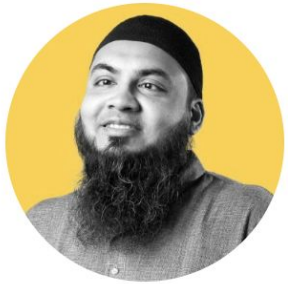
High Performance MySQL

https://unsplash.com/photos/timelapse-on-road-J7EUjSlNQtg



Follow

Mohammad Emran ⚡

@phpfour

Sharing practical tips, resources, and ideas on Laravel, PHP, and Web Development.

Follow along to become a PRO software developer.

#FreePalestine

127.0.0.1 newsletter.phpfour.com Joined October 2007

662 Following **7,273** Followers

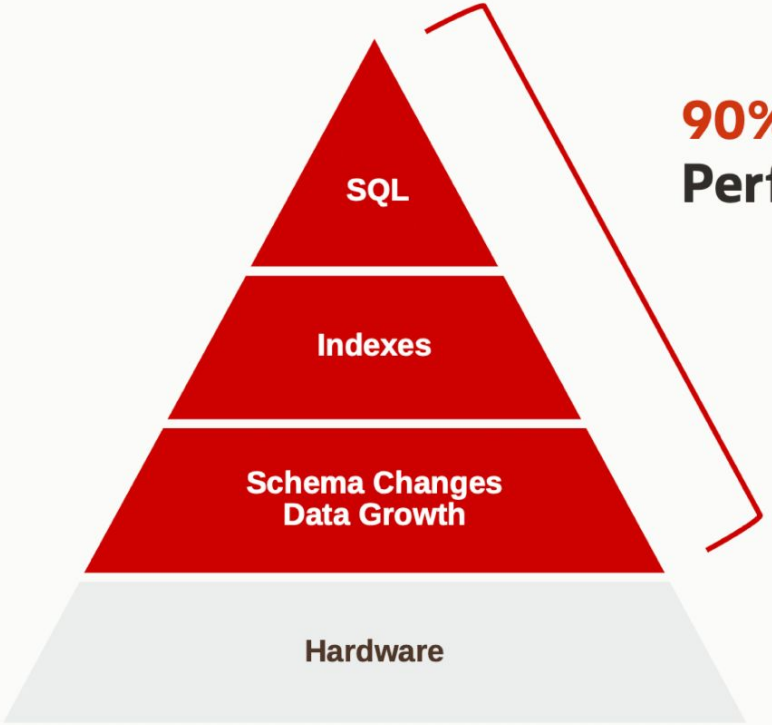
Slow Query Response

Too Many Connections

100% CPU Usage!

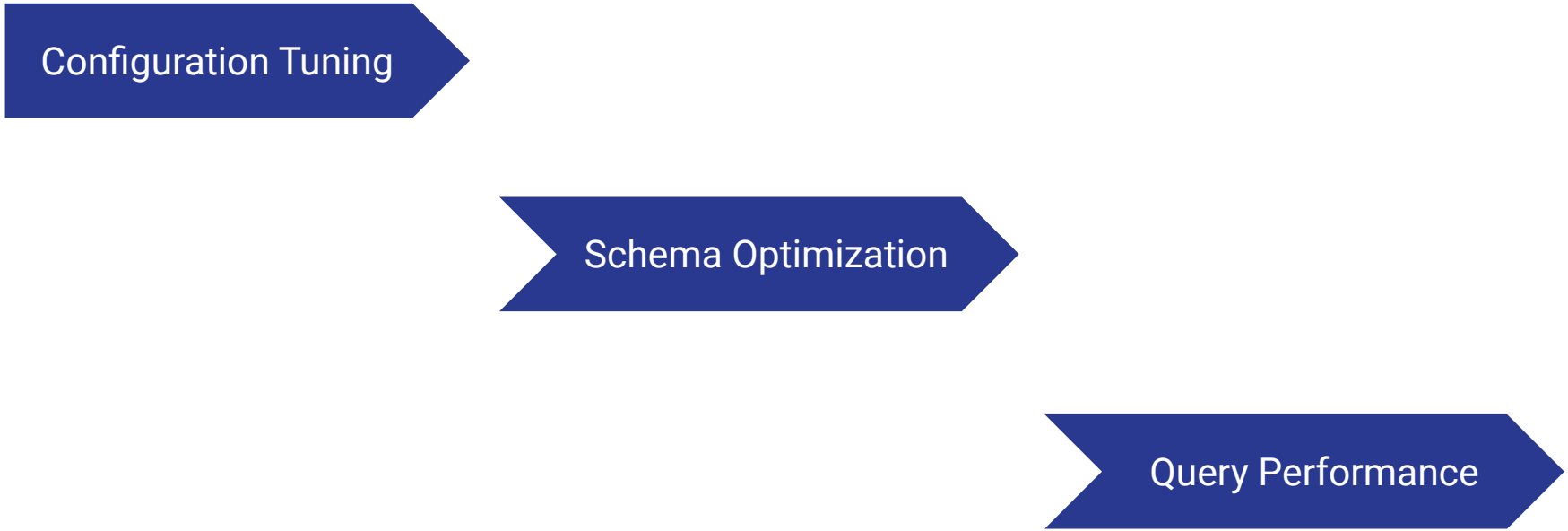
Low Disk Space

Database Non-Responsive



90% of
Performance Problems

Plan of Attack!



