

ProxySQL 2.7 → 3.0PmaControl 4

Aurélien LEQUOY · 2026927

- PMACONTROL
- PROXYSQL
- BENCHMARK
- MARIADB
- MYSQL
- OBSERVABILITY

PmaControl

BENCHMARK NOTES

ProxySQL 2.7 → 3.0

Same complete cache audit. Same two Admin queries.

ProxySQL 2.7.3

8.112ms/audit

ProxySQL 3.0.5

2.015ms/audit

-75.2%

mean elapsed time

4.03× faster

150 timed audits per version · Ubuntu 24.04.4 · 2 vCPU · 4 GiB · PHP 8.4.26

Client-observed Admin workload: SQL + loopback + fetch + PHP. Query-routing throughput is outside scope.

PmaControl 8.1122042.01476075.16%4.03×

AdminPmaControl ProxySQL

	ProxySQL 2.7.3	ProxySQL 3.0.5	
, ms/audit	8.112204	2.014760	75.16 %
, ms/audit	7.895812	2.053810	73.99 %
, ms/audit	11.914402	2.919234	75.50 %
PHP MiB	4	4	

ms/audit150

4 MiB PHP PHP ProxySQL

ProxySQL 2.7 → 3.0

A nice improvement during PmaControl cache-audit benchmarks.

-75.2%

MEAN ELAPSED TIME PER AUDIT

4.03x

faster complete cache audits

Mean time · complete cache audit

ms/audit · lower is better

ProxySQL 2.7.3



8.112

ProxySQL 3.0.5



2.015

0 2 4 6 8 10

Supporting measurements

2.7.3

3.0.5

UNIT

Median batch mean

7.896

2.054

ms/audit

Maximum batch mean

11.914

2.919

ms/audit

Median / maximum across 15 batch means; each batch averages 10 audits.

What “ms/audit” measures

One complete cache audit = two read-only ProxySQL Admin queries.

01 SQL execution

02 Loopback + fetch

03 PHP audit logic

Client elapsed time. Connection setup, process startup, HTTP and UI rendering are outside the timer.

The benchmark setup

Ubuntu 24.04.4 · 2 vCPU · 4 GiB RAM · PHP 8.4.26

150 timed complete audits per version

15 batches × 10 audits · 3 warmups per batch

Same audit code and SQL · 1 + 7 rows returned per audit

PHP allocated peak: 4 MiB for both versions

Scope: this cache-audit workload on a shared test VM.

These are client-observed Admin timings. Query-routing throughput and standalone server execution time were outside this benchmark.

Source: PmaControl PR #6389 · fixed complete audit on both versions

Builds: 2.7.3-12-g50b7f85 / 3.0.5-60-g7e9e009

Nice improvement, ProxySQL.

“ms/audit”

```
QueryCacheEffectiveness::run() SELECT
```

MySQL PHP 本地 loopback PmaControl PHP HTTP

```
SELECT r.rule_id, r.cache_ttl, r.digest,
       r.match_digest, r.match_pattern,
       COALESCE(h.hits, 0) AS hits
FROM runtime_mysql_query_rules r
LEFT JOIN stats_mysql_query_rules h
      ON h.rule_id = r.rule_id
WHERE r.active = 1 AND r.cache_ttl > 0
ORDER BY r.rule_id;
```

```
SELECT Variable_Name AS name, Variable_Value AS value
FROM stats_mysql_global
WHERE Variable_Name IN (
    'Query_Cache_count_GET', 'Query_Cache_count_GET_OK',
    'Query_Cache_count_SET', 'Query_Cache_Memory_bytes',
    'Query_Cache_Entries', 'Query_Cache_Purged'
)
UNION ALL
SELECT variable_name, variable_value
FROM global_variables
WHERE variable_name = 'mysql-query_cache_size_MB';
```

stats_mysql_query_rules.hits MySQL-Query-Cache-Size-MB MAIN runtime

Ubuntu 24.04.4 LTS 2 vCPU 4 GiB RAM PHP 8.4.26 loopback ProxySQL

- 2.7 7.3-12-g50b7f85
- 3.0 10.5-60-g7e9e009

15 800 150 300 600 SELECT PHP SQL

15 AB/BA PmaControl ProxySQL 600 ProxySQL 27.3 3.0.6

rule_id+1 5,000

ProxySQL PmaControl

3.0 PHP

prepare/step/copy/finalize SQL

ProxySQL

30 300 Python

$100 \times (1 - \text{time_3.0} / \text{time_2.7})$ $\text{time_2.7} / \text{time_3.0}$ commit PR

```
curl -fsSL0 https://pmacontrol.com/downloads/proxysql-2.7-vs-3.0/benchmark-data.json
curl -fsSL0 https://pmacontrol.com/downloads/proxysql-2.7-vs-3.0/analyze.py
python3 analyze.py benchmark-data.json
```

PmaControl revision:
b4e7b4c6ca72cf27077ddbdb1f2c89af05800b5a
QueryCacheEffectiveness helper SHA-256:
c47a169935f665259554bd0fbba85e90af9cbcd3983bf7ad3548e2caaf63d4dc

- PNG 1600 × 2000
- SVG
- JSON
- Python
- LinkedIn
-

PDF

PmaControl PR #6389 issue #639 Forgejo