The **hidden life** of internal temporary tables

Unraveling MySQL/MariaDB Internal Management

Gabriel Ciciliani
Releem

Agenda

1. Intro
2. Temporary tables mechanics
3. Telemetry
4. Dealing with temporary tables
5. Lessons and takeaways

About me..

- DB journey started in 2006, as a Jr Oracle DBA
- Fully into OSS DBs around 2014
    - *DBRE, Consulting, MySQL development*

- Started building **Tuning Wizard** in Oct 2022
    - *Automatic query optimization REST API*

- Acquired by **Releem** last year
    - *Helping Releem to take automatic server performance optimization to the next level*
- I'm from Buenos Aires, Argentina

Kinds of Temporary tables in MySQL / MariaDB

- User-defined temporary tables
- Temporary “*shadow*” tables when doing schema changes
- Internal temporary tables (ITT) ⟵ **You are here!**

Internal temporary tables are used as **auxiliary space** to store query rows temporarily in order to **compute** the final **resultset**


SO WHY SHOULD WE
CARE?
makeameme.org

Performance!

(more on this later..)

Most common scenarios where ITT are created

#

Queries with different ORDER BY and GROUP BY clauses

#

When a subquery needs to be materialized

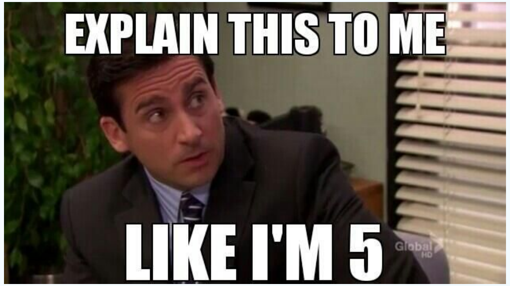
#

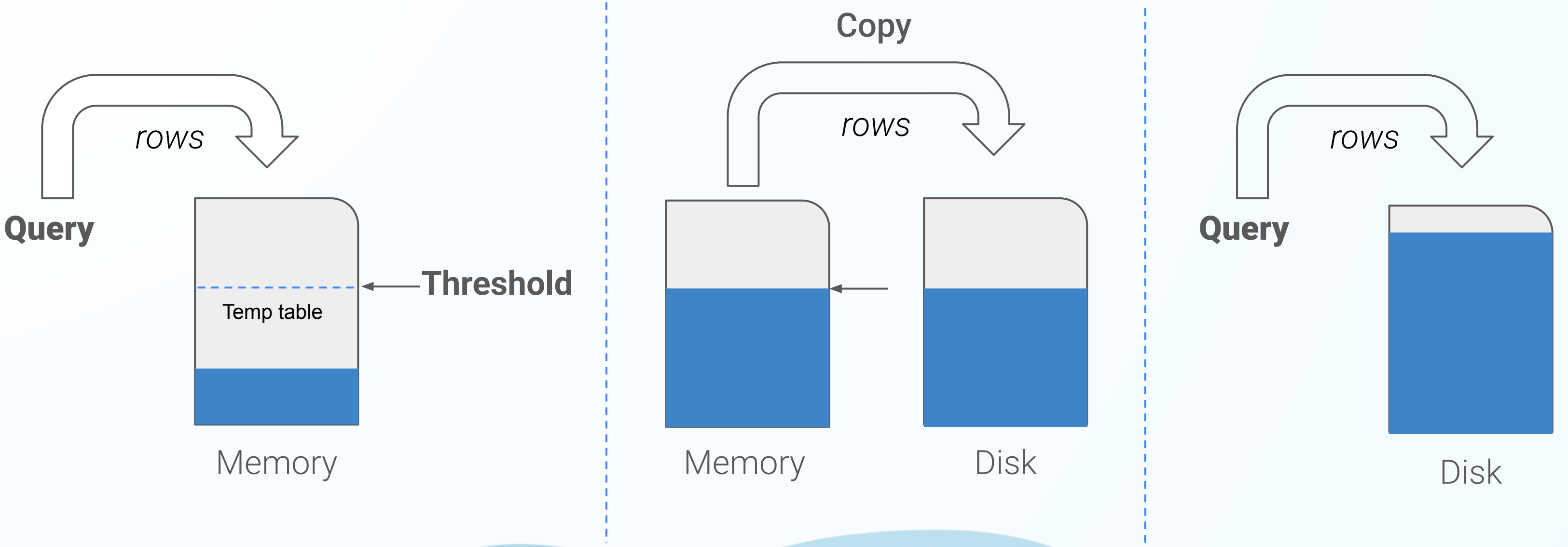
Multi-table UPDATEs

#

INSERT ... SELECT statements that select from and insert into the same table

IIT mechanics



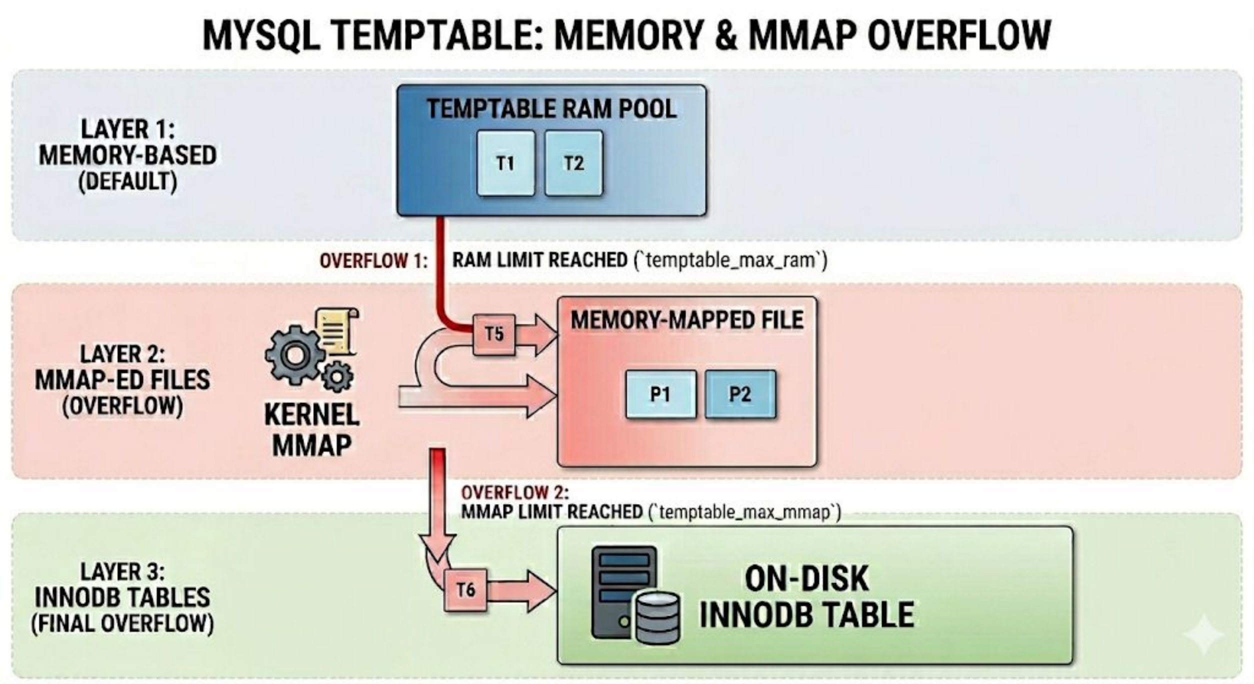


IIT and storage engines

- Historically..
  - The **MEMORY** storage engine was used for the in-memory stage
  - **MyISAM** was the storage engine used when the table as moved on-disk

- With **MySQL v5.7**, **Innodb** was made the default on-disk engine
- With **MySQL v8.0**, the **TempTable** engine for temporary tables was introduced.
- **MariaDB** is still using **MEMORY** for the on-memory phase and **Aria** for when the temporary table is spilled to disk

TempTable: one additional layer



MySQL parameters governing IITs

5.7

- `internal_tmp_disk_storage_engine`
- `tmp_table_size`
- `max_heap_table_size`

8.0 and above

- `tmp_table_size`
- `temptable_max_ram`
- `temptable_max_mmap`
- `temptable_use_mmap` ← **deprecated!**
- `internal_tmp_mem_storage_engine`

MariaDB parameters governing IITs

- `aria_used_for_temp_tables`
- `tmp_table_size`
- `max_heap_table_size`

ITTs will go directly to disk..