Monitoring

Before you start…

- **Monitor** your database performance
- Identify areas of **improvements**
- Make **small changes** at a time
- **Test** your changes before deploying to production

Monitoring

Data Collection

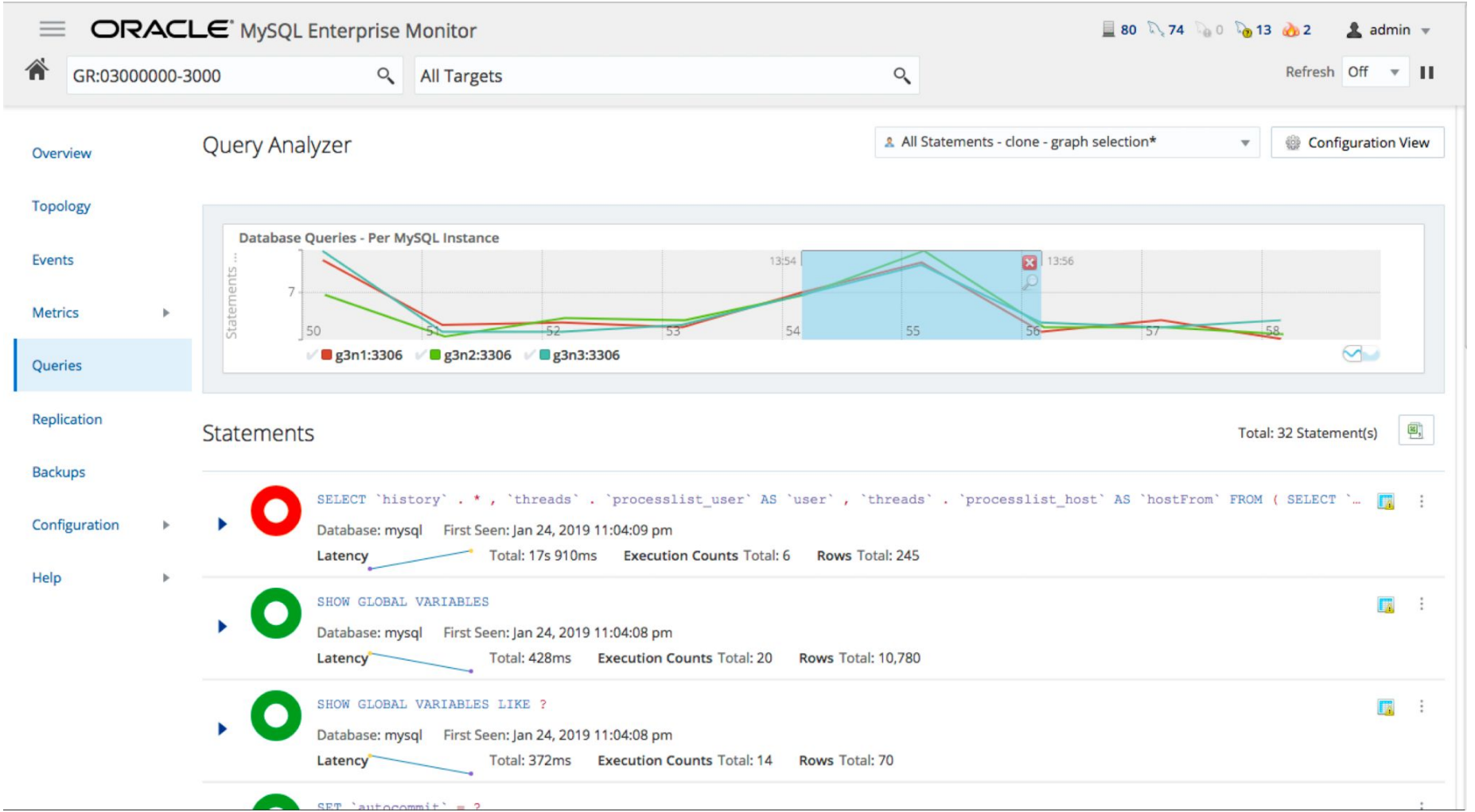
- Slow Query Logs
- Performance Schema
- Application Metrics

