# Одного чтения MariaDB достаточно, чтобы распознать повтор наблюдения

**Aurélien LEQUOY** · November 21, 2026

PMACONTROL MARIADB MYROCKS PERFORMANCE SQL BENCHMARK

Если PmaControl дважды получает одно измерение, ему нужно распознать повтор, чтобы снова не открыть и не закрыть инцидент. Раньше приложение запрашивало у MariaDB дату последнего измерения, а затем искало поступившую точку. **Первое чтение мы убрали.**

Для пяти реальных проверок медиана чтений SQL при повторе снизилась **с 6,645 до 0,363 мс**, то есть **на 95%**. В лаборатории медиана полной транзакции с блокировкой и записью снизилась **с 2,554 до 0,904 мс**. Исправление приложения проверено и ожидает развёртывания.

## Проверка часовых поясов по шагам

21 ноября в 22:29:19 проверка согласованности часовых поясов показала тревогу. Что произойдёт, если измерение поступит повторно? Исходный код начинал со следующего запроса. Знак `?` обозначает отпечаток проверки, одинаковый в обоих запросах.

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

По наличию строки приложение определяло, существует ли история. Затем оно искало точное измерение и читало его состояние. Выбранная версия начинает с этого шага. Совпадение точки и состояния означает повтор. Противоречащие состояния в один момент вызывают ошибку.

```
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';
```

**Планы, полученные на сервере для этой проверки.** Первый столбец результатов показывает удалённое чтение, второй — чтение, достаточное для повтора.

| EXPLAIN PARTITIONS | Последняя дата | Точная точка |
|---|---|---|
| Доступ ( `type` ) | `range` | `const` |
| Индекс ( `key` ) | `PRIMARY` | `PRIMARY` |
| Байты ключа ( `key_len` ) | 32 | 48 |
| Ссылка ( `ref` ) | — | `const,const,const` |
| Оценка строк ( `rows` ) | 8 617 | 1 |
| Секции в плане | 45 | 1 ( `p740307` ) |
| Extra | `Using where; Using index` | — |

Оба запроса используют `PRIMARY`. В первом заданы только 32 байта отпечатка, и в плане остаются все 45 секций. Во втором заданы также epoch и дата, то есть все 48 байт ключа. MariaDB находит один ключ в секции нужного дня.

Значение 8 617 в `rows` — оценка. В `ANALYZE FORMAT=JSON` итератор первого запроса вернул одну строку ( `r_rows = 1` , `r_loops = 1` ), как и предполагает `LIMIT 1` . Эти счётчики не измеряют поиск внутри секций. С 19 по 21 ноября дайджест зарегистрировал **1 643 071 вызов**, суммарное время SQL составило **8 ч 08 мин**.

Открыть все 20 планов MariaDB для пяти проверок (текст)

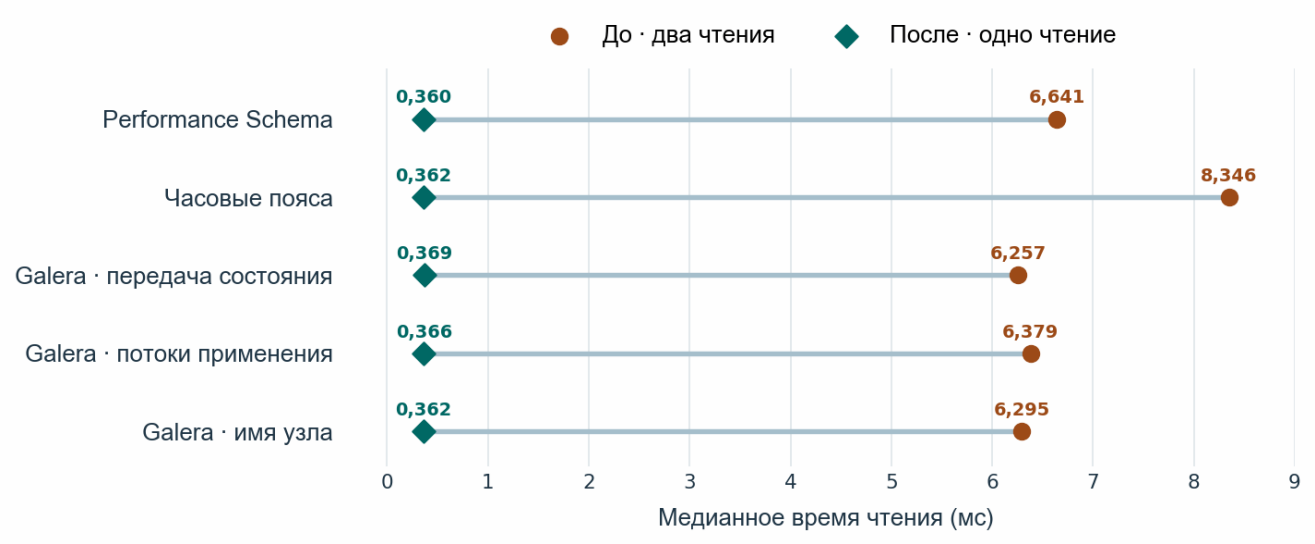
## Какие пять проверок участвовали

На странице дайджеста были пять выполнений одного SQL с разными значениями fingerprint. Каждое значение соответствует конкретной проверке. Названия правил взяты из самих наблюдений. Буква в скобках соответствует обозначению в CSV.

