Jeden odczyt MariaDB wystarcza do rozpoznania powtórzonej obserwacji alertu

Aurélien LEQUOY · November 21, 2026

PMACONTROL MARIADB MYROCKS PERFORMANCE SQL BENCHMARK

Gdy PmaControl dwukrotnie otrzyma ten sam pomiar, musi rozpoznać powtórzenie, żeby ponownie nie otworzyć ani nie zamknąć incydentu. Dotąd aplikacja najpierw pytała MariaDB o datę ostatniego pomiaru, a potem szukała właśnie otrzymanego punktu. **Usunęliśmy pierwszy odczyt.**

Dla pięciu rzeczywistych kontroli mediana czasu odczytów SQL przy powtórzeniu spadła **z 6,645 do 0,363 ms**, czyli **o 95%**. W laboratorium mediana całej transakcji, łącznie z blokadą i zapisami, spadła **z 2,554 do 0,904 ms**. Poprawka aplikacji jest przetestowana i czeka na wdrożenie.

Kontrola stref czasowych krok po kroku

21 listopada o 22:29:19 kontrola zgodności stref czasowych zgłosiła alarm. Co się stanie, gdy ten pomiar dotrze ponownie? Pierwotny kod zaczyna od poniższego zapytania. Znak `?` oznacza sygnaturę kontroli, tę samą w obu zapytaniach.

```
EXPLAIN PARTITIONS
SELECT observed_at
FROM pmc_alert_level_point
WHERE fingerprint = UNHEX(?)
ORDER BY observed_epoch DESC, observed_at DESC
LIMIT 1;
```

Aplikacja sprawdzała jedynie, czy istnieje wiersz historii. Następnie szukała dokładnego pomiaru i odczytywała jego stan. Wybrana wersja zaczyna od tego drugiego kroku. Zgodność punktu i stanu oznacza powtórzenie. Sprzeczne stany w tej samej chwili powodują błąd.

```
EXPLAIN PARTITIONS
SELECT is_open
FROM pmc_alert_level_point
WHERE fingerprint = UNHEX(?)
```

```
  AND observed_epoch = 1795296559
  AND observed_at = '2026-11-21 22:29:19.000000';
```

Plany z serwera dla tej samej kontroli. Pierwsza kolumna wyników przedstawia usunięty odczyt. Druga przedstawia odczyt wystarczający do rozpoznania powtórzenia.

EXPLAIN PARTITIONS	Ostatnia data	Dokładny punkt
Dostęp (`type`)	`range`	`const`
Indeks (`key`)	`PRIMARY`	`PRIMARY`
Bajty klucza (`key_len`)	32	48
Odniesienie (`ref`)	—	`const,const,const`
Szacowane wiersze (`rows`)	8 617	1
Partycje w planie	45	1 (`p740307`)
Extra	`Using where; Using index`	—

Oba zapytania używają `PRIMARY`. Pierwsze ustala tylko 32 bajty sygnatury i w planie zostaje 45 partycji. Drugie określa również epoch i datę, razem 48 bajtów klucza. MariaDB wskazuje jeden klucz w partycji właściwego dnia.

Wartość 8 617 w `rows` jest szacunkiem. W `ANALYZE FORMAT=JSON` iterator pierwszego zapytania zwrócił jeden wiersz (`r_rows = 1` , `r_loops = 1`), zgodnie z `LIMIT 1` . Te liczniki nie mierzą wyszukiwań wewnątrz partycji. Od 19 do 21 listopada digest zarejestrował **1 643 071 wywołań** i łączny czas SQL **8 godz. 08 min**.

Otwórz 20 pełnych planów MariaDB dla pięciu kontroli (tekst)