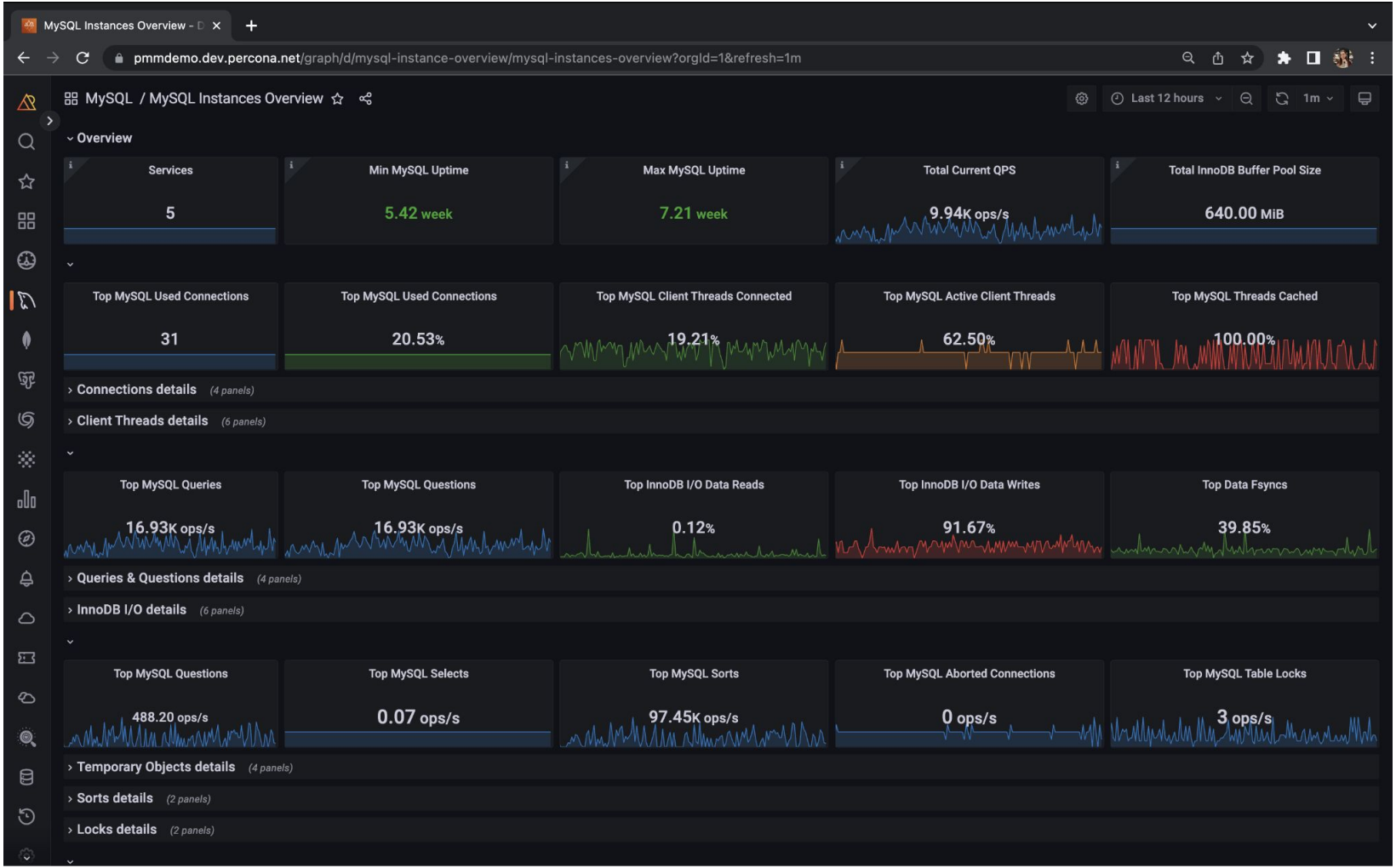
Digging Deep

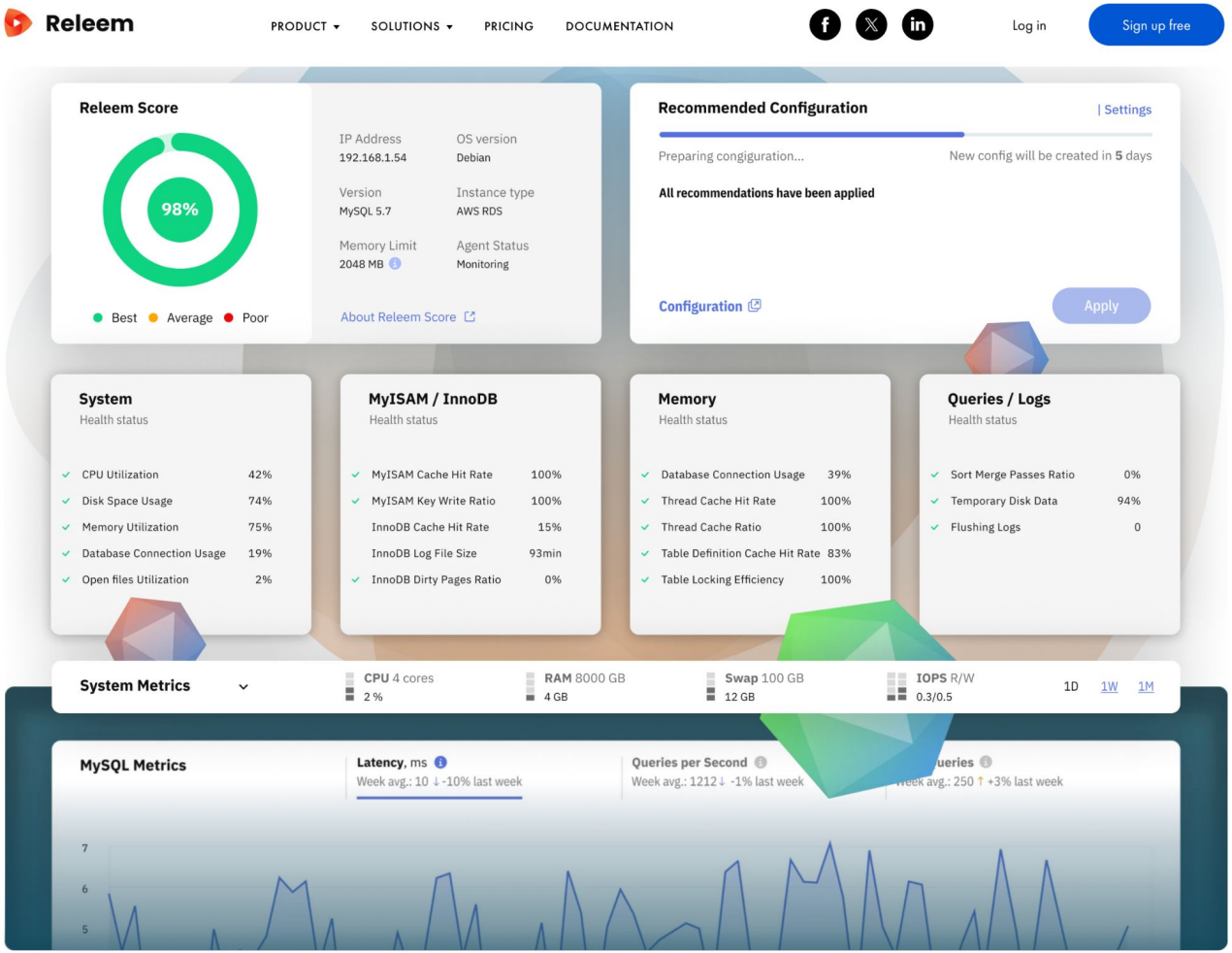
- Manual query execution
- EXPLAIN & ANALYZE
- Network Usage

Tools

- MySQL Enterprise Monitor
- Percona Monitoring and Management (PMM)
- Releem


ORACLE MySQL Enterprise Monitor
80 74 0 13 2
admin
GR:03000000-3000
All Targets
Refresh Off
Overview
Topology
Events
Metrics
Queries
Replication
Backups
Configuration
Help
Query Analyzer
All Statements - clone - graph selection*
Configuration View
Database Queries - Per MySQL Instance
Statements ...
7
50
51
52
53
54
55
56
57
58
13:54
13:56
g3n1:3306
g3n2:3306
g3n3:3306
Statements
Total: 32 Statement(s)
SELECT `history` . * , `threads` . `processlist_user` AS `user` , `threads` . `processlist_host` AS `hostFrom` FROM (SELECT `...
Database: mysql First Seen: Jan 24, 2019 11:04:09 pm
Latency Total: 17s 910ms Execution Counts Total: 6 Rows Total: 245
SHOW GLOBAL VARIABLES
Database: mysql First Seen: Jan 24, 2019 11:04:08 pm
Latency Total: 428ms Execution Counts Total: 20 Rows Total: 10,780
SHOW GLOBAL VARIABLES LIKE ?
Database: mysql First Seen: Jan 24, 2019 11:04:08 pm
