Engineer

Оптимізація продуктивності

30 досягнень

Поєднайте частини системи

Роботи за темою «Оптимізація продуктивності» показують одну частину дієвої системи. Ми можемо допомогти пов'язати її з рештою вашої діяльності. Пошук запитів і відсутніх індексів у системі, що рік за роком важчає, та виправлення тих, що справді мають значення.

Переглянути наші послуги Зв'язатися з нами →

Знайдіть схоже завдання

Відкрийте приклад, схожий на вашу ситуацію. Огляд пояснює рішення, що привели до результату, а не лише називає технологію. 30 днів даних про електроенергію за 180 мс — 180 ms на агрегацію місяця показань лічильників

30 днів даних про електроенергію за 180 мс Зв'язатися з нами →

30 днів даних про електроенергію за 180 мс

Розробила варіанти погодинної агрегації споживання електроенергії на Python, SQL і Bash + Jq; під час вимірювання Python обробив один 30-денний набір даних JSONL за 180 мс.

Прискорила пайплайн географічних даних у 50 разів

Зафіксувала прискорення географічного пайплайна у 50 разів у роботі засновниці для Utiligize завдяки змінам SQL і моделей даних у PostgreSQL, MS SQL та Google Cloud BigQuery.

Пришвидшила GIS-карти в 10 разів на PostgreSQL

Підвищила продуктивність застосунку географічних карт у 10 разів завдяки стратегічному переходу бази даних з MSSQL на PostgreSQL, включно з фінальним перенесенням даних (cutover), оптимізувавши обробку та безпеку даних.

Усунула 1 000 проблем у географічних даних

Усунула 1 000 проблем у географічних даних та часових рядах, використовуючи GDAL, ArcGIS, PostGIS, Mapbox, QGIS, SQL (PL/pgSQL, Transact-SQL), Bash, забезпечивши високоякісну обробку великих даних.

Вела 100 сховищ PostgreSQL, MS SQL і BigQuery

Спроєктувала та створила 100 баз даних для сховищ даних на PostgreSQL, MS SQL та Google BigQuery, переважно з GIS-даними та часовими рядами, і керувала ними, оптимізувавши продуктивність і масштабованість.

Підтримувала 10 серверів PostgreSQL і MS SQL

Спроєктувала, розгорнула та підтримувала 10 серверів PostgreSQL і MS SQL на Ubuntu Linux VPS, забезпечивши оптимальну продуктивність і надійність серверів.

Прискорила PostgreSQL у 10 разів

Прискорила PostgreSQL у 10 разів завдяки стратегічній індексації, партиціонуванню та оптимізації запитів, підвищивши ефективність бази даних для користувацьких, тенантних, геопросторових та часових електричних даних.

Керувала full-stack-розробкою GIS-карт

Керувала full-stack-розробкою GIS-карт, наглядаючи за роботою з PostgreSQL, Mapbox, ReactJS та NodeJS, щоб створити інтегроване рішення.

Реорганізувала процеси, заощадивши 8 000 годин

Реорганізувала внутрішні процеси, заощадивши 8 000 годин завдяки вдосконаленню архітектури програмного забезпечення, систем та ефективності планування.

Заощадила 150 годин на місяць у 30 проєктах

Підвищила ефективність 30 проєктів: оптимізувала робочі процеси й управління ресурсами та заощадила 150 годин на місяць.

Адмініструвала 40 вебсайтів на Apache і Nginx

Адмініструвала 40 вебсайтів на хостинг-серверах Ubuntu Linux з Apache та Nginx, забезпечивши високу доступність і продуктивність.

Розробила 600 пайплайнів ETL/ELT на PL/pgSQL

Розробила 600 пайплайнів ETL/ELT на основі PL/pgSQL для оптимізації складних робочих процесів обробки даних у кількох середовищах розробки та продакшну PostgreSQL.

Спроєктувала наскрізний JSON від Postgres до API

Спроєктувала архітектуру наскрізної передачі JSON, у якій функції PostgreSQL повертають повний JSON, що дослівно пересилається через Go API, усунувши проміжну десеріалізацію (unmarshalling) та унезалежнивши фронтенд від змін схеми.

Впорядковані за часом ключі для моделі даних

Запровадила впорядковані за часом ідентифікатори UUID v7 (PostgreSQL 18) як ключі сутностей, щоб зменшити фрагментацію індексів B-tree та пришвидшити запити.

Змоделювала 348 таблиць у 34 схемах PostgreSQL

Змоделювала морську предметну галузь — фахівців, компанії, судна, вакансії, відгуки та решту — у 348 нормалізованих таблицях у межах 34 схем PostgreSQL, з довідниками SMALLINT та ключами UUID v7.

Побудувала фронтенд Next.js 16: SSR, SSG і CSR

Побудувала фронтенд на Next.js 16 зі свідомими стратегіями SSR, SSG і CSR та багаторазовою фабрикою prefetched-server-page, що усуває каскад запитів N+1 на кожній автентифікованій сторінці, перенесеній на неї.

Реалізувала повну підтримку Progressive Web App

Реалізувала повну підтримку Progressive Web App — з можливістю встановлення та офлайн-роботи — з кешуванням Workbox під час виконання через next-pwa.

Перенесла повільні операції в чергу завдань River

Перенесла повільну роботу зі шляху запиту в чергу завдань River — 15 модулів воркерів, 8 запланованих задач та 20 завдань pg_cron — щоб запит повертався, поки робота за ним триває.

Побудувала ліміти запитів і захист від зловживань

Побудувала засоби протидії зловживанням за принципом fail-closed — 22 обмежувачі частоти на Redis, Cloudflare Turnstile, ідемпотентність запитів та прив'язку до origin — щоб платформа відсікала ботів і напливи, а не довіряла тим, хто її викликає.

Побудувала власну аналітику в PostgreSQL

Побудувала власну продуктову аналітику в PostgreSQL — 47 функцій над партиціонованими таблицями подій, які самі себе очищають, — псевдонімізовану за допомогою солі, що регулярно змінюється, і ввімкнену лише за згодою відвідувача.

Побудувала систему лояльності та репутації

Побудувала систему лояльності та репутації — 67 функцій на основі реєстру з 31 таблиці, з лігами, бейджами та крамницею обміну, — і блокувала рядок балансу, щоб закрити вікно для подвійного витрачання.

Публічні служби на невеликому сервері

Розмістила публічні служби компанії на одноядерному хості з 512 МБ — git-сервер, вебсервер для семи доменів, Tor, два сервери альтернативних протоколів і блокування вторгнень, — використовуючи 464 МБ доступної пам'яті як проєктне обмеження; автоматизація резервування є окремою реалізацією.

Побудувала моніторинг хоста з dead man's switch

Побудувала моніторинг за принципом dead man's switch, який надсилає пінг, лише поки пам'ять і диск у нормі, тож хост у погіршеному стані здіймає тривогу своїм мовчанням, — і виявила шість імен змінних зі словом «free» там, де перевірка правильно вимірювала «available»: на машині з 464 МБ ці значення різняться на порядок.

Побудувала сайт на Hugo 3 мовами

Побудувала багатомовний статичний сайт на Hugo зі 108 файлів шаблонів, з яких 53 партіали, що публікує кожну сторінку в чотирьох форматах з одного дерева контенту трьома мовами, а також на семи тематичних субдоменах, зібраних із того самого дерева.

Страховка, що більше не забирає пів години

Скоротила браузерну перевірку якості сайту з 1 636 