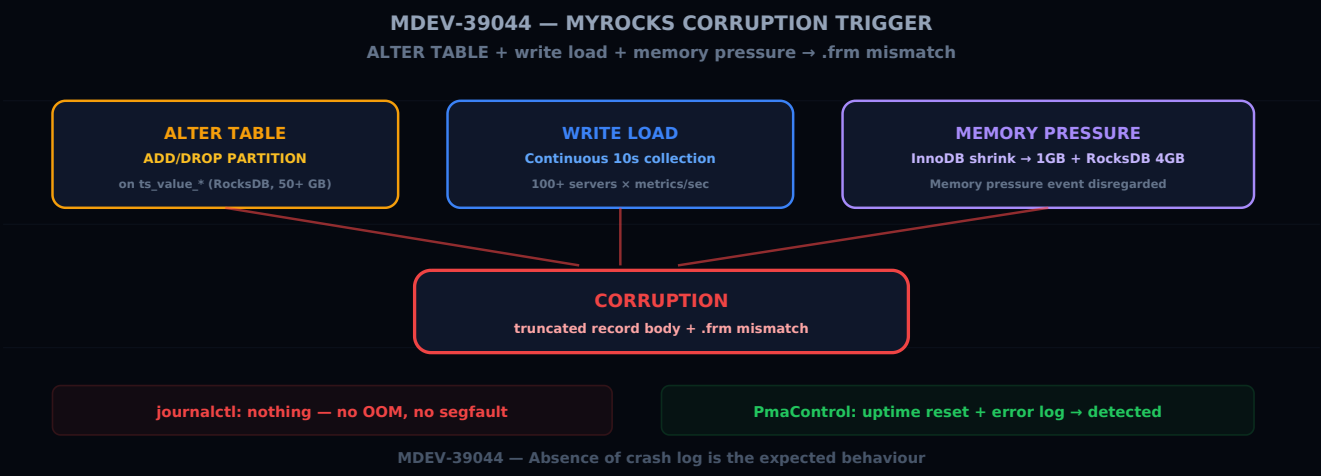
MyRocks pod obciążeniem: gdy ALTER TABLE wywołuje korupcję

Aurélien LEQUOY · March 6, 2026

MARIADB ROCKSDB CORRUPTION DDL INCIDENT-RESPONSE MDEV-39044



Kontekst

6 marca 2026 produkcyjny serwer MariaDB 10.11.15 nadzorowany przez PmaControl doznał **poważnego incydentu**. W odróżnieniu od typowych awarii (OOM, segfault), ten miał bezprecedensowe objawy:

```
RocksDB: Error opening instance, Status Code: 2,
  Status: Corruption: truncated record body
Incorrect information in file: './pmacontrol/ts_value_general_int.frm'
Can't init tc log
Aborting
```

Serwer restartował się w pętli kilka razy, zanim się ustabilizował, z błędami `.frm mismatch` na kilku tabelach szeregów czasowych.

Zgłoszenie MDEV-39044

Po zbadaniu skorelowaliśmy ten incydent ze zgłoszeniem MariaDB **MDEV-39044**:

> *MyRocks corruption after restart during/after ALTER workload: Corruption: truncated record body, .frm mismatch, no crash log, no OOM killer*

Co opisuje zgłoszenie

Zgłoszenie dokumentuje odtwarzalny scenariusz korupcji:

1. **Wolumenowe tabele RocksDB z partycjonowaniem** — dokładnie to, czego PmaControl używa do metryk (tabele `ts_value_*` partycjonowane według dni)
2. `ALTER TABLE` **pod obciążeniem zapisu** — dodawanie partycji, gdy aplikacja ciągle pisze
3. **Jednoczesna presja pamięciowa InnoDB** — tabele InnoDB i RocksDB współistnieją na tym samym serwerze
4. **Brak śladu kernelowego** — brak OOM killera, brak segfault, brak logu awarii

Dlaczego to podstępne

Najniebezpieczniejszy punkt zgłoszenia: **brak logu awarii jest oczekiwanym zachowaniem w tym scenariuszu**. Serwer restartuje się, wykonuje `InnoDB crash recovery`, ale metadane RocksDB są uszkodzone (`.frm mismatch`).