Latency Total: 372ms Execution Counts Total: 14 Rows Total: 70


MySQL Instances Overview -
pmmdemo.dev.percona.net/graph/d/mysql-instance-overview/mysql-instances-overview?orgId=1&refresh=1m
MySQL / MySQL Instances Overview
Last 12 hours
1m
Overview
Services
5
Min MySQL Uptime
5.42 week
Max MySQL Uptime
7.21 week
Total Current QPS
9.94K ops/s
Total InnoDB Buffer Pool Size
640.00 MiB
Top MySQL Used Connections
31
Top MySQL Used Connections
20.53%
Top MySQL Client Threads Connected
19.21%
Top MySQL Active Client Threads
62.50%
Top MySQL Threads Cached
100.00%
Connections details (4 panels)
Client Threads details (6 panels)
Top MySQL Queries
16.93K ops/s
Top MySQL Questions
16.93K ops/s
Top InnoDB I/O Data Reads
0.12%
Top InnoDB I/O Data Writes
91.67%
Top Data Fsyncs
39.85%
Queries & Questions details (4 panels)
InnoDB I/O details (6 panels)
Top MySQL Questions
488.20 ops/s
Top MySQL Selects
0.07 ops/s
Top MySQL Sorts
97.45K ops/s
Top MySQL Aborted Connections
0 ops/s
Top MySQL Table Locks
3 ops/s
Temporary Objects details (4 panels)
Sorts details (2 panels)
Locks details (2 panels)


