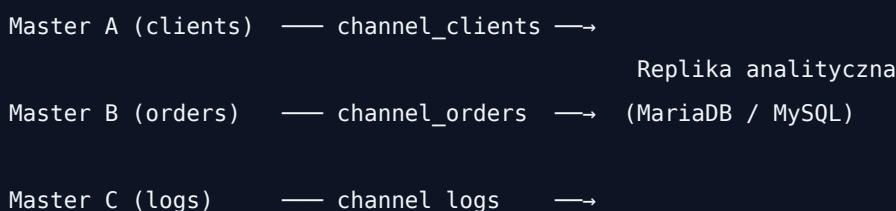
Brak alertingu per kanał

Obecne alerty działają na poziomie serwera, nie kanału. Jeśli kanał `channel_clients` się zatrzyma, ale `channel_orders` działa, alert wskazuje po prostu "replikacja w błędzie na serwerze X" bez precyzowania kanału.

Przypadki praktyczne

Konsolidacja analityczna

Typowa architektura:



Replika konsoliduje trzy bazy. PmaControl nadzoruje trzy kanały. Jeśli master C (logs) jest wolny, opóźnienie kanału `channel_logs` rośnie bez wpływu na dwa pozostałe.

W PmaControl zobaczysz trzy bloki na stronie slave repliki, z trzema niezależnymi wykresami opóźnień. Kanał `channel_logs` będzie miał własny pasek zdrowia w kolorze bursztynowym lub czerwonym, podczas gdy dwa pozostałe pozostaną na zielono.

Migracja przez kanał

Podczas migracji możesz tymczasowo mieć:

```
Stary master   — channel_legacy —→  
                                                    Nowa replika  
Nowy master   — channel_new   —→
```

Kanał `channel_legacy` zostanie usunięty po zakończeniu migracji. Podczas przejścia PmaControl nadzoruje oba kanały i pozwala sprawdzić, czy nowy kanał nadrobił zaległości, zanim odłączysz stary.

Filtrowanie po bazie

Z `REPLICATE_DO_DB` lub `REPLICATE_REWRITE_DB` każdy kanał może być filtrowany, aby replikować tylko wybrane bazy:

```
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 wyświetla aktywne filtry dla każdego kanału, co ułatwia diagnostykę, gdy oczekiwana tabela nie pojawia się na replice.

Mapa drogowa

Planowane ulepszenia wsparcia multi-source w PmaControl:

Zunifikowany dashboard multi-source

Dedykowany panel multi-source z:

- Widokiem macierzowym: replika w wierszu, jej kanały w kolumnach
- Globalnym wskaźnikiem zdrowia (zielony, jeśli wszystkie kanały OK, czerwony, jeśli przynajmniej jeden KO)
- Zagregowanym opóźnieniem (maks., średnia, liczba błędów)

Matryca zdrowia kanałów

Kompaktowa wizualizacja typu "heatmap":

	Pon	Wt	Śr	Czw	Pt	Sob	Ndz
ch_clients	☐	☐	☐	☐	☐	☐	☐
ch_orders	☐	☐	☐	☐	☐	☐	☐
ch_logs	☐	☐	☐	☐	☐	☐	☐

Alerting per kanał

Alerty Telegram specyficzne per kanał:

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

Automatyczne wykrywanie zależności

Analiza kluczy obcych cross-kanałowych w celu wykrycia potencjalnych zależności i alertowania, jeśli kanał zależny nabiera opóźnień.

Dobre praktyki multi-source

1. Nazywaj kanały jasno

Używaj opisowych nazw: `channel_clients`, `channel_orders`, nie `ch1`, `ch2`. PmaControl wyświetla nazwy takimi, jakie są — jasne nazwy ułatwiają diagnostykę.

2. Jeden kanał = jedna baza (lub logiczna grupa)

Unikaj mieszania niepowiązanych baz w jednym kanale. Jeśli kanał zatrzyma się na błędzie, wszystko co zawiera jest zablokowane.

3. Monitoruj przestrzeń relay log

Przy wielu kanałach przestrzeń dyskowa relay log może eksplodować. Każdy kanał ma swój własny relay log. Monitoruj `Relay_Log_Space` per kanał w PmaControl.

4. Testuj niezależne zatrzymywanie kanałów

Upewnij się, że zatrzymanie jednego kanału nie wpływa na pozostałe. PmaControl pozwala na STOP/START każdego kanału niezależnie — korzystaj z tej możliwości, aby walidować izolację.

5. Dokumentuj architekturę

Multi-source dodaje złożoności. Dokumentuj, który kanał niesie jaką bazę, z jakiego mastera, z jakimi filtrami. PmaControl wyświetla te informacje, ale zewnętrzna dokumentacja pozostaje przydatna przy onboardingu.

Podsumowanie

Replikacja multi-source to potężne narzędzie do konsolidacji danych MariaDB / MySQL.

PmaControl automatycznie wykrywa wiele kanałów i nadzoruje je indywidualnie za pomocą wykresów opóźnień, pasków zdrowia i akcji korekcyjnych per kanał.

Obecne ograniczenia — brak zunifikowanego widoku, brak alertingu per kanał, brak zagregowanego opóźnienia — są zidentyfikowane i stanowią część mapy drogowej. W międzyczasie nadzór kanał po kanale pokrywa podstawowe potrzeby: wiedzieć, który kanał ma opóźnienie, który kanał jest w błędzie, i móc na niego działać indywidualnie.

Dla DBA zarządzających architekturami multi-source w produkcji, PmaControl pozostaje najbardziej odpowiednim narzędziem — nawet niedoskonałym — ponieważ żaden konkurent nie oferuje tej widoczności per kanał od razu po instalacji.