YOORshop SAS
RCS 817 466 147 — 254 Rue Vendôme, 69003 Lyon, France — yoorhosting.com
[company details to be completed]

Report Ref.: YH-DEMO-2026-02
Analysis Date: October 30, 2026

DIAGNOSTICS & KEY FIXES

APPLICATION & DATABASE ENGINEERING REPORT • DELIVERABLE #2

Website Performance Optimization

Analysis of 70% CPU spikes, menu cache contention, and SQL rewrite of a 787,504-row table scan.

RECIPIENT CLIENT

Company: SARL Moto Pièces Express (Fictional)
Online Store: https://boutique-moto.example (PrestaShop 8.2.3)
Dedicated Server: srv-demo.yoorhosting.com (16-core Xeon, 64 GB RAM)
Environment: PHP-FPM pool motodemo_prod (48 workers) • MariaDB

OBSERVED CPU LOAD
70% to 1,270%
45 spikes recorded during imports

PHP SLOW QUERIES
540 / 976
Stalled in menu preg_grep

MEASURED SQL SPEEDUP
x 4,000
From 1.60 s down to 0.0004 s per call

FINAL NEW-PRODUCTS QUERY TIME
0.0004 s
Via direct indexed comparison

CLIENT REQUEST

"Experiencing severe slowdowns during regular price and stock updates via automated import modules, along with recurring processor saturation (spikes reaching 1,270% cumulative CPU across 16 cores), the client commissioned YOORshop to **pinpoint the exact root causes of the stalls** and prescribe an **optimization action plan with zero negative impact on the catalog**."

IDENTIFIED COMMON OVERLOAD MECHANISM

Cross-analysis of the PHP slow query log (*PHP slowlog*) and database update timestamps (`date_upd`) uncovered a structural bottleneck: **every single product save triggers the simultaneous purge of two heavy application caches**, which are immediately rebuilt in a loop during synchronization streams:

Established Temporal Correlation: Updating 22 products within one hour coincides to the exact second with the surge in slow queries (20:44, 20:47, 20:49, 20:50, then 20:52 to 20:56). A total of 45 load spikes were recorded, triggered by back-office POST requests executed from operator IP address `198.51.100.22` .

1 Issue #1: Excessive Menu Regeneration (Module `ps_mainmenu`)

ANALYSIS POINT	TECHNICAL FINDING & VALIDATED SOLUTION
Excessive Hooking	The module is unnecessarily hooked onto `actionObjectProductAddAfter` , `actionObjectProductUpdateAfter` , and `actionObjectProductDeleteAfter` (table `psdemo_hook_module`). These hooks force a call to `clearMenuCache()` (ps_mainmenu.php:1007), wiping the entire JSON cache.
Rebuild Cost	**540 of the 976 slow PHP requests** stall inside `preg_grep` (ps_mainmenu.php:796). Rebuilding compares **9,788 categories against 26,968 images**. Two concurrent worker processes recalculate the exact same file in parallel (write collision at 21:02:01 and 21:02:05).
Immediate Remediation	**Unhook `ps_mainmenu` from the three product hooks** in the PrestaShop back-office (*Design* > *Positions*). 100% risk-free change: the displayed menu contains no individual products (configuration `MOD_BLOCKTOPMENU_ITEMS = CAT77,CAT71,LNK1,LNK3`). Category and CMS page hooks remain active.

2 Issue #2: Non-Indexed Query in "New Products" Block (`ps_newproducts`)

After every product update, module `ps_newproducts` runs `Product::getNewProducts()`. The original query applies a date function directly on the indexed column, preventing MariaDB from using the index:

Original Query (Non-Sargable) — **1.60 seconds**

`WHERE DATEDIFF(date_add, NOW()) < 20`

MariaDB Impact: The `date_add` index is bypassed. Full Table Scan across **787,504 rows** in `psdemo_category_product`, on-disk temporary table creation, and heavy filesort.

✓ Optimized Query (Override) — **0.0004 seconds**

`WHERE date_add > '2026-10-07'`

MariaDB Impact: Immediate B-Tree index lookup. Scan restricted to the 8 required records. **Measured speedup: factor of 4,000**.

3 Custom PHP-FPM Pool Capacity Sizing

To absorb concurrent traffic spikes without HTTP 503 rejections, the `motodemo_prod` pool was custom-sized by our engineering team:

MAX PROCESSES (CHILDREN)
48 workers
Shared hosting default: 15

ALLOCATED MEMORY
1,024 MB / worker
Absorbs heavy catalog imports

AVERAGE CRON LOAD
57% of 1 core
Stabilized per-minute execution

CACHE REBUILD
Immediate
Process contention eliminated

4 Summary of Actions & Measured Gains

#	RECOMMENDED ACTION	IMPLEMENTATION METHOD	MEASURED IMPACT
1	Unhook module `ps_mainmenu`	Back-office (Design > Positions)	**-540 slow queries** Eliminates `preg_grep` contention.
2	Deploy SQL override for `getNewProducts`	File in `override/classes/Product.php`	**Query time ÷ 4,000** Ends the 787,504-row MariaDB scan.
3	Clean up monitoring probes	Purge of `/root/cpuwatch/`	**Completed** Probes stopped on Oct 28 at 12:44.

Technical Engineering Sign-off:
Web Engineering & Performance Division — YOORshop SAS

✓ Validated Optimization Plan