Releem
PRODUCT
SOLUTIONS
PRICING
DOCUMENTATION
Log in
Sign up free
Releem Score
98%
Best
Average
Poor
IP Address
192.168.1.54
OS version
Debian
Version
MySQL 5.7
Instance type
AWS RDS
Memory Limit
2048 MB
Agent Status
Monitoring
About Releem Score
Recommended Configuration
| Settings
Preparing congiguration...
New config will be created in 5 days
All recommendations have been applied
Configuration
Apply
System
Health status
CPU Utilization 42%
Disk Space Usage 74%
Memory Utilization 75%
Database Connection Usage 19%
Open files Utilization 2%
MyISAM / InnoDB
Health status
MyISAM Cache Hit Rate 100%
MyISAM Key Write Ratio 100%
InnoDB Cache Hit Rate 15%
InnoDB Log File Size 93min
InnoDB Dirty Pages Ratio 0%
Memory
Health status
Database Connection Usage 39%
Thread Cache Hit Rate 100%
Thread Cache Ratio 100%
Table Definition Cache Hit Rate 83%
Table Locking Efficiency 100%
Queries / Logs
Health status
Sort Merge Passes Ratio 0%
Temporary Disk Data 94%
Flushing Logs 0
System Metrics
CPU 4 cores
2 %
RAM 8000 GB
4 GB
Swap 100 GB
12 GB
IOPS R/W
0.3/0.5
1D
1W
1M
MySQL Metrics
Latency, ms
Week avg.: 10 ↓ -10% last week
Queries per Second
Week avg.: 1212 ↓ -1% last week
Week avg.: 250 ↑ +3% last week
7
6
5

Configuration Tuning

```
[mysqld]

# GENERAL
datadir  = /var/lib/mysal
socket   = /var/lib/mysql/mysql.sock
pid file = /var/lib/mysql/mysql.pid
user     = mysql
port     = 3306

# INNODB
innodb_buffer_pool_size = <value>
innodb_file_per_table   = 1
innodb_flush_method     = O_DIRECT

# LOGGING
log_error = /var/lib/mysql/mysql-error.log
log_slow_queries = /var/lib/mysql/mysql-slow.log

# OTHER
tmp_table_size      = 32M
max_heap_table_size = 32M
max_connections     = <value>
thread_cache_size   = <value>
table_open_cache    = <value>
open_files_limit    = 65535
```

A sane, minimal configuration for MySQL 8.0 to start with.

InnoDB Buffer Pool Size (innodb_buffer_pool_size)

- The single-most **important** configuration to tweak
- Memory area where InnoDB **caches** table and index data
- Optimal size:

  Total host memory

  (–) memory for OS & other processes

  (–) memory required by MySQL other than the InnoDB buffer pool

Auto Configuration (innodb_dedicated_server)

- Only consider if MySQL is running on a **dedicated** server / VM
- Automatic configuration of many InnoDB configuration
- Override is still possible

Maximum Connections (max_connections)

- Maximum permitted number of simultaneous client connections
- Be careful setting this too large as each connection requires memory
- Affect the maximum number of files the server keeps open

  (set the correct **ulimits** and **file descriptors** settings in Linux servers)

Schema Optimization

Minimize Space Usage

- Tables should be designed for **minimum** space usage
- Smaller tables require **less memory**
- Smaller indexes can be **processed faster**

Choose Optimal Data Types

- **Smaller** is usually better:

  TINYINT vs MEDIUMINT vs INT vs BIGINT

- Pick the **native data type** for their intended purpose

  Don’t store dates as string, store them using built-in types

- **Avoid NULL** (if possible)

  Nullable columns require more space and slows down processing

Primary Keys

- The primary key of a table should be **as short as possible**
- InnoDB duplicates primary key on **secondary indexes**, so it would affect storage space
- Ensure foreign key **matches exactly** with the referring the primary key type, collation and character set

Query Performance Optimization

Slow Query Log

```
-- Enable
SET GLOBAL slow_query_log = 'ON';

-- Keeping Slow query log in file or table
SET GLOBAL slow_query_log_file = '/tmp/slow_queries.log';
SET GLOBAL log_output = 'table';

-- Additional settings
SET GLOBAL log_queries_not_using_indexes = 'ON';
SET GLOBAL long_query_time = 5;

-- Disable
SET GLOBAL slow_query_log = 'OFF';
```

