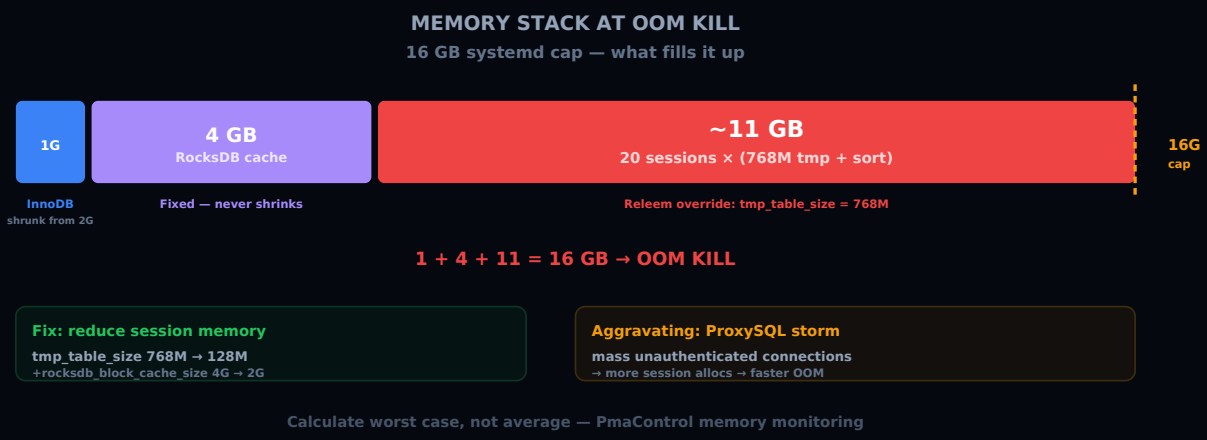
OOM Tueuse : Comment 768 MB de Session ont Noyé 16 GB de RAM

Aurélien LEQUOY · 2 avril 2026

MARIADB OOM-KILLER PERFORMANCE-TUNING SYSTEMD MEMORY-MANAGEMENT



Résumé exécutif

MariaDB ne s'arrête pas à cause d'une corruption, d'un problème Galera ou d'un bug SQL. Le kernel Linux **tue le processus** `mariadbd` pour dépassement mémoire.

La preuve est explicite dans systemd et le kernel log :

```
mariadb.service: Failed with result 'oom-kill'
Out of memory: Killed process 1177 (mariadbd) total-vm:22267612kB, anon-rss:16649820kB
Memory cgroup out of memory: Killed process 1146610 (mariadbd)
```

L'environnement

Composant	Valeur
RAM totale	19.5 GB
Swap	~1 GB
systemd MemoryMax	**16 GB**

Composant	Valeur
innodb_buffer_pool_size	2 GB (auto-shrink → 1 GB)
rocksdb_block_cache_size	**4 GB**
tmp_table_size (Releem override)	**768 MB**
max_heap_table_size (Releem override)	**768 MB**
sort_buffer_size	32 MB
max_connections	100

Ce qui se passe avant le kill

MariaDB détecte la pression mémoire et tente de se protéger en réduisant le buffer pool InnoDB :

```
Memory pressure event shrunk innodb_buffer_pool_size=1536m from 2048m
→ 1280m → 1152m → 1088m → 1056m → 1040m → 1032m → 1024m
Memory pressure event disregarded; innodb_buffer_pool_size=1024m,
  innodb_buffer_pool_size_auto_min=1024m
```

InnoDB a déjà réduit son buffer pool au minimum (1 GB). Mais ça ne suffit pas. Les autres consommateurs ne reculent pas.

Le calcul du pire cas

Avec 100 connexions simultanées, le pire cas de consommation mémoire par session :

```
100 × (768 MB tmp_table + 768 MB heap + 32 MB sort) = ~153 GB
```

Évidemment, toutes les sessions ne créent pas des tables temporaires de 768 MB. Mais il suffit de **20 sessions** faisant des requêtes avec `GROUP BY` ou `ORDER BY` sur de gros jeux de données pour exploser les 16 GB :

```
InnoDB buffer pool :   1 GB (shrunk)
RocksDB cache :        4 GB (fixe, ne recule pas)
20 sessions × 768 MB : 15 GB
Total :                20 GB → kill
```

Le facteur aggravant : connection storm ProxySQL

Juste avant l'OOM, le log MariaDB montre des connexions avortées en masse depuis `10.68.68.103` (ProxySQL) :

```
Aborted connection ... user: 'unauthenticated' host: '10.68.68.103'
Too many connections
```

Plus de connexions = plus de mémoire de session = plus de pression.

Le correctif

Actions immédiates

1. **Réduire la mémoire de session** :

```
tmp_table_size = 128M
max_heap_table_size = 128M
sort_buffer_size = 8M
```

2. **Remonter le cap systemd** :

```
MemoryMax=18G
```

3. **Auditer le cache RocksDB** — 4 GB est peut-être surdimensionné :

```
rocksdb_block_cache_size = 2G
```

Actions moyen terme

- Supprimer le fichier Releem qui override les valeurs (`/etc/mysql/releem.conf.d/z_aiops_mysql.cnf`)
- Monitorer `memory_mysqld` via PmaControl pour alerter avant le kill
- Configurer ProxySQL avec un `max_connections` côté backend plus bas que le `max_connections` MariaDB

Ce que ce n'est pas

Ce n'est **pas** :

- un échec de démarrage
- une recovery Galera cassée
- un datadir corrompu
- un problème de file descriptors

MariaDB a redémarré proprement et est revenu `active (running)` immédiatement.

Conclusion

Un outil de tuning automatique (Releem) a poussé `tmp_table_size` à 768 MB — une valeur qui semble raisonnable isolément. Mais combinée avec un cap systemd à 16 GB, un cache RocksDB de 4 GB et des connection storms ProxySQL, elle devient une bombe à retardement.

La mémoire d'un serveur MariaDB, ça se calcule en pire cas, pas en cas moyen.