| Проверка (метка CSV) | Код правила | Тревожные точки / всего |
|---|---|---|

| | | |
|---|---|---|
| Включение Performance Schema (A) | `AdvisorPerformanceSchema` | 0 / 5 692 |
| Согласованность часовых поясов (B) | `AdvisorTimeZoneAlignment` | 5 694 / 5 695 |
| Galera · метод передачи состояния (SST) (C) | `AdvisorWsrepSstMethod` | 0 / 5 692 |
| Galera · потоки применения транзакций (D) | `AdvisorWsrepSlaveThreads` | 0 / 5 692 |
| Galera · имя узла (E) | `AdvisorWsrepNodeName` | 0 / 5 692 |

Все пять повторных измерений относятся к 21 ноября, 22:29:19. В этот момент тревогу показывала только проверка часовых поясов. На графике линия соединяет исходную медиану и медиану при обращении сразу по точному ключу.




Каждую проверку выполнили по 50 раз для каждой версии на наблюдаемом сервере. Последняя строка объединяет 250 измерений версии. В обеих таблицах время SQL указано в миллисекундах.

## Повторная обработка точки · с двух чтений до одного

| Проверка | Среднее до | Среднее после | Медиана до | Медиана после |
|---|---|---|---|---|
| Включение Performance Schema | 7,171 | 0,372 | 6,641 | 0,360 |
| Согласованность часовых поясов | 8,808 | 0,400 | 8,346 | 0,362 |

| Проверка | Среднее до | Среднее после | Медиана до | Медиана после |
|---|---|---|---|---|
| Galera · метод передачи состояния (SST) | 7,372 | 0,575 | 6,257 | 0,369 |
| Galera · потоки применения транзакций | 6,970 | 0,389 | 6,379 | 0,366 |
| Galera · имя узла | 7,132 | 0,374 | 6,295 | 0,362 |
| Все пять проверок | 7,491 | 0,422 | 6,645 | 0,363 |

### Добавление точки в историю · с четырёх чтений до трёх

| Проверка | Среднее до | Среднее после | Медиана до | Медиана после |
|---|---|---|---|---|
| Включение Performance Schema | 39,832 | 29,522 | 29,189 | 26,465 |
| Согласованность часовых поясов | 41,518 | 32,461 | 35,533 | 29,123 |
| Galera · метод передачи состояния (SST) | 39,431 | 31,138 | 31,509 | 28,157 |
| Galera · потоки применения транзакций | 36,855 | 31,667 | 33,790 | 25,943 |
| Galera · имя узла | 39,696 | 31,043 | 35,195 | 28,511 |
| Все пять проверок | 39,466 | 31,166 | 33,243 | 27,975 |

## Почему добавление новой точки ускоряется меньше

Запоздавшее наблюдение может оказаться между двумя точками и изменить границы инцидента. Если точного ключа нет, приложение ищет предыдущую и следующую точки в порядке `(observed_epoch, observed_at)`. Нахождение любой из них одновременно подтверждает наличие истории.

Это планы двух чтений для проверки часовых поясов с параметром добавления от 22 ноября, 22:29:19. Даже после исключения запроса последней даты оба чтения требуют

времени.

| Сосед | Доступ | Индекс | Секции | Оценка строк |
|---|---|---|---|---|
| Предыдущая точка | `range` | `PRIMARY` | 45 | 8 653 |
| Следующая точка | `range` | `PRIMARY` | 45 | 81 |

**V1** — исходная версия. **V2** убирает запрос последней даты и использует точный ключ и соседей. Если все три поиска пусты, она проверяет, сохранилось ли состояние старой установки. В первой серии медиана восстановления выросла с 9,665 до 12,963 мс: V2 повторила поиск соседей в заведомо пустой истории.

Выбранная **V3** обрабатывает этот случай напрямую. Если история пуста и старых инцидентов нет, она записывает единственную сохранённую здоровую точку с исходной датой, epoch и содержимым. Затем под той же блокировкой проверяет входящее наблюдение.

**Полная лабораторная транзакция, среднее / медиана в мс.** Таймер PHP охватывает получение блокировки, чтения, записи, `COMMIT` и снятие блокировки.

| Операция | V1 · исходная | V2 · без первого запроса | V3 · прямое восстановление |
|---|---|---|---|
| Повтор точки | 3,954 / 2,554 | 1,586 / 0,924 | 1,518 / 0,904 |
| Добавление к истории | 23,907 / 17,297 | 23,516 / 13,954 | 22,895 / 14,017 |
| Первая точка | 28,373 / 10,477 | 20,031 / 8,689 | 22,708 / 8,660 |
| Восстановление старой здоровой точки | 23,609 / 10,422 | 29,137 / 13,423 | 17,151 / 7,916 |

Основное различие V2 и V3 проявилось при восстановлении. Медианы добавления близки. Все три версии дали одинаковые решения, точки, инциденты, интервалы и последние состояния.

