ИНЖЕНЕРНЫЙ КЕЙС · 10 МЛН SKU

# Когда PostgreSQL перестаёт успевать

Каталог автозапчастей на 10 млн позиций: замерили, нашли узкое место, вынесли поиск в Meilisearch. Ниже — цифры, реализация и честные ограничения.

**12,5 мс**

p50 на 10 млн
против 4,35 с у PostgreSQL

| | |
|---|---|
| p95, 10 клиентов | **40 мс** |
| Индекс | **13,2 ГБ** |
| Индексация | **1 ч 23 мин** |

ЗАДАЧА

## Что было на входе

**Данные**

100 поставщиков, разные CSV/XLSX/XML, цены, фото на чужих CDN, опечатки и разные названия одной детали.

**Объём**

Целевые 500 тыс. позиций. Крайний случай — 10 млн. Демо жило на 2 vCPU и 1,4 ГБ, поэтому 10 млн проверялись в отдельной лаборатории.

**Ограничения**

Без Bitrix и платных CMS. Go, PostgreSQL, бесплатный поиск. SEO, быстрый каталог, заказы и админка.

**Что нельзя**

Нельзя хранить оригиналы фото на сервере. Нельзя отдавать 10 млн с одной базы без деградации поиска.

ДИАГНОСТИКА

# Где именно болит

Сначала измерили каждый тип запроса, а не «оптимизировали всё».

| ЗАПРОС НА 10 МЛН | P50 | ЧТО ПРОИСХОДИТ |
|---|---|---|
| Страница марки, 24 товара | 0,20 мс | **индекс `brand_id, price`** |
| Страница модели | 0,67 мс | **индекс `brand_id, model_id`** |
| Keyset-пагинация | 0,21 мс | **вместо `OFFSET`** |
| `OFFSET 100000` | 146 мс | **перебирает и выбрасывает строки** |
| Фасеты марок | 462 мс | **агрегат по всей таблице** |
| Поиск «бампер»: count | 2,68 с | **1 млн совпадений** |
| Поиск «бампер»: топ-24 | 6,55 с | **`ts_rank_cd` по миллионам строк** |

**Вывод:** каталог по марке/модели/группе масштабируется на PostgreSQL. Болит только полнотекстовый поиск: rank и count по миллионам совпадений.

РЕШЕНИЕ

# Разделили ответственность

БРАУЗЕР

**HTML + CSS**

SEO, без SPA

GO

**Приложение**

каталог, импорт, заказы

POSTGRESQL

**Источник истины**

цена, наличие, заказы

MEILISEARCH

**Поисковый индекс**

текст, префиксы, опечатки

S3 + CDN

**Фото**

WebP, без оригиналов на сервере

**PostgreSQL — истина.**

Цена и остаток всегда из базы. Индекс не транзакционный.

**Meili — только поиск.**

Возвращает ID и total. Карточки собираются из PostgreSQL.

**Fallback.**

Meili упал → поиск идёт в PostgreSQL, сайт не отдаёт 500.

РЕАЛИЗАЦИЯ

# Что написано, а не нарисовано

01

### HTTP-клиент Meili

Индекс, настройки полей, батчи, ожидание задач, поиск с фильтрами и сортировкой. Без SDK.

02

### Индексатор

`cmd/searchindex`: полная переиндексация PostgreSQL → Meili батчами, restartable.

03

### Интеграция

`SEARCH_BACKEND`, `CardsByIDs`, сохранение порядка, fallback на PostgreSQL.

04

### Дельта после импорта

`IndexRun(runID)`: индексируются только изменённые строки, архивные удаляются.

05

### Метрики

`/metrics`: p50/p95 HTTP, поиск, ошибки, пул БД, кэш, количество товаров.

06

### Лаборатория

`cmd/bench` + `cmd/searchlab`: 10 млн строк, одинаковый запрос, p50/p95/QPS.

ЗАМЕРЫ

# 10 млн: до и после

Один и тот же запрос «бампер» с 1 млн совпадений. Не среднее по редким словам.

## PostgreSQL

БЫЛО

| | |
|---|---|
| count | **2,68 с** |
| топ-24 + rank | **6,55 с** |
| count + top в одном тесте | **4,35 с** |

Rank по 1 млн совпадений. Индексы не спасают: нужно читать и сортировать совпадения.