TempTable engine

- On **UNION** or UNION ALL queries, when string columns with a maximum length **larger than 512 bytes /characters** are present in the **SELECT** list

MEMORY engine

- Presence of **BLOB** or **TEXT** Columns
- If (`row-size * row-count`) **>** `min(tmp_table_size, max_heap_table_size)`
- When combined unique key **> 512 bytes** (DISTINCT, GROUP BYs)

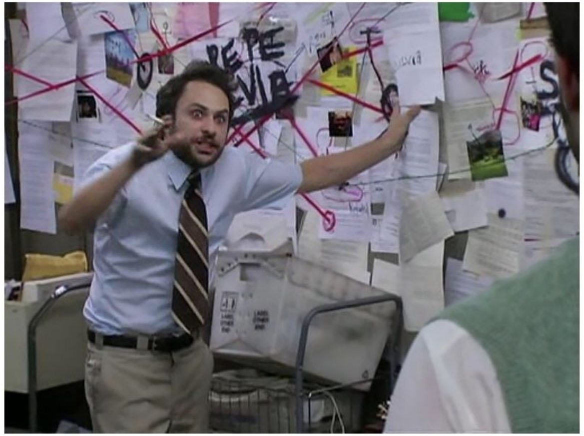
Temptable vs Memory

Feature	TempTable Engine	MEMORY Engine
Row Format	Variable-length *(stores only what is needed)*	Fixed-length *(always uses max column width)*
BLOB/TEXT Support	Yes *(supported in-memory)*	No *(forces immediate move to disk)*
Efficiency	High *(lower RAM footprint for VARCHAR)*	Low *(wastes RAM on empty VARCHAR space)*
Overflow Behavior	Can overflow to MMAP files before hitting disk	Spills directly to disk *(InnoDB/Aria)*

Telemetry

Status Variables

- `Created_tmp_disk_tables`
- `Created_tmp_tables`
- `TempTable_count_hit_max_ram` (MySQL 9)



Telemetry

Performance_schema (MySQL only)

- `memory/temptable/physical_disk (mmap activity)`
- `memory/temptable/physical_ram`
- MySQL Bug #98741: `Created_tmp_disk_tables` does not increment when the TempTable engine uses memory-mapped files

How to know if my Query is using a temporary table?

MySQL EXPLAIN FORMAT = JSON

```
"grouping_operation": {
  "using_temporary_table": true
```

```
"duplicates_removal": {
  "using_temporary_table": true
```

```
"ordering_operation": {
  "using_temporary_table": true
```

```
"materialized_from_subquery": {
  "using_temporary_table": true,
```

How to know if my Query is using a temporary table

MariaDB

```
"filesort": {
   "sort_key": "booking.flight_id, booking.seat",
   "temporary_table": {
```

How to confirm a query is using an on-disk temp table? *(MySQL only)*

```
> SET OPTIMIZER_TRACE="enabled=on";

> SET OPTIMIZER_TRACE_MAX_MEM_SIZE=1000000;

> (run Query)


> SELECT
     TRACE,
     MISSING_BYTES_BEYOND_MAX_MEM_SIZE
  FROM
     INFORMATION_SCHEMA.OPTIMIZER_TRACE\G
```

How to confirm a query is using an on-disk temp table? *(MySQL only)*

```
{
  "converting_tmp_table_to_ondisk": {
    "cause": "memory_table_size_exceeded",
    "tmp_table_info": {
      "table": "<temporary>",
      "in_plan_at_position": 1,
      "columns": 3,
      "row_length": 30,
      "key_length": 21,
      "unique_constraint": false,
      "makes_grouped_rows": true,
      "cannot_insert_duplicates": false,
      "location": "disk (InnoDB)",
      "record_format": "fixed"
    }
  }
}
```

Can we estimate how the size of tmp_table_size?

With MEMORY, calculation is more straightforward

- Fixed-length storage
- Lowest value between `tmp_table_size` and `max_heap_table_size`
- No minimum value
- Basically, `row-count x (coltype_size_1 + coltype_size_2 … coltype_size_n)`

Can we estimate how the size of tmp_table_size?

With TempTable more variables come into play

- This storage engine uses a more efficient row format for variable-length columns (VARCHAR/VARBINARY/BLOB) where only the required space is allocated
- The TempTable engine has the three-tiers storage strategy
- For `tmp_table_size` values lower than the minimum allocation size, the sort buffer is used instead

Lesson #1: Avoid them!

- UNION vs UNION ALL
- Indexes that cover GROUP BYs (and ORDER BYs if combined)

An index on columns (A,B,C) could be leveraged for sorting in the following scenarios:

- `WHERE A = value GROUP BY B`
- `WHERE A = value GROUP BY B,C`
- `WHERE A = value AND B = value GROUP BY C`
- `WHERE A = value AND B = value AND C > value GROUP BY C`

Lesson #2: can't avoid them? keep them in memory

- `tmp_table_size (can be set at the session level)`
- `max_heap_table_size`
- `temptable_max_ram`
- Avoid BLOBs / TEXT (MariaDB)
- Avoid excessively large columns if they are not needed (512 bytes /characters mark)

Questions **time!**

MY LINKEDIN

Gabriel Ciciliani

Releem

Thanks!

Releem