DBA patrzący tylko na `journalctl` lub `dmesg` nic nie znajdzie. Sklasyfikuje incydent jako "niewyjaśniony restart" i przejdzie dalej.

Nasz konkretny przypadek

Dotknięte tabele

Wszystkie to tabele RocksDB partycjonowane według dni, masowo obciążone zapisem:

- `ts_value_general_int` — metryki całkowite (zmienne statusu, liczniki)
- `ts_value_general_json` — złożone metryki JSON
- `ts_mysql_digest_stat` — statystyki zapytań (digesty)
- `ts_value_general_text` — metryki tekstowe
- `ts_value_slave_int` — metryki replikacji
- `ts_value_slave_text` — szczegółowe stany replikacji

Prawdopodobny wyzwalacz

PmaControl automatycznie zarządza partycjami tych tabel: dodawanie partycji na następny dzień, usuwanie wygasłych partycji. To operacje `ALTER TABLE ... ADD PARTITION` / `DROP PARTITION` wykonywane na tabelach o rozmiarze kilkudziesięciu GB, **podczas gdy workery zbierające ciągle zapisują** (co 10 sekund na nadzorowany serwer).

Sygnały presji pamięciowej

Przed awarią log MariaDB pokazuje:

```
InnoDB: Memory pressure event disregarded
```

Zgłoszenie MDEV-39044 wyraźnie wymienia ten wzorzec jako czynnik obciążający. Presja pamięciowa InnoDB nie powoduje bezpośrednio korupcji, ale tworzy kontekst, w którym DDL na RocksDB staje się nieatomowy.

Jak PmaControl wykrył incydent

1. **Reset uptime** wykryty w 10 sekund przez szereg czasowy `ts_variable.uptime`
2. **Alert Telegram** wysłany natychmiast
3. **Automatyczna korelacja** z error logiem: wykrycie sygnatur `crash recovery` + `truncated record body`
4. **Analiza retrospektywna**: metryki z poprzedniej godziny (wątki, pamięć, CPU) były normalne — potwierdzając, że to nie klasyczny problem obciążenia

Rekomendacje

Natychmiastowe działania

1. **Nie wykonywać DDL na tabelach RocksDB pod obciążeniem zapisu**. Planować `ALTER TABLE ... ADD/DROP PARTITION` podczas okien niskiej aktywności.
2. **Monitorować błędy** `.frm` w error logu. To pierwszy wskaźnik korupcji po DDL.
3. **Śledzić zgłoszenie MDEV-39044** w oczekiwaniu na oficjalną poprawkę.

Działania strukturalne

4. **Rozdzielić silniki**: jeśli to możliwe, nie mieszać InnoDB i RocksDB na tym samym serwerze dla krytycznych tabel.
5. **Rozważyć migrację gorących tabel do InnoDB**. RocksDB jest doskonały dla sekwencyjnego zapisu, ale jego DDL nie są atomowe pod obciążeniem.
6. **Odpowiednio wymiarować pamięć**, by uniknąć presji InnoDB, która pogarsza problem. Zobacz nasz artykuł o OOM killerze dla obliczenia najgorszego przypadku.

Czym to nie jest

- To **nie** jest problem sprzętowy (dysk, RAM)
- To **nie** jest problem konfiguracji MySQL (parametry są poprawne)
- To **nie** jest odtwarzalne na żądanie (to race condition w silniku RocksDB/DDL)

To **bug silnika** udokumentowany przez sam zespół MariaDB.

Podsumowanie

MDEV-39044 to przypomnienie, że używanie alternatywnych silników przechowywania (RocksDB, TokuDB) na produkcyjnych obciążeniach wymaga szczególnej czujności wobec DDL. Brak logu awarii nie oznacza braku korupcji.

PmaControl wykrywa te incydenty przez nadzór `uptime` + korelację z error logiem, tam gdzie klasyczne narzędzia nic nie widzą.