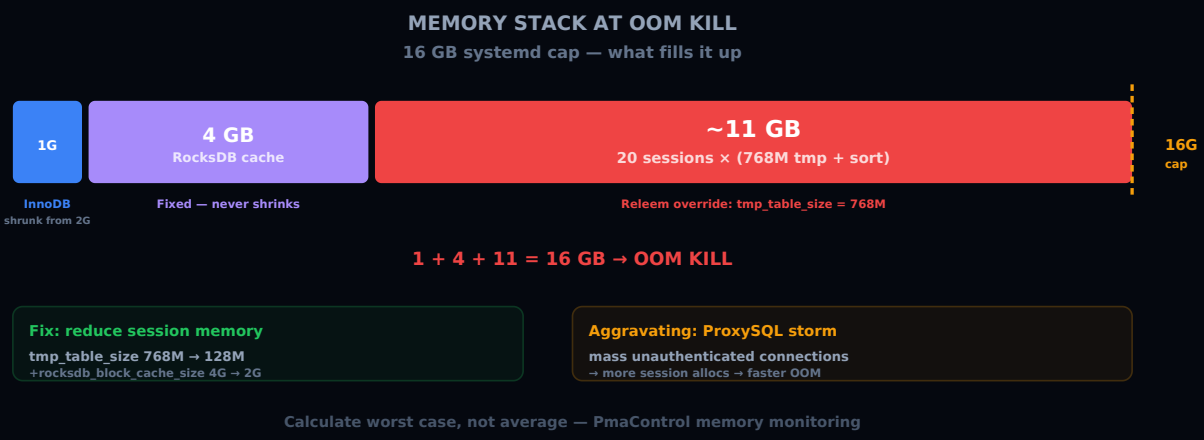
The Silent OOM Killer: How 768MB Session Settings Drowned 16GB of Memory

Aurélien LEQUOY · April 2, 2026

MARIADB OOM-KILLER PERFORMANCE-TUNING SYSTEMD MEMORY-MANAGEMENT



Executive summary

MariaDB does not stop because of corruption, a Galera issue, or an SQL bug. The Linux kernel **kills the `mariadbd` process** for exceeding memory limits.

The evidence is explicit in systemd and the 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)
```

The environment

Component	Value
Total RAM	19.5 GB
Swap	~1 GB
systemd MemoryMax	**16 GB**

Component	Value
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

What happens before the kill

MariaDB detects memory pressure and tries to protect itself by shrinking the InnoDB buffer pool:

```
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 has already reduced its buffer pool to the minimum (1 GB). But it's not enough. The other memory consumers don't back down.

Worst-case calculation

With 100 simultaneous connections, the worst-case per-session memory consumption:

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

Obviously, not all sessions create 768 MB temporary tables. But just **20 sessions** running queries with `GROUP BY` or `ORDER BY` on large datasets are enough to blow through the 16 GB cap:

```
InnoDB buffer pool:    1 GB (shrunk)
RocksDB cache:         4 GB (fixed, doesn't shrink)
20 sessions × 768 MB:  15 GB
Total:                 20 GB → kill
```

Aggravating factor: ProxySQL connection storm

Just before the OOM, the MariaDB log shows mass aborted connections from `10.68.68.103` (ProxySQL):

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

More connections = more session memory = more pressure.

The fix

Immediate actions

1. **Reduce session memory**:

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

2. **Raise the systemd cap**:

```
MemoryMax=18G
```

3. **Audit the RocksDB cache** — 4 GB may be oversized:

```
rocksdb_block_cache_size = 2G
```

Medium-term actions

- Remove the Releem override file (`/etc/mysql/releem.conf.d/z_aiops_mysql.cnf`)
- Monitor `memory_mysqld` via PmaControl to alert before the kill
- Configure ProxySQL with a lower backend `max_connections` than MariaDB's `max_connections`

What it is not

It is **not**:

- a startup failure

- a broken Galera recovery
- a corrupted datadir
- a file descriptor issue

MariaDB restarted cleanly and came back `active (running)` immediately.

Conclusion

An automatic tuning tool (Releem) pushed `tmp_table_size` to 768 MB — a value that seems reasonable in isolation. But combined with a 16 GB systemd cap, a 4 GB RocksDB cache, and ProxySQL connection storms, it becomes a ticking time bomb.

A MariaDB server's memory must be calculated for worst case, not average case.