

OOM Killer: jak 768 MB ustawień sesji zatopilo 16 GB RAM

Aurélien LEQUOY · April 2, 2026

MARIADB

OOM-KILLER

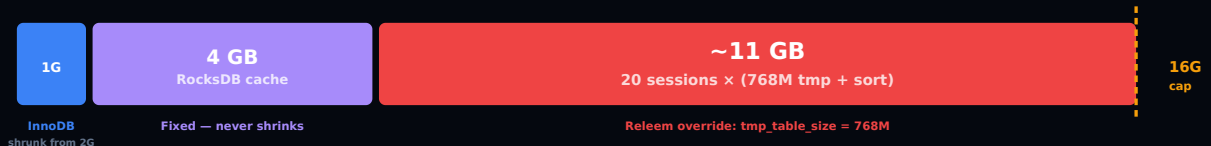
PERFORMANCE-TUNING

SYSTEMD

MEMORY-MANAGEMENT

MEMORY STACK AT OOM KILL

16 GB systemd cap — what fills it up



1 + 4 + 11 = 16 GB → OOM KILL

Fix: reduce session memory

tmp_table_size 768M → 128M
+rocksdb_block_cache_size 4G → 2G

Aggravating: ProxySQL storm

mass unauthenticated connections
→ more session allocs → faster OOM

Calculate worst case, not average — PmaControl memory monitoring

Streszczenie wykonawcze

MariaDB nie zatrzymuje się z powodu korupcji danych, problemu Galera ani bugu SQL. Kernel Linux **zabija proces** `mariadb` za przekroczenie pamięci.

Dowód jest jednoznaczny w logach systemd i kernela:

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

Środowisko

| Komponent | Wartość |
|-------------------|--------------|
| RAM całkowita | 19.5 GB |
| Swap | ~1 GB |
| systemd MemoryMax | 16 GB |

| Komponent | Wartość |
|---------------------------------------|---------------------------|
| 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 |

Co dzieje się przed zabiciem procesu

MariaDB wykrywa presję pamięciową i próbuje się chronić, zmniejszając 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 zmniejszył już swój buffer pool do minimum (1 GB). Ale to nie wystarczy. Pozostali konsumenci pamięci nie ustępują.

Obliczenie najgorszego przypadku

Przy 100 jednoczesnych połączeniach najgorszy przypadek zużycia pamięci na sesję:

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

Oczywiście nie wszystkie sesje tworzą tabele tymczasowe o rozmiarze 768 MB. Ale wystarczy **20 sesji** wykonujących zapytania z `GROUP BY` lub `ORDER BY` na dużych zestawach danych, by przekroczyć 16 GB:

```
InnoDB buffer pool : 1 GB (zmniejszony)
RocksDB cache : 4 GB (stałe, nie cofa się)
20 sesji × 768 MB : 15 GB
Razem : 20 GB → kill
```

Czynnik obciążający: burza połączeń ProxySQL

Tuż przed OOM log MariaDB pokazuje masowe przerywane połączenia z 10.68.68.103

(ProxySQL):

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

Więcej połączeń = więcej pamięci sesji = więcej presji.

Poprawka

Natychmiastowe działania

1. Zmniejszenie pamięci sesji:

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

2. Podniesienie limitu systemd:

```
MemoryMax=18G
```

3. Audyt cache RocksDB — 4 GB jest prawdopodobnie przewymiarowane:

```
rocksdb_block_cache_size = 2G
```

Działania średnioterminowe

- Usunięcie pliku Releem nadpisującego wartości (/etc/mysql/releem.conf.d/z_aiops_mysql.cnf)
- Monitoring `memory_mysql_d` przez PmaControl w celu alarmowania przed zabiciem procesu
- Konfiguracja ProxySQL z niższym `max_connections` po stronie backendu niż `max_connections` MariaDB

Czym to nie jest

To **nie jest**:

- błąd uruchamiania
- uszkodzone odzyskiwanie Galera
- uszkodzony datadir
- problem z deskryptorami plików

MariaDB zrestartował się prawidłowo i natychmiast wrócił do stanu `active (running)`.

Podsumowanie

Narzędzie automatycznego tuningu (Releem) ustawiło `tmp_table_size` na 768 MB — wartość, która w izolacji wygląda rozsądnie. Ale w połączeniu z limitem `systemd` 16 GB, cache RocksDB 4 GB i burzami połączeń ProxySQL staje się bombą zegarową.

Pamięć serwera MariaDB oblicza się dla najgorszego przypadku, nie dla przypadku średniego.