

Výkonnostní audit: deep dive (vzorová zpráva)

Toto je vzor. „Acme Outdoor s.r.o.“ je smyšlený klient. Nálezy jsou kompozity ze skutečných auditů, které jsme odevzdali na jiných e-shopech, se změněnými detaily i čísly. Struktura a úroveň detailu odpovídají tomu, co dostane platící klient.

Klient	Acme Outdoor s.r.o. (smyšlený)
E-shop	acme-outdoor.example (Magento 2.4.6, frontend Luma)
Stack	Varnish 7, Redis 7, MySQL 8, 2 aplikační nody za load balancerem
Datové okno	28 dní field dat, 14 dní APM trace, jeden zátěžový test
Nástroje	New Relic, CrUX, Lighthouse, k6, varnishstat, MySQL slow log
Autor	Luboš Zápotočný, Zapolu s.r.o.
Zakázka	Deep dive, 7 pracovních dní, fixní cena

Pro stručný přehled si přečtete kapitulu 1 a kapitulu 5. Text mezi nimi tvoří podklady k jednotlivým nálezům.

1. Shrnutí

Pomalost e-shopu vzniká na straně serveru, nikoli kvůli nedostatečně výkonnému hardwaru. LCP z field dat je na katalogových stránkách **4,6 s na p75** (cíl: $\leq 2,5$ s); hlavní příčinou je **TTFB 1,9 s**. TTFB je vysoké proto, že full-page cache je fakticky vypnutá: Varnish obsluhuje jen **34 %** katalogových požadavků a 38 % veškerého serverového času připadá na vykreslování produktových stránek, v průměru 2,2 s na požadavek. Během večera květnové kampaně dosáhl checkout p95 **6,1 s a 3,4 % požadavků selhalo**, zatímco CPU aplikačních serverů nepřekročilo 12 %.

Následuje jedenáct nálezů. Šest z nich stojí den práce nebo méně a dva z těch šesti se týkají jen konfiguračních souborů. Provedení řádků 1-5 plánu oprav by mělo dostat katalogové LCP pod 2,5 s na p75 a zastavit selhání checkoutu ve špičce; řádek 6 pak dostane checkout p95 pod 2 s během kampaní. Zbývající řádky jsou seřazené podle návratnosti, takže plán lze kdykoli ukončit.

Přepočteno na peníze: během tří hodin kampaně 12. května skončilo chybou zhruba 230 checkoutů. Při průměrné hodnotě objednávky 1800 Kč (≈ 72 €) je to asi 415 000 Kč v objednávkách, které selhaly, ještě před započtením zákazníků, kteří checkout po šesti sekundách čekání opustili. Ostatní nálezy uvádíme v milisekundách; částka v korunách by u nich byla spekulativní, proto ji neuvádíme.

2. Co a jak jsme měřili

- Metriky reálných uživatelů:** Core Web Vitals z 28denního CrUX dotazu, segmentované podle typu stránky (homepage / kategorie / produkt / checkout). Syntetické Lighthouse běhy sloužily jen k reprodukci nálezů, ne k jejich hodnocení. Jak si e-shop stojí proti prahovým hodnotám Googlu:

Metrika (p75, field data)	E-shop	Práh	Verdikt
LCP	4,6 s	< 2,5 s	neprošlo
INP	280 ms	< 200 ms	neprošlo
CLS	0,04	< 0,1	prošlo
FCP	2,9 s	< 1,8 s	neprošlo
TTFB	1,9 s	< 0,8 s	neprošlo

- **Trasování na serveru:** transakční profil z New Relic za 14 dní, tedy kam serverový čas doopravdy jde:

Transakce	Podíl serverového času	Průměrná odpověď
catalog/product/view	38 %	2,2 s
catalog/category/view	17 %	1,8 s
checkout REST (košík + doprava)	11 %	1,4 s (6,1 s p95 ve špičce)

K tomu MySQL slow query log, profil volání externích služeb a statistiky hit/miss Varnishe podle URL vzorů z varnishstat.