Slow Query Log

```
sudo mysqldumpslow /tmp/slow_queries.log
```

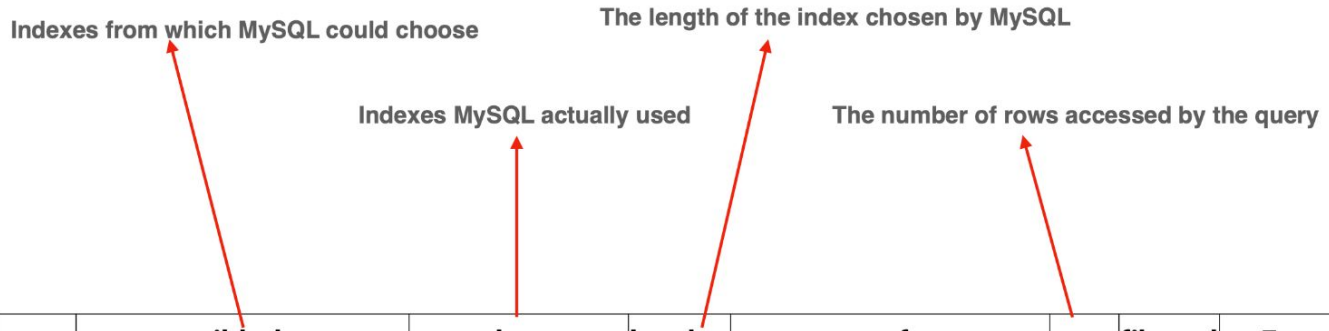
```
Reading mysql slow query log from /var/lib/mysql/mysql-slow.log
Count: 1  Time=2.43s (2s)  Lock=0.00s (0s)  Rows=16.0 (16), user1[user1]@localhost
  select year(e.hire_date), max(s.salary) from employees e join salaries s on e.emp_no=s.emp_no group by N

Count: 1  Time=2.03s (2s)  Lock=0.00s (0s)  Rows=16.0 (16), user1[user1]@localhost
  select year(e.hire_date), max(s.salary) from employees e join salaries s on e.emp_no=s.emp_no group by year(e.hire_date)

Count: 3  Time=0.71s (2s)  Lock=0.00s (0s)  Rows=94709.0 (284127), user1[user1]@localhost
  select * from salaries where salary >= N

Count: 3  Time=0.26s (0s)  Lock=0.00s (0s)  Rows=95012.7 (285038), user1[user1]@localhost
  select * from employees e join salaries s on e.emp_no=s.emp_no where year(e.hire_date) = N

Count: 2  Time=0.18s (0s)  Lock=0.00s (0s)  Rows=24.0 (48), user1[user1]@localhost
  select * from titles where title = 'S'

Count: 1  Time=0.14s (0s)  Lock=0.00s (0s)  Rows=6.0 (6), user1[user1]@localhost
  select * from employees natural join salaries natural join titles where employees.last_name='S'
```

Use EXPLAIN

```
EXPLAIN
SELECT first_name, last_name, city, country
FROM customer
INNER JOIN address USING(address_id)
INNER JOIN city USING(city_id)
INNER JOIN country USING(country_id);
```

id	select_type	table	artition	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	country		ALL	PRIMARY				109	100.00	
1	SIMPLE	city		ref	PRIMARY,idx_fk_country_i	idx_fk_country_ic	2	sakila.country.country_ic	5	100.00	
1	SIMPLE	address		ref	PRIMARY,idx_fk_city_id	idx_fk_city_id	2	sakila.city.city_id	1	100.00	Using index
1	SIMPLE	customer		ref	idx_fk_address_id	idx_fk_address_ic	2	sakila.address.address_i	1	100.00	

Use EXPLAIN



id	select_type	table	partition	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	country		ALL	PRIMARY				109	100.00	
1	SIMPLE	city		ref	PRIMARY,idx_fk_country_i	idx_fk_country_ic	2	sakila.country.country_ic	5	100.00	
1	SIMPLE	address		ref	PRIMARY,idx_fk_city_id	idx_fk_city_id	2	sakila.city.city_id	1	100.00	Using index
1	SIMPLE	customer		ref	idx_fk_address_id	idx_fk_address_ic	2	sakila.address.address_i	1	100.00	

Retrieve Data Selectively

- Only retrieve the columns you need, instead of SELECT *
- Limit the number of rows (if possible)

⚠️
```
SELECT * FROM customer;
```

✅
```
SELECT id, first_name, last_name FROM customer;
```

Use proper INDEX

- Add indexes for columns used in WHERE clause
- Add indexes for columns used in GROUP BY clause
- Add indexes for columns used in ORDER BY clause
- Use composite indexes whenever possible
- Indexing has cost on INSERT/UPDATE, so don’t add too many of them