## Meilisearch

СТАЛО

| | |
|---|---|
| p50, 1 клиент | **7,3 мс** |
| p50, 20 клиентов | **12,5 мс** |
| p95, 10 клиентов | **40,0 мс** |
| QPS, 10 клиентов | **764** |
| QPS, 20 клиентов | **599** |

| КЛИЕНТОВ | P50 | P95 | QPS | КОММЕНТАРИЙ |
|---|---|---|---|---|
| 1 | 7,3 мс | 7,8 мс | 136 | **чистая latency** |
| 10 | 10,2 мс | 40,0 мс | 764 | **рабочий режим** |
| 20 | 12,5 мс | 114,3 мс | 599 | **p95 начинает расти** |
| 50 | 148,4 мс | 398,4 мс | 286 | **перегруз на этом железе** |

**Честно:** 50 клиентов на одном поисковом узле — уже деградация. Для SLA нужен отдельный узел, запас CPU и, при росте, шардирование. Кейс показывает не «магию», а точку перегруза.

ЭКСПЛУАТАЦИЯ

# Что происходит при сбоях

**Meili упал**

Приложение логирует ошибку и отдаёт поиск из PostgreSQL. Проверено: 200 за 3,0 с вместо 500.

**Импорт идёт**

Импорт пишет в PostgreSQL; после завершения вызывается дельта-индексация. Поиск не блокируется импортом.

**Индекс отстал**

Полная переиндексация — отдельная команда. В продакшене нужна очередь changed IDs и мониторинг расхождения.

**Индекс большой**

10 млн = 13,2 ГБ, 1 ч 23 мин на macOS x86_64 через Rosetta. Это отдельное окно и ресурсы, не «нажал кнопку».

ГРАНИЦЫ

# Что мы не делали и почему

**Не тащили Kubernetes**

Один Go-сервис + PostgreSQL + Meili проще в эксплуатации и дешевле.

**Не партиционировали «для галочки»**

Каталог уже 0,2–0,7 мс. Партиции нужны для обслуживания/архива, а не для этих запросов.

**Не хранили оригиналы фото**

Только ссылка и WebP-превью в кэше; в продакшене S3 + CDN.

**Не обещали 10 млн на 2 vCPU**

10 млн — lab. Демо — 500 тыс. Продакшен — отдельное железо.

СТОИМОСТЬ

## Сколько это требует

| КОМПОНЕНТ | 500 ТЫС. | 10 МЛН |
|---|---|---|
| Приложение | 2 vCPU / 1–2 ГБ | **4–8 vCPU / 8–16 ГБ** |
| PostgreSQL | 2 vCPU / 4 ГБ | **8 vCPU / 32 ГБ, отдельный диск** |
| Поиск | не обязателен | **8 vCPU / 32 ГБ, индекс 13+ ГБ** |
| Фото | диск + кэш | **S3 + CDN** |
| Бэкапы | ежедневно | **PITR + бэкап индекса/переиндексация** |

Ориентир, не коммерческое предложение: зависит от провайдера, трафика и требований к SLA. Точная смета — после замера на целевом железе.

ВОСПРОИЗВОДИМОСТЬ

## Как проверить

```
make lab-up
```

```
go run ./cmd/bench -dsn "$BENCH_DSN" -seed 10000000
```

```
go run ./cmd/searchlab -dsn "$BENCH_DSN" -index -bench
```

```
SEARCH_BACKEND=meilisearch go run ./cmd/server
```

Подробно: `lab/README.md`, `docs/MEILI.md`, `docs/CASE-JOURNAL.md`. Код и датасет — синтетика, без прайсов заказчика.

ПОДБОР ДЕТАЛИ

## Не нашли нужную запчасть?

Оставьте заявку: подберём деталь по VIN, артикулу или названию и вернёмся с ценой и фото.

✓ Проверка перед отправкой ✓ Ответ в течение рабочего дня ✓ Отправка по России

ВАШЕ ИМЯ

Александр

ТЕЛЕФОН

+7 (___) ___-__-__

МАРКА, МОДЕЛЬ, ДЕТАЛЬ

Audi A6 C5, бампер передний

КОММЕНТАРИЙ

Год выпуска, VIN или ссылка на фото

Отправить заявку

Нажимая кнопку, вы соглашаетесь с политикой конфиденциальности.