- **Zátěžový test:** replay tvaru provozu z kampaně 12. května (0 → 420 req/s během 20 minut) proti stagingu se stejným stavem cache jako v produkci.
- **Revize konfigurace:** nastavení PHP a OPcache, Composer autoloader, env.php, Varnish VCL a nastavení MySQL, každé proti dokumentovaným doporučením pro Magento.
- **Audit bundlů a assetů:** co se posílá při prvním zobrazení produktové stránky, co blokuje vykreslení a co je zbytečná zátěž.

U většiny nálezů citujeme přesný příkaz nebo dotaz, který jsme použili, jednak aby si váš tým mohl naši práci překontrolovat, jednak proto, že stejné sondy poběží na kterémkoli Magento e-shopu, včetně toho vašeho. U každého čísla níže je uvedeno, odkud pochází; kde je dopad nálezu odhadem, je tak označen.

3. Přehled nálezů

Vysoká: dopadá na zákazníky, nebo na ní závisí tržby. **Střední:** denně ji pociťuje tým, nebo tiše poškozuje data. **Nízká:** malé úsilí, které odstraní reálné provozní riziko.

#	Nález	Závažnost	Pracnost	Očekávaný dopad
F1	Full-page cache je fakticky vypnutá a nejde invalidovat	Vysoká	1-2 dny	-700 ms TTFB na p75 v katalogu
F2	PHP runtime a autoloader špatně nastavené pro Magento	Vysoká	0,5 dne	-400 ms na každém necachovaném požadavku
F3	Checkout blokuje synchronní dotaz na sklad v ERP	Vysoká	3-5 dní	-2,8 s checkout p95 ve špičce; konec timeoutových chyb
F4	Frontend posílá 2,3 MB JavaScriptu, blokuje vykreslení	Vysoká	projekt	-1,2 s LCP (dočasná zmírnění: -0,4 s)
F5	Produktové obrázky: 1,6MB PNG, bez responzivních velikostí	Střední	1 den	-0,6 s LCP na produktových stránkách

#	Nález	Závažnost	Pracnost	Očekávaný dopad
F6	Indexery ponechané na „Update on Save“	Střední	0,5 dne	Uložení v adminu 30 s → ~2 s; konec nesouladu skladových stavů
F7	PHP-FPM pool vyčerpaný session locky ve špičce	Vysoká	2 dny	Checkout přežije kampaňový provoz
F8	Varnish bez grace módu	Nízká	0,5 dne	Ztlumí nápor po každém flushi cache
F9	Dvě extenze poslouchají stejný event; jedna je opuštěná	Střední	1 den	~300 ms na přidání do košíku
F10	41 GB mrtvých indexerových temp tabulek v MySQL	Nízká	0,5 dne	Zálohy o 60 % menší; obnova zhruba za polovinu času
F11	Order grid v adminu filtruje přes neindexovaný sloupec	Nízká	0,5 dne	Grid objednávek 12 s → pod 1 s

4. Nálezy podrobně

F1: Full-page cache je fakticky vypnutá a nejde invalidovat (Vysoká)

Co jsme viděli. Varnish hlásí na katalogových URL hit rate 34 %. Pro katalog této velikosti a s tímto podílem crawlerů je normál nad 90 %. Každá kategorie i produktová stránka nese X-Magento-Cache-Debug: MISS. Příčiny jsou dvě nezávislé:

1. Upravený default.xml označuje jeden blok v hlavičce (přepínač store view, který čte zákaznickou session) jako cacheable="false". V Magentu jediný necachovatelný blok v layoutu udělá necachovatelnou **celou stránku**. Blok přibyl v listopadové aktualizaci extenze; hit rate se zhroutil tentýž týden.
2. Varnish je zapnutý v administraci (system/full_page_cache/caching_application = 2), ale v env.php chybí sekce http_cache_hosts, takže Magento vůbec neumí posílat purge požadavky do Varnishe. Tým to obchází restartem Varnishe po každém deployi, a právě proto jsou deploye problematické (viz F8).