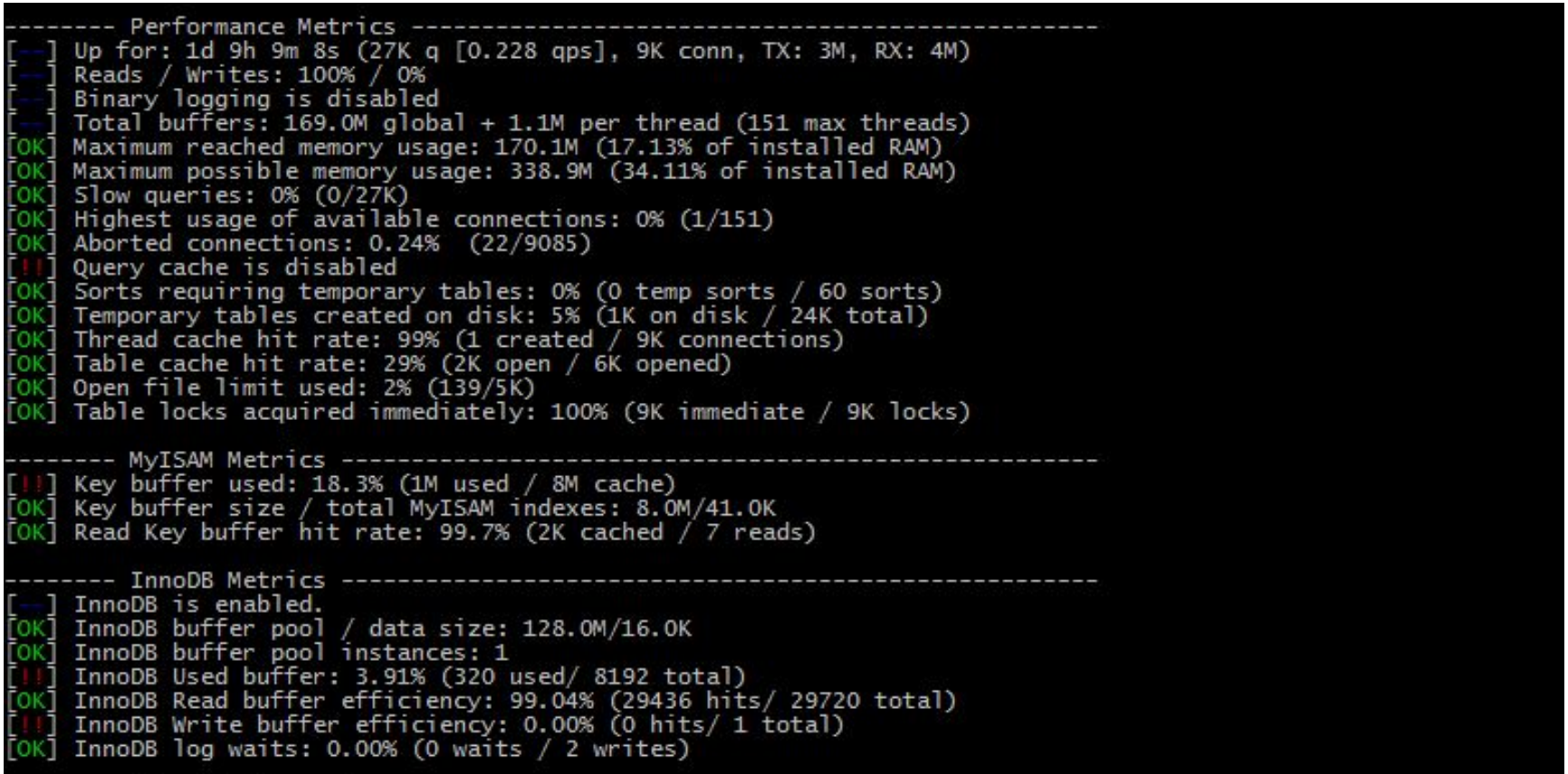
Optimize JOINs

- Use INNER JOIN and LEFT JOIN
- Ensure the columns used in JOINs are indexed
- Use EXPLAIN to verify the right indexes are being used
- The order of joining matters, so start with the smallest result set

