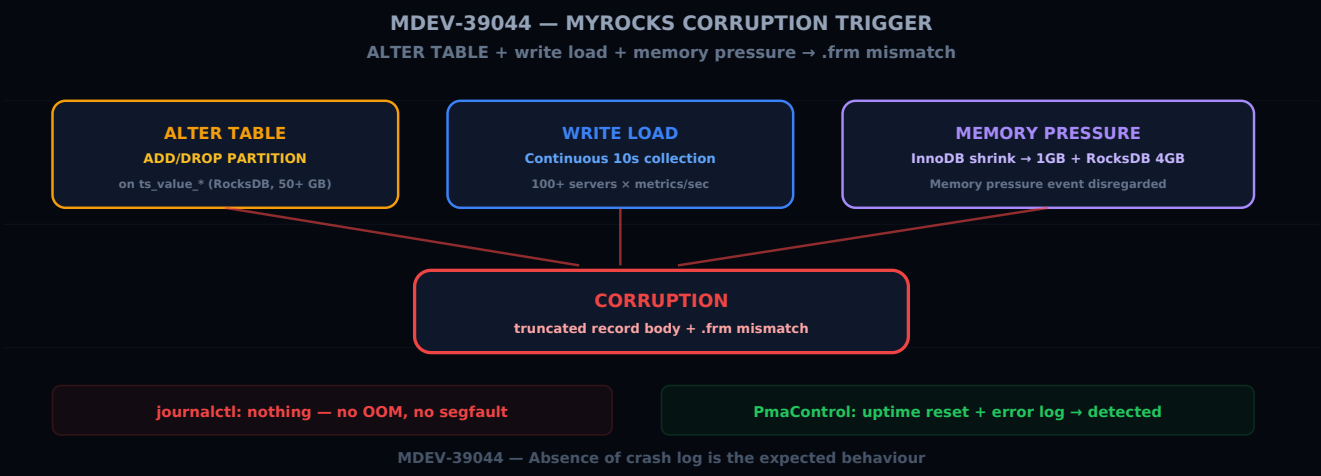
MyRocks под нагрузкой: когда ALTER TABLE вызывает коррупцию

Aurélien LEQUOY · March 6, 2026

MARIADB ROCKSDB CORRUPTION DDL INCIDENT-RESPONSE MDEV-39044



Контекст

6 марта 2026 года продакшн-сервер MariaDB 10.11.15, контролируемый PmaControl, испытал **серьёзный инцидент**. В отличие от обычных сбоев (OOM, segfault), этот имел беспрецедентные симптомы:

```
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
```

Сервер перезапускался в цикле несколько раз, прежде чем стабилизироваться, с ошибками `.frm mismatch` на нескольких таблицах временных рядов.

Тикет MDEV-39044

После расследования мы коррелировали этот инцидент с тикетом MariaDB **MDEV-39044**:

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

Что описывает тикет

Тикет документирует воспроизводимый сценарий коррупции:

1. **Объёмные партиционированные таблицы RocksDB** — именно то, что PmaControl использует для метрик (таблицы `ts_value_*`, партиционированные по дням)
2. `ALTER TABLE` **под нагрузкой записи** — добавление партиций, пока приложение непрерывно пишет
3. **Одновременное давление памяти InnoDB** — таблицы InnoDB и RocksDB сосуществуют на одном сервере
4. **Никаких следов в ядре** — нет OOM killer, нет segfault, нет crash log

Почему это коварно

Самый опасный аспект тикета: **отсутствие crash log — это ожидаемое поведение в этом сценарии**. Сервер перезапускается, выполняет `InnoDB crash recovery`, но метаданные RocksDB повреждены (`.frm mismatch`).

DBA, который смотрит только в `journalctl` или `dmesg`, ничего не найдёт. Он классифицирует инцидент как «необъяснимый перезапуск» и пойдёт дальше.

Наш конкретный случай

Затронутые таблицы

Все — партиционированные по дням таблицы RocksDB, массово нагруженные записью:

- `ts_value_general_int` — целочисленные метрики (status variables, счётчики)
- `ts_value_general_json` — сложные JSON-метрики
- `ts_mysql_digest_stat` — статистика запросов (дайджесты)
- `ts_value_general_text` — текстовые метрики
- `ts_value_slave_int` — метрики репликации
- `ts_value_slave_text` — детальные состояния репликации

Вероятный триггер

PmaControl автоматически обслуживает партиции этих таблиц: добавление партиции следующего дня, удаление просроченных партиций. Это `ALTER TABLE ... ADD PARTITION` / `DROP PARTITION`, выполняемые на таблицах в десятки гигабайт, **пока воркеры сбора метрик непрерывно пишут** (каждые 10 секунд на каждый контролируемый сервер).

Сигналы давления памяти

Перед сбоем лог MariaDB показывает:

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

Тикет MDEV-39044 прямо указывает этот паттерн как усугубляющий фактор. Давление памяти InnoDB не вызывает коррупцию напрямую, но создаёт контекст, в котором DDL RocksDB становится неатомарным.

Как PmaControl обнаружил инцидент

1. **Сброс uptime** обнаружен за 10 секунд через временной ряд `ts_variable.uptime`
2. **Оповещение Telegram** отправлено немедленно
3. **Автоматическая корреляция** с error log: обнаружение сигнатур `crash recovery` + `truncated record body`
4. **Ретроспективный анализ**: метрики предыдущего часа (потоки, память, CPU) были нормальными — подтверждая, что это не проблема классической нагрузки

Рекомендации

Немедленные действия

1. **Не выполнять DDL на таблицах RocksDB под нагрузкой записи**. Планировать `ALTER TABLE ... ADD/DROP PARTITION` на окна низкой активности.
2. **Мониторить ошибки** `.frm` в error log. Это первый индикатор коррупции после DDL.
3. **Следить за тикетом MDEV-39044** для получения официального исправления.

Структурные действия

4. **Разделить движки**: по возможности не смешивать InnoDB и RocksDB на одном сервере для критических таблиц.
5. **Рассмотреть миграцию горячих таблиц на InnoDB**. RocksDB отлично подходит для последовательной записи, но его DDL не атомарны под нагрузкой.
6. **Рассчитать память** для предотвращения давления InnoDB, которое усугубляет проблему. Смотрите нашу статью об OOM killer для расчёта наихудшего случая.

Чем это не является

- Это **не** проблема оборудования (диск, RAM)
- Это **не** проблема конфигурации MySQL (параметры корректны)
- Это **не** воспроизводится по требованию (это race condition в движке RocksDB/DDL)

Это **баг движка**, задокументированный самой MariaDB.

Заключение

MDEV-39044 — это напоминание о том, что использование альтернативных движков хранения (RocksDB, TokuDB) на продакшн-нагрузках требует особой бдительности при DDL. Отсутствие crash log не означает отсутствие коррупции.

PmaControl обнаруживает такие инциденты через мониторинг `uptime` + корреляцию error log, там, где классические инструменты ничего не видят.