Jak jsme to ověřili:

```
$ curl -sI https://acme-outdoor.example/stany | grep -i cache-debug
X-Magento-Cache-Debug: MISS
# stejně na každé katalogové URL

$ grep -R 'cacheable="false"' app/design app/code | wc -l
1
# přepínač store view v default.xml

$ php -r 'var_export(array_key_exists("http_cache_hosts", (include "app/etc/env.php")));'
false
# Magento neumí poslat purge do Varnishe
```

Oprava. Vykreslit přepínač přes private content (customer data JS), takže stránka zůstane cachovatelná (dokumentovaný vzor pro fragmenty závislé na session), a doplnit http_cache_hosts do env.php, aby fungovala invalidace přes tagy a restarty mohly skončit.

Pracnost / dopad / riziko. 1-2 dny včetně regresních testů. Očekávaných ~700 ms TTFB na p75 katalogu (změřeno: cachovaná katalogová odpověď se vrací za 80 ms, necachovaná za 1,9 s). Riziko: nízké; obě změny jsou ohraničené a vratné po jednotlivých deployích.

F2: PHP runtime a autoloader špatně nastavené pro Magento (Vysoká)

Co jsme viděli. Codebase obsahuje ~125 000 PHP souborů; OPcache je nastavená na 10 000. Composer autoloader nebyl nikdy optimalizován: `vendor/composer/autoload_static.php` neobsahuje class mapu a v New Relic trace jsou vidět celé sekundy strávené v resoluci `objectManager factory` na studených cestách. Aktuální versus doporučené:

Nastavení	Aktuální	Doporučené
<code>opcache.max_accelerated_files</code>	10 000	100 000
<code>opcache.memory_consumption</code>	128	512
<code>opcache.validate_timestamps</code>	On, revalidace každé 2 s	Off; reset při deployi
<code>opcache.enable_cli</code>	Off	On (indexery a crony jsou taky PHP)
<code>realpath_cache_size</code>	4M	32M
<code>realpath_cache_ttl</code>	120	7200
Composer autoloader	neoptimalizovaný	<code>dump-autoload --optimize</code> v buildu

Jak jsme to ověřili:

```
$ find . -type f -name '*.php' | wc -l
124862
# PHP souborů v codebase

$ php -i | grep 'opcache.max_accelerated_files'
opcache.max_accelerated_files => 10000
# pojme 8 % z nich

$ grep -c 'classMap' vendor/composer/autoload_static.php
0
# autoloader nebyl nikdy optimalizován
```

Proč. S těmito hodnotami PHP průběžně znovu statuje a překládá velkou část codebase; profily připisují přes polovinu času necachovaného renderu načítání tříd a sestavování stránky místo byznysové logiky.

Oprava. Jedna změna konfigurace plus jeden krok v buildu. Jediná vazba, kterou je nutné respektovat: s vypnutým `validate_timestamps` musí deploy pipeline při release resetovat OPcache. Přesný krok pro vaši současnou pipeline jsme popsali v plánu oprav.

Pracnost / dopad / riziko. 0,5 dne. Odhadem –400 ms na každém necachovaném požadavku (a rychlejší crony a indexery jako vedlejší efekt). Riziko: nízké, pokud krok v deployi přistane spolu s konfigurací.

F3: Checkout blokuje synchronní dotaz na sklad v ERP (Vysoká)

Co jsme viděli. V kroku dopravy ukazuje trace synchronní HTTPS volání na skladový endpoint ERP s 3sekundovým timeoutem a jedním opakováním; profil externích služeb připisuje tomuto jedinému endpointu 92 % veškerého čekání na externí systémy. Při kampaňové zátěži se ERP zpomalí, volání si vyberou celý timeout a požadavky checkoutu se začnou hromadit. Přesně odsud pochází p95 6,1 s i 3,4 % selhání. Zátěžový test to na stagingu reprodukuje.