## Может ли fingerprint быть первичным ключом?

20 ноября мы насчитали **543 719 точек и 9 478 отпечатков**. Один отпечаток встретился за день 83 раза. Отпечаток обозначает проверку, а тройка `(fingerprint, observed_epoch, observed_at)` — отдельное измерение этой проверки.

`SHOW INDEX` оценил первый префикс в 3 672 630 различных значений, а каждый более длинный в 2 448 420. Точное количество не может снизиться при добавлении столбца. Снижение объясняется оценками. В копии с суточными секциями `PRIMARY KEY(fingerprint)` дал ошибку 1503, поскольку каждый уникальный ключ должен включать `observed_at`. Без секций вторая точка с тем же отпечатком дала ошибку дубликата 1062. MariaDB · SHOW INDEX · MariaDB · Partitioning

## Другие способы проверить наличие истории

Время рассчитано для пяти проверок вместе. План показан для **согласованности часовых поясов**. Стрелка обозначает обращение ко всей истории после промаха, тире — таблицу без секций. В каждой серии есть собственный исходный замер.

| Серия и запрос | Доступ / индекс | Секции | Среднее / медиана (мс) |
|---|---|---|---|
| I · Последняя дата | `range / PRIMARY` | 45 | 7,577 / 6,660 |
| I · Последняя дата, `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 · Проверка последнего состояния в истории с резервным поиском | `const / PRIMARY` + `const / PRIMARY` → `ref / level_neighbour` | — + 45 → 45 | 4,543 / 3,995 |
| II · Последняя дата | `range / PRIMARY` | 45 | 6,401 / 5,623 |
| II · Поиск в дне наблюдения с резервным поиском | `ref / level_neighbour_epoch` → `ref / level_neighbour` | 1 → 45 | 3,739 / 3,411 |
| II · Только последнее состояние | `const / PRIMARY` | — | 0,277 / 0,257 |

Чтение только последнего состояния вернуло ту же дату в 250 сравнениях из 250. Однако в таблице может остаться состояние от старой версии или от точек, удалённых по сроку хранения. Этого чтения недостаточно для решения о восстановлении истории.

Запросы SQL и планы всех вариантов (текст)

## Что дают индекс времени и секционирование

Для сравнения структур исходный запрос с сортировкой выполнялся на **621 865 одинаковых точках** в MariaDB 11.8.10 с MyRocks. Планы приведены для часовых поясов. Проверяли индекс `when_observed(observed_epoch, observed_at)`.

| Структура | Доступ / индекс | Секции в плане | Среднее / медиана (мс) |
|---|---|---|---|
| Составной ключ, суточные секции | `range / PRIMARY` | 45 | 612,119 / 549,937 |
| Добавлен индекс времени | `range / PRIMARY` | 45 | 637,779 / 525,603 |
| Принудительный индекс времени | `index / when_observed` | 45 | 64,429 / 48,306 |
| Таблица без секций | `range / PRIMARY` | — | 14,052 / 7,676 |
| 16 секций по отпечатку | `ref / PRIMARY` | 1 | 0,786 / 0,410 |

После добавления индекса оптимизатор сохранил `PRIMARY` для всех пяти проверок. Принудительный `when_observed` даёт доступ `index` с перебором дат до нужного отпечатка. При секционировании по отпечатку поиск указывает на одну из 16 секций, но способ ежедневной очистки придётся изменить.

В лаборатории выделили один CPU и 96 МиБ кеша блоков MyRocks. Медиана исходного запроса на первой копии из 571 957 точек была 1,350 мс, на расширенной копии — 549,937 мс. Причина различия между сериями не установлена. Эти данные сравнивают структуры внутри одной серии. Задержки в рабочей среде и затраты на миграцию требуют отдельного измерения.

## Методика и исходные данные

Мы сохранили пять одновременно показанных примеров. В каждом из **50 раундов** перемешивали порядок параметров и вариантов с фиксированным начальным значением генератора. Получили **50 замеров на проверку и вариант, 250 на вариант**. Учтены первые запуски и пики задержки. На сервере использовались снимки только для чтения, отключённый кеш результатов и `SHOW PROFILES`.

Транзакции с записью выполняли на отдельных копиях с 45 секциями, 28 463 историческими ключами пяти проверок и синтетическим содержимым оповещений. Проверили запоздалые точки, перевод часов, конфликты и повтор после отката. Архив содержит все замеры, варианты SQL и полные планы для проверок.

Скачать замеры, соответствие пяти проверок и планы SQL (ZIP)

## Выбор для PmaControl

Вернёмся к измерению часовых поясов. В 50 повторах среднее время чтений снизилось **с 8,808 до 0,400 мс**, медиана — **с 8,346 до 0,362 мс**. Решение определяется проверкой, моментом и записанным состоянием. **Оставляем V3 и составной первичный ключ.** Повтор требует одного поиска, а история позволяет обработать следующую точку, даже если она опоздает.