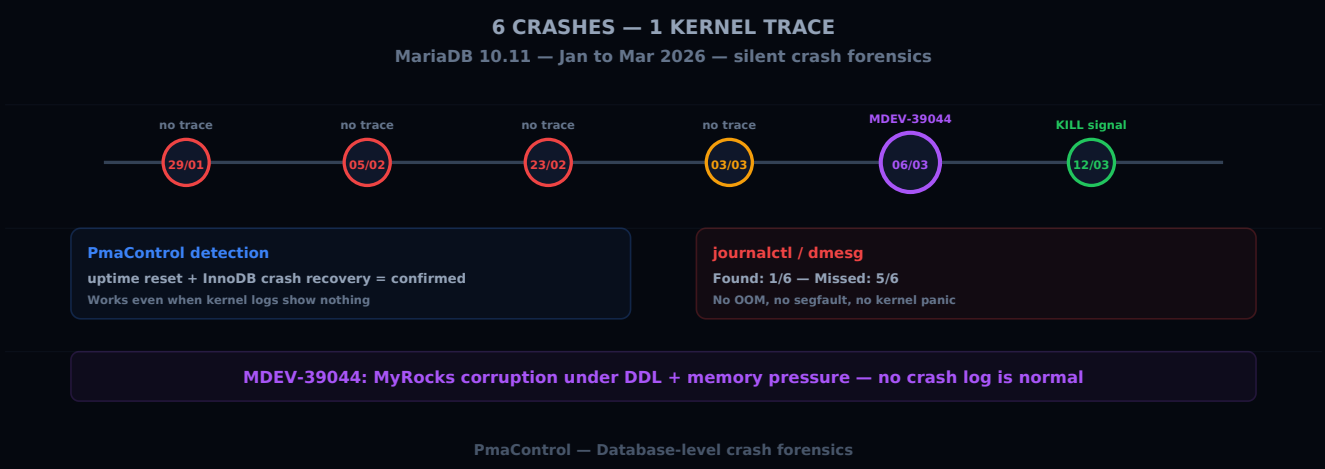
Les Crashs Silencieux de MariaDB : Quand l'Absence de Logs Cache l'Urgence

Aurélien LEQUOY · 12 mars 2026

MARIADB CRASH-ANALYSIS FORENSICS INCIDENT-RESPONSE ROCKSDB



Le problème

Entre janvier et mars 2026, nous avons observé **6 resets d'uptime anormaux** sur un serveur MariaDB 10.11.15 de production supervisé par PmaControl. Le serveur héberge des tables RocksDB partitionnées très volumineuses (métriques time-series).

Le réflexe classique d'un DBA face à un crash : regarder les logs système. `journalctl`, `dmesg`, `/var/log/syslog`. On cherche `OOM`, `Killed process`, `segfault`, `kernel panic`.

Sur 6 crashs, un seul avait une trace kernel exploitable. Les 5 autres : silence complet côté système.

Méthode de détection

Plutôt que de faire confiance aux logs système, nous avons utilisé PmaControl pour détecter les crashs via la **série temporelle** `uptime` :

1. Récupération des valeurs `uptime` via `ts_value_general_int`
2. Filtrage des resets anormaux (uptime qui retombe à 0)

3. Corrélation avec les logs MariaDB (`error.log`)
4. Corrélation avec `journalctl` pour chercher des signatures kernel
5. Analyse des métriques sur l'heure précédente (threads, CPU, mémoire)

C'est l'approche la plus fiable : **un reset d'uptime + une signature `InnoDB crash recovery` = crash confirmé**, même sans trace système.

Les 6 crashs identifiés

Date	Classification	Signature principale
29 janv.	crash probable	`InnoDB crash recovery` + recovery binlog
5 fév.	crash probable	crash recovery + `Event invalid` dans le binlog
23 fév.	crash probable	`InnoDB crash recovery` + `Crash table recovery`
3 mars	crash probable	`Too many connections` puis crash recovery
6 mars	incident majeur	**Corruption MyRocks** : `truncated record body` , `.frm mismatch`
12 mars	crash confirmé	`systemd: status=9/KILL` + crash recovery

Le crash du 6 mars : corrélation MDEV-39044

L'incident le plus sévère est celui du 6 mars. Les erreurs étaient différentes :

```
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
```

Ce pattern correspond exactement au ticket MariaDB **MDEV-39044** : corruption MyRocks déclenchée par :

- des `ALTER TABLE` sur des tables RocksDB partitionnées volumineuses
- une charge d'écriture continue très forte
- une pression mémoire InnoDB simultanée

Le ticket confirme explicitement que **l'absence de crash log ou d'OOM killer est normale dans ce scénario**.

Pourquoi les logs système ne suffisent pas

Sur les 6 incidents, `journalctl` n'a trouvé qu'**une seule trace exploitable** (le `status=9/KILL` du 12 mars).

Pour les 5 autres :

- pas de `Out of memory`
- pas de `Killed process`
- pas de `segfault`
- pas de `kernel panic`

L'inférence est simple : **l'absence de signature kernel n'innocente pas un crash**. C'est même cohérent avec le pattern MDEV-39044, qui documente des crashs sans trace système.

Ce que PmaControl détecte que les logs ne montrent pas

PmaControl surveille `uptime` en continu (toutes les 10 secondes). Un reset = alerte immédiate. Ensuite l'agent corrèle automatiquement :

- les métriques de l'heure précédente (threads, mémoire, CPU)
- la présence de `crash recovery` dans l'error log
- les erreurs de métadonnées (`.frm mismatch`)

Ce qui permet de classifier l'incident **même sans coopération du noyau**.

Recommandations

1. **Ne jamais se fier uniquement aux logs système** pour détecter les crashs MariaDB
2. **Monitorer `uptime` comme indicateur primaire** de stabilité
3. **Corréler avec l'error log MariaDB**, pas avec `journalctl`
4. **Si vous utilisez RocksDB** : limiter les DDL sur les tables partitionnées volumineuses, surtout sous charge d'écriture

5. **Suivre MDEV-39044** pour un éventuel correctif MyRocks

Conclusion

Un serveur MariaDB peut crasher **6 fois en 6 semaines** sans qu'aucun log système ne le documente. Seule une supervision dédiée aux bases de données — qui comprend les signatures internes de MariaDB — permet de détecter et classifier ces incidents.

C'est exactement le rôle de PmaControl.