YOORshop SAS
RCS 817 466 147 — 254 Rue Vendôme, 69003 Lyon, France — yoorhosting.com
[Unternehmensangaben zu ergänzen]

Bericht-Nr.: YH-DEMO-2026-02
Analysedatum: 30. Oktober 2026

● DIAGNOSE & KERNLÖSUNGEN

ANWENDUNGS- & DATENBANK-ENGINEERING • ERGEBNIS NR. 2

Website-Leistungsoptimierung

Analyse von 70-%-CPU-Spitzen, Menü-Cache-Blockaden und SQL-Optimierung eines Scans über 787 504 Zeilen.

AUFTRAGGEBER

Unternehmen: SARL Moto Pièces Express (Fiktiv)
Online-Shop: https://boutique-moto.example (PrestaShop 8.2.3)
Dedizierter Server: srv-demo.yoorhosting.com (Xeon 16 Kerne, 64 GB RAM)
Umgebung: PHP-FPM-Pool motodemo_prod (48 Worker) • MariaDB

GEMESSENE CPU-LAST
70 % bis 1 270 %
45 Lastspitzen bei Katalogimporten

LANGSAME PHP-ANFRAGEN
540 / 976
Blockiert in preg_grep des Menüs

SQL-BESCHLEUNIGUNG
x 4 000
Von 1,60 s auf 0,0004 s pro Abfrage

ABFRAGEZEIT NEUE ARTIKEL
0,0004 s
Durch direkten Indexvergleich

KUNDENANFRAGE

„Angesichts starker Verzögerungen beim automatisierten Import von Preisen und Beständen sowie wiederkehrender CPU-Sättigungen (Spitzen bis 1 270 % auf 16 Kernen) beauftragte der Kunde YOORshop, **die genaue Ursache der Blockaden zu ermitteln** und einen **Optimierungsplan ohne negative Auswirkungen auf den Katalog** vorzulegen."

IDENTIFIZIERTER GEMEINSAMER ÜBERLASTUNGSMECHANISMUS

Der Abgleich des PHP-Slowlogs mit den Änderungszeitstempeln der Datenbank (`date_upd`) zeigte einen strukturellen Engpass: **Jede Produktänderung leert gleichzeitig zwei aufwendige Anwendungs-Caches**, die während laufender Synchronisationen permanent neu aufgebaut werden:

Nachgewiesene Zeitkorrelation: Die Änderung von 22 Produkten pro Stunde deckt sich sekundengenau mit den Slowlog-Spitzen (20:44, 20:47, 20:49, 20:50, 20:52–20:56). Insgesamt wurden 45 Lastspitzen erfasst, ausgelöst durch Admin-POST-Anfragen von der Operator-IP `198.51.100.22` .

1 Problem Nr. 1: Unnötige Menü-Neuberechnung (Modul `ps_mainmenu`)

ANALYSEPUNKT	TECHNISCHER BEFUND & VALIDIERTE LÖSUNG
Übermäßige Hooks	Das Modul ist unnötig an `actionObjectProductAddAfter` , `actionObjectProductUpdateAfter` und `actionObjectProductDeleteAfter` gekoppelt (Tabelle `psdemo_hook_module`). Diese Hooks rufen `clearMenuCache()` (ps_mainmenu.php:1007) auf und löschen den gesamten JSON-Cache.
Neubaukosten	**540 von 976 langsamen PHP-Anfragen** hängen in `preg_grep` (ps_mainmenu.php:796). Der Neuaufbau vergleicht **9 788 Kategorien mit 26 968 Bildern**. Zwei Prozesse schreiben dieselbe Datei parallel (Schreibkonflikt um 21:02:01 und 21:02:05).
Sofortmaßnahme	`ps_mainmenu` **von den drei Produkt-Hooks lösen** (Back-Office: *Design* > *Positionen*). 100 % risikofrei: Das Menü enthält keine Einzelprodukte (`MOD_BLOCKTOPMENU_ITEMS = CAT77,CAT71,LNK1,LNK3`). Kategorie- und CMS-Hooks bleiben aktiv.

2 Problem Nr. 2: Nicht indizierte Abfrage im Block „Neue Produkte“ (`ps_newproducts`)

Das Modul `ps_newproducts` führt nach jedem Update `Product::getNewProducts()` aus. Die Originalabfrage wendet eine Funktion auf die indizierte Spalte an, wodurch MariaDB den Index nicht nutzen kann:

Originalabfrage (nicht sargable) — **1,60 Sekunden**

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

MariaDB-Folge: Index `date_add` wird ignoriert. *Full Table Scan* über **787 504 Zeilen** in `psdemo_category_product`, temporäre Disk-Tabelle und aufwendige Sortierung.

✓ Optimierte Abfrage (Override) — **0,0004 Sekunden**

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

MariaDB-Folge: Direkte Nutzung des B-Tree-Index. Zugriff nur auf die 8 benötigten Datensätze. **Gemessener Geschwindigkeitsgewinn: Faktor 4 000**.

3 Kapazitätsdimensionierung des PHP-FPM-Pools

Um parallele Lastspitzen ohne HTTP-503-Abweisungen abzufangen, wurde der Pool `motodemo_prod` individuell dimensioniert:

MAX. PROZESSE (CHILDREN)
48 Worker
Shared-Standard: 15

ZUGEWIESENER RAM
1 024 MB / Worker
Fängt große Katalogimporte ab

MITTLERE CRON-LAST
57 % eines Kerns
Stabilisierter Minutenzyklus

CACHE-WIEDERAUFBAU
Sofort
Keine Prozesskonflikte mehr

4 Maßnahmenübersicht & erwartete Gewinne

#	EMPFOHLENE MASSNAHME	UMSETZUNG	GEMESSENE WIRKUNG
1	Entkopplung des Moduls `ps_mainmenu`	Back-Office (Design > Positionen)	**-540 langsame Anfragen** Beendet die `preg_grep` -Blockade.
2	Deployment SQL-Override `getNewProducts`	Datei in `override/classes/Product.php`	**Laufzeit ÷ 4 000** Beendet den 787 504-Zeilen-Scan.
3	Bereinigung der Messsonden	Aufräumen von `/root/cpuwatch/`	**Erledigt** Sonden am 28.10. um 12:44 gestoppt.

Technische Fachfreigabe:
Abteilung Web-Engineering & Performance — YOORshop SAS

✓ Optimierungsplan Validiert