Proč. Potvrzování skladu v reálném čase přibýlo po loňském incidentu s přeprodejem. Záměr je správný, jen vkládá latenci cizího systému do každého checkoutu.

Oprava. Přesunout srovnávání skladu mimo request: rezervovat proti lokální inventuře, potvrzovat asynchronně a alertovat na nesoulad skladových stavů. Ochrana proti přeprodeji zůstane zachována a checkout přestane na ERP čekat. (Je to tentýž vzor, o kterém píšeme v textu o synchronizaci skladu mezi ERP a e-shopem; odkaz vám na vyžádání poskytneme.)

Pracnost / dopad / riziko. 3–5 dní včetně alertu na nesoulad stavů. Odstraní ~2,8 s z checkout p95 ve špičce a timeoutová selhání úplně (obojí změřeno v zátěžovém testu se zaslepeným voláním). Riziko: střední; vyžaduje písemné odsouhlasení nového okna přeprodeje (odhadem < 0,1 % kampaňových objednávek, proti dnešním 3,4 % selhávajících checkoutů).

F4: Frontend posílá 2,3 MB JavaScriptu (Vysoká)

Co jsme viděli. První zobrazení produktové stránky stáhne 2,3 MB JavaScriptu v 61 souborech; blokování hlavního vlákna dělá na běžném telefonu 2,9 s. Čtyři ze čtrnácti marketingových tagů už nemají za sebou funkční účet.

Proč. Zčásti architektura frontendu Luma, zčásti osm let přirůstání tagů; žádný jednotlivý tag v součtu nepřevažuje.

Oprava, krátkodobě (toto čtvrtletí). Odstranit čtyři mrtvé tagy, odložit analytický bundle, líně načítat widget recenzí níže na stránce. Odhadem –0,4 s LCP, 2 dny.

Oprava, skutečná (samostatné rozhodnutí). Přestavba frontendu na Hyvä. Na srovnatelných katalozích dostane LCP do zelených čísel na p75 a sníží náklady na údržbu frontendu; je to projekt na 6–10 týdnů a zaslouží si vlastní nacenění, ne řádek v seznamu oprav. Zmiňujeme ho zde, aby krátkodobý odhad výše nebyl chápán jako dosažitelné maximum.

F7: PHP-FPM pool vyčerpaný session locky ve špičce (Vysoká)

Co jsme viděli. Během zátěžového testu začínají selhání kolem 280 req/s, zatímco CPU sedí na 12 %. Status FPM ukazuje všechny workery obsazené; trace ukazují čekání na session locky v Redisu: zákazníci pollující stránku potvrzení objednávky drží zámky, které řadí do fronty všechny další požadavky téže session.

Oprava. Vypnout zamykání session pro pollingové endpointy, zvednout počet workerů na to, co paměťová rezerva reálně dovolí (dnešní číslo pochází z menšího typu instance), a přidat alert na saturaci FPM; tento výpadek zůstal neoznamován, protože grafy CPU vypadaly zdravě.

Pracnost / dopad / riziko. 2 dny včetně opakování zátěžového testu. Spolu s F3 přežije staging 420 req/s s checkout p95 1,9 s. Riziko: nízké.

F5, F6, F8, F9, F10, F11 stručně

- **F5 Obrázky:** hlavní (hero) obrázky produktů jsou PNG o 1,6 MB servírované v jedné velikosti všem zařízením. Konverze do WebP + responzivní velikosti přes CDN: 1 den, –0,6 s LCP na produktových stránkách (změřeno na převedeném vzorku).
- **F6 Indexery:** přepnuté na „Update on Save“ při únorovém hromadném importu a už nikdy zpět. Uložení produktu v adminu trvá 30 s a nesoulad mezi skladem na webu a realitou ve skladu pochází z race conditions při reindexu. Vrátit plánované indexování: 0,5 dne.
- **F8 Varnish grace:** každý flush cache dnes pošle celou vlnu provozu na PHP. Grace mód servíruje staré objekty, dokud se cache znovu nenaplní: 0,5 dne.
- **F9 Duplicitní observery:** dvě extenze odebírají stejný checkout event; jednu vendor opustil v roce 2023 a při každém volání znovu načítá quote. Odstranit ji (její funkce se nepoužívá): 1 den včetně ověření, –300 ms

na přidání do košíku.