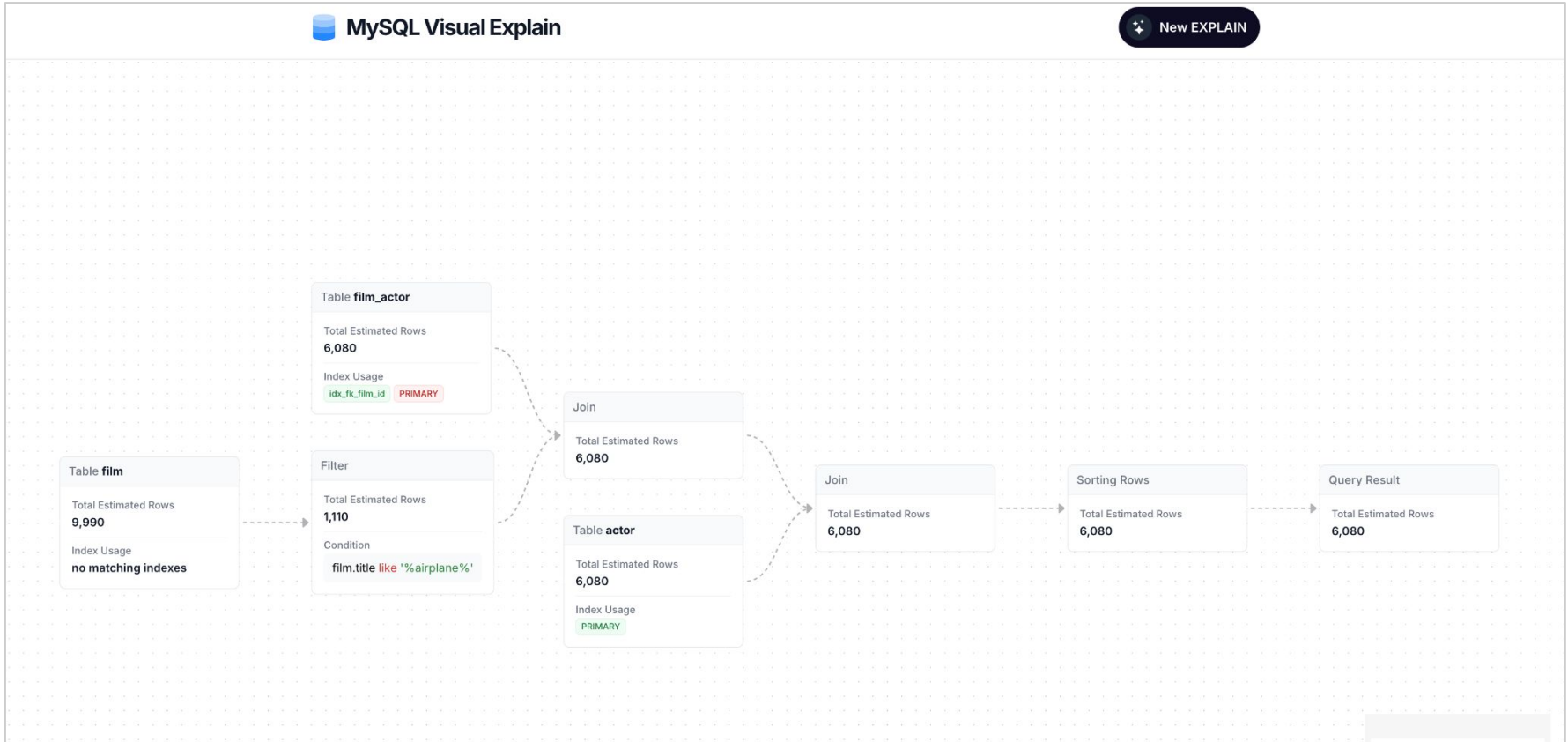
Miscellaneous

- Use **materialized views** for statistical data
- Use **replication/InnoDB Clusters** to split read/write operations
- Use fast enough **SSD or NVMe** disks
- Use the most optimal **storage engine** for the task
- Avoid **long-running** transactions

MySQL Tuner


-------- Performance Metrics ---
[--] Up for: 1d 9h 9m 8s (27K q [0.228 qps], 9K conn, TX: 3M, RX: 4M)
[--] Reads / Writes: 100% / 0%
[--] Binary logging is disabled
[--] Total buffers: 169.0M global + 1.1M per thread (151 max threads)
[OK] Maximum reached memory usage: 170.1M (17.13% of installed RAM)
[OK] Maximum possible memory usage: 338.9M (34.11% of installed RAM)
[OK] Slow queries: 0% (0/27K)
[OK] Highest usage of available connections: 0% (1/151)
[OK] Aborted connections: 0.24% (22/9085)
[!!] Query cache is disabled
[OK] Sorts requiring temporary tables: 0% (0 temp sorts / 60 sorts)
[OK] Temporary tables created on disk: 5% (1K on disk / 24K total)
[OK] Thread cache hit rate: 99% (1 created / 9K connections)
[OK] Table cache hit rate: 29% (2K open / 6K opened)
[OK] Open file limit used: 2% (139/5K)
[OK] Table locks acquired immediately: 100% (9K immediate / 9K locks)
-------- MyISAM Metrics ---
[!!] Key buffer used: 18.3% (1M used / 8M cache)
[OK] Key buffer size / total MyISAM indexes: 8.0M/41.0K
[OK] Read Key buffer hit rate: 99.7% (2K cached / 7 reads)
-------- InnoDB Metrics ---
[--] InnoDB is enabled.
[OK] InnoDB buffer pool / data size: 128.0M/16.0K
[OK] InnoDB buffer pool instances: 1
[!!] InnoDB Used buffer: 3.91% (320 used/ 8192 total)
[OK] InnoDB Read buffer efficiency: 99.04% (29436 hits/ 29720 total)
[!!] InnoDB Write buffer efficiency: 0.00% (0 hits/ 1 total)
[OK] InnoDB log waits: 0.00% (0 waits / 2 writes)

MySQL Visual Explain



References

- MySQL Tuner:
  https://github.com/major/MySQLTuner-perl
- MySQL Visual Explain:
  https://mysqlexplain.com/
- Releem:
  https://releem.com/
- Percona Toolkit:
  https://www.percona.com/software/database-tools/percona-toolkit
- Percona Monitoring & Management:
  https://www.percona.com/software/database-tools/percona-monitoring-and-management/mysql-monitoring

Thank you!