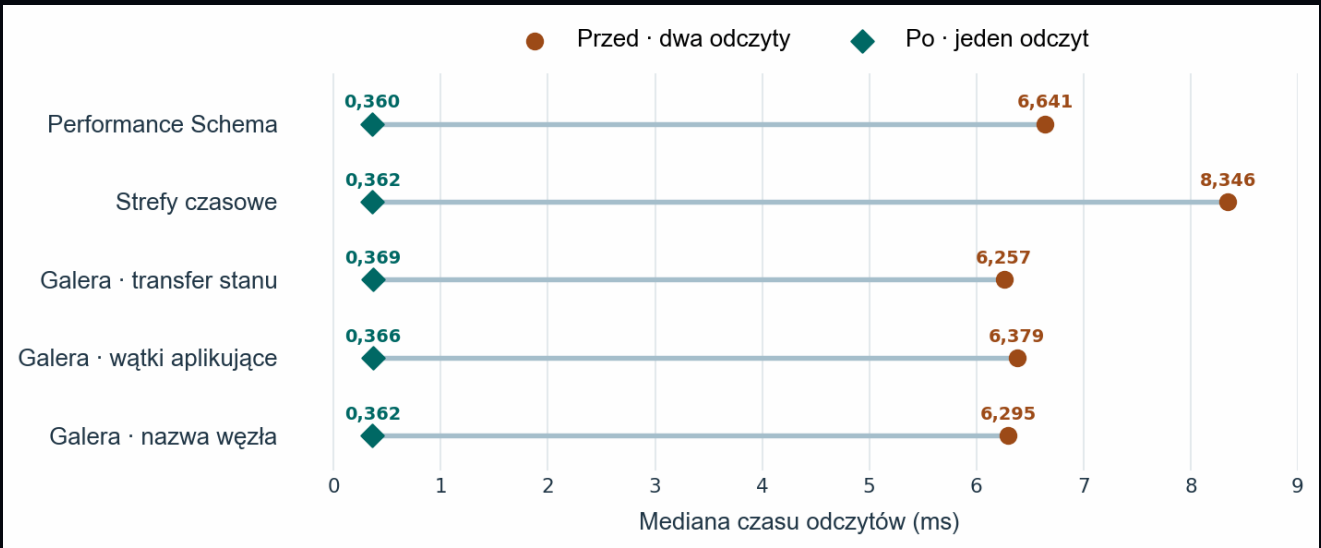
Pięć kontroli użytych w teście

Strona digestu pokazała pięć wywołań tego samego SQL z różnymi wartościami fingerprint. Każda identyfikuje jeden monitorowany warunek. Nazwę reguły odczytaliśmy z odpowiedniej obserwacji. Litera w nawiasie odpowiada oznaczeniu w plikach CSV.

Kontrola (oznaczenie CSV)	Kod reguły	Punkty alarmowe / wszystkie

Włączenie Performance Schema (A)	`AdvisorPerformanceSchema`	0 / 5 692
Zgodność stref czasowych (B)	`AdvisorTimeZoneAlignment`	5 694 / 5 695
Galera · metoda transferu stanu (SST) (C)	`AdvisorWsrepSstMethod`	0 / 5 692
Galera · wątki stosujące transakcje (D)	`AdvisorWsrepSlaveThreads`	0 / 5 692
Galera · nazwa węzła (E)	`AdvisorWsrepNodeName`	0 / 5 692

Wszystkie pięć odtwarzanych obserwacji nosi datę 21 listopada, 22:29:19. W tej chwili tylko kontrola stref czasowych zgłaszała alarm. Linia na wykresie łączy medianę początkową z medianą po przejściu od razu do dokładnego klucza.



Każdą kontrolę wykonano 50 razy na wersję na obserwowanym serwerze. Ostatni wiersz łączy 250 pomiarów każdej wersji. Obie tabele podają czas SQL w milisekundach.

Ponowne przetworzenie punktu · z dwóch odczytów do jednego

Kontrola	Średnia przed	Średnia po	Mediana przed	Mediana po
Włączenie Performance Schema	7,171	0,372	6,641	0,360
Zgodność stref czasowych	8,808	0,400	8,346	0,362
Galera · metoda transferu stanu (SST)	7,372	0,575	6,257	0,369

Kontrola	Średniaprzed	Średniapo	Medianaprzed	Medianapo
Galera · wątki stosujące transakcje	6,970	0,389	6,379	0,366
Galera · nazwa węzła	7,132	0,374	6,295	0,362
Pięć kontroli	7,491	0,422	6,645	0,363

Dodanie punktu do historii · z czterech odczytów do trzech

Kontrola	Średnia przed	Średnia po	Mediana przed	Mediana po
Włączenie Performance Schema	39,832	29,522	29,189	26,465
Zgodność stref czasowych	41,518	32,461	35,533	29,123
Galera · metoda transferu stanu (SST)	39,431	31,138	31,509	28,157
Galera · wątki stosujące transakcje	36,855	31,667	33,790	25,943
Galera · nazwa węzła	39,696	31,043	35,195	28,511
Pięć kontroli	39,466	31,166	33,243	27,975

Dlaczego dopisanie punktu przyspiesza mniej

Opóźniony punkt może trafić między dwa zapisane pomiary i zmienić granice incydentu. Gdy nie ma dokładnego klucza, aplikacja wyszukuje poprzednika i następnika według `(observed_epoch, observed_at)`. Obecność dowolnego sąsiada potwierdza też istnienie historii.

To plany obu odczytów dla kontroli stref czasowych i parametru dopisania z 22 listopada, 22:29:19. Oba nadal kosztują czas po usunięciu zapytania o ostatnią datę.

Sąsiad	Dostęp	Indeks	Partycje	Szacowane wiersze
Poprzednik	`range`	`PRIMARY`	45	8 653
Następnik	`range`	`PRIMARY`	45	81

V1 to kod pierwotny. **V2** pomija zapytanie o ostatnią datę i korzysta z dokładnego punktu i jego sąsiadów. Gdy wszystkie trzy wyszukiwania są puste, sprawdza stan pozostawiony przez

wcześniejszą instalację. W pierwszej serii mediana odtwarzania wzrosła z 9,665 do 12,963 ms, bo V2 ponownie szukała sąsiadów w pustej historii.

Wybrana **V3** obsługuje ten przypadek bezpośrednio. Gdy historia jest pusta i brak starszego incydentu, odtwarza jedyny zapisany prawidłowy punkt z jego datą, epoch i treścią. Potem sprawdza nadchodzący punkt pod tą samą blokadą.

Cała transakcja w laboratorium, średnia / mediana w ms. Czas PHP obejmuje uzyskanie blokady, odczyty, zapisy, `COMMIT` i zwolnienie blokady.

Operacja	V1 · pierwotna	V2 · bez zapytania wstępnego	V3 · bezpośrednie odtworzenie
Powtórzenie punktu	3,954 / 2,554	1,586 / 0,924	1,518 / 0,904
Dopisanie do historii	23,907 / 17,297	23,516 / 13,954	22,895 / 14,017
Utworzenie pierwszego punktu	28,373 / 10,477	20,031 / 8,689	22,708 / 8,660
Odtworzenie starszego prawidłowego punktu	23,609 / 10,422	29,137 / 13,423	17,151 / 7,916

Największa różnica między V2 i V3 dotyczy odtwarzania. Mediany dopisania są zbliżone. Wszystkie trzy wersje dały identyczne decyzje, punkty, incydenty, przedziały i ostatnie stany.

Czy fingerprint może być kluczem głównym?

20 listopada naliczyliśmy **543 719 punktów dla 9 478 sygnatur**. Jedna sygnatura pojawiła się 83 razy tego dnia. Sygnatura identyfikuje kontrolę, a trójka `(fingerprint, observed_epoch, observed_at)` jedno jej wykonanie.

`SHOW INDEX` oszacował 3 672 630 różnych wartości dla pierwszego prefiksu i 2 448 420 dla obu dłuższych. Rzeczywista liczność nie może zmaleć po dodaniu kolumny. Spadek wynika z szacowania. Na kopii z partycjami dziennymi `PRIMARY KEY(fingerprint)` zwrócił błąd 1503, ponieważ każdy klucz unikatowy musi obejmować `observed_at`. Na kopii bez partycji drugi punkt o tej samej sygnaturze dał błąd duplikatu 1062. MariaDB · SHOW INDEX · MariaDB · Partitioning

Inne sposoby sprawdzenia, czy historia istnieje

Czasy obejmują wszystkie pięć kontroli. Pokazany plan dotyczy **zgodności stref czasowych**. Strzałka oznacza przejście do całej historii przy braku trafienia, a kreska oznacza tabelę bez partycji. Każda seria ma własny pomiar odniesienia.

Seria i zapytanie	Dostęp / indeks	Partycje	Średnia / mediana (ms)
I · Ostatnia data	`range / PRIMARY`	45	7,577 / 6,660
I · Ostatnia data, `FORCE INDEX (PRIMARY)`	`ref / PRIMARY`	45	4,927 / 4,283
I · `SELECT 1 … LIMIT 1`	`ref / level_neighbour`	45	6,361 / 5,434
I · `SELECT EXISTS(…)`	`ref / level_neighbour`	45	6,870 / 5,625
I · Ostatni stan potwierdzony w historii, z wyszukaniem awaryjnym	`const / PRIMARY` + `const / PRIMARY` → `ref / level_neighbour`	— + 45 → 45	4,543 / 3,995
II · Ostatnia data	`range / PRIMARY`	45	6,401 / 5,623
II · Najpierw dzień obserwacji, potem cała historia	`ref / level_neighbour_epoch` → `ref / level_neighbour`	1 → 45	3,739 / 3,411
II · Tylko ostatni stan	`const / PRIMARY`	—	0,277 / 0,257

Sam odczyt ostatniego stanu zwrócił tę samą datę w 250 porównaniach na 250. Tabela może jednak zachować stan starszej wersji lub stan sprzed usunięcia punktów przez retencję. Nie wystarcza do rozstrzygnięcia, czy trzeba odbudować historię.

Zapytania SQL i plany wszystkich wariantów (tekst)

Wpływ indeksu czasu i partycjonowania

Porównaliśmy to samo uporządkowane zapytanie na **621 865 identycznych punktach** w MariaDB 11.8.10 z MyRocks. Pokazane plany dotyczą stref czasowych. Badany indeks to `when_observed(observed_epoch, observed_at)`.

Struktura	Dostęp / indeks	Partycje w planie	Średnia / mediana (ms)
Klucz złożony, partycje dzienne	`range / PRIMARY`	45	612,119 / 549,937
Dodany indeks czasu	`range / PRIMARY`	45	637,779 / 525,603
Wymuszony indeks czasu	`index / when_observed`	45	64,429 / 48,306
Tabela bez partycji	`range / PRIMARY`	—	14,052 / 7,676
16 partycji według sygnatury	`ref / PRIMARY`	1	0,786 / 0,410

Po dodaniu indeksu optymalizator nadal wybierał `PRIMARY` dla pięciu kontroli. Wymuszony `when_observed` daje dostęp `index` według dat aż do trafienia na sygnaturę. Podział według sygnatury ogranicza wyszukiwanie do jednej z 16 partycji i wymaga innej strategii dziennego usuwania danych.

Laboratorium miało jeden CPU i 96 MiB pamięci podręcznej bloków MyRocks. Mediana bazowa wynosiła 1,350 ms na pierwszej kopii z 571 957 punktami, a 549,937 ms na rozszerzonej kopii. Przyczyna różnicy między seriami pozostaje nieustalona. Wyniki porównują struktury wewnątrz jednej serii. Koszt migracji i czasy produkcyjne wymagają odrębnego pomiaru.

Metoda i dane do pobrania

Zapisaliśmy pięć przykładów pokazanych jednocześnie na stronie digestu. W każdej z **50 rund** mieszaliśmy parametry i warianty przy stałym ziarnie generatora. To **50 pomiarów na kontrolę i wariant oraz 250 na wariant**. Uwzględniliśmy pierwsze wywołania i najwyższe czasy. Na serwerze korzystaliśmy z migawek tylko do odczytu, wyłączonej pamięci podręcznej wyników i `SHOW PROFILES`.

Transakcje z zapisami wykonano na odrębnych kopiach z 45 partycjami, 28 463 historycznymi kluczami pięciu kontroli i syntetyczną treścią alertów. Testy objęły również spóźnione obserwacje, zmianę czasu, konflikty i ponowienie po wycofaniu. Archiwum zawiera pojedyncze pomiary, warianty SQL i pełne plany dla kontroli.

Pobierz pomiary, nazwy pięciu reguł i plany SQL (ZIP)

Wybór dla PmaControl

Wróćmy do obserwacji stref czasowych. W 50 powtórzeniach średni czas odczytu spadł **z 8,808 do 0,400 ms**, a mediana **z 8,346 do 0,362 ms**. Do decyzji wystarczają kontrola, chwila pomiaru i zapisany stan. **Zachowujemy V3 i złożony klucz główny**. Powtórzenie wymaga jednego wyszukania, a historia nadal obsługuje spóźnione punkty.