- **F10 Úklid databáze:** databáze nese 41 GB tabulek catalogrule_product__temp* (pozůstatky nedoběhnutých indexerů ze starší verze Magenta, která po sobě neuklízela) plus vlastní frontovou tabulku s 12 miliony řádků bez jakékoli retence. Najde je jedna query do information_schema:

```
mysql> SELECT COUNT(*) tables, ROUND(SUM(data_length+index_length)
-> /1024/1024/1024, 1) gb FROM information_schema.tables
-> WHERE table_name LIKE 'catalogrule\_product\_temp%';
+-----+-----+
| tables | gb   |
+-----+-----+
|      96 | 41.2 |
+-----+-----+
```

Smazání pozůstatků a nastavení pravidelného čištění: 0,5 dne. Zálohy se zmenší z 68 GB na 27 GB a obnova trvá zhruba polovinu času, což se počítá hlavně v den, kdy obnovu opravdu potřebujete.

- **F11 Order grid v adminu:** sloupec přidáný pro expedici filtruje přes neindexovaný atribut; grid se načítá 12 s a pokaždé blokuje jeden worker. Jedna indexová migrace: 0,5 dne.

5. Plán seřazený podle návratnosti

Pořadí	Položky	Pracnost	Co získáte
1	F2 + F8	1 den	Každý necachovaný požadavek –400 ms; flush cache už nevyvolají špičku zátěže
2	F1	1-2 dny	TTFB katalogu –700 ms; deploye přestanou vyžadovat restart Varnishe
3	F6 + F10 + F11	1,5 dne	Použitelný admin; konec nesouladu skladových stavů; zálohy na polovinu
4	F7	2 dny	Checkout přestane ve špičce selhávat
5	F5 + F9	2 dny	LCP produktové stránky –0,9 s dohromady
6	F3	3-5 dní	Checkout p95 pod 2 s během kampaní
7	F4 krátkodobě	2 dny	LCP –0,4 s, než padne rozhodnutí o frontendu
8	F4 přestavba	samostatné nacenění	LCP v zelené natrvalo

Řádky 1-6 dají dohromady 10-14 inženýrských dní a řeší všechna selhání, která zákazník pocítí. Po kterémkoli řádku můžete skončit a práce do té chvíle obstojí sama o sobě. Řádek 6 jako jediný zasahuje do architektury objednávkové cesty; ať už ho provede kdokoli, nechte návrh posoudit seniorním inženýrem.

Po řádku 6 přeměřte stejným dotazem na field data, kterým tato zpráva začala. Pokud se čísla nepohnou podle predikce, kontaktujte nás a zjistíme příčinu.

6. Co jsme nekontrolovali

Bezpečnost, SEO, přístupnost a kvalita kódu mimo profilované horké cesty jsou mimo rozsah této zakázky. Audit pokrývá výchozí store view; B2B store view sdílí stack, ale neměřili jsme ho zvlášť. Jedno omezení

přístupů: náš auditní uživatel se během okna nedostal k admin socketu Varnishe, takže runtime čítače Varnishe pocházejí ze snapshotů varnishstat, které nám exportoval hosting, ne z živé inspekce.

7. Další kroky

Plán oprav zvládne provést kterýkoli kompetentní tým včetně vašeho: každý nález pojmenovává změnu, ne jen symptom. Pokud ho máme provést my, řádky 1-6 se vejdou do dvou- až třítýdenní zakázky za fixní cenu podle našich standardních [podmínek spolupráce](#).

Dotazy k jednotlivým nálezům: info@zapolu.com, nebo si [domluvte hovor](#) a přineste vlastní čísla; na hovoru vám rovnou sdělíme, zda se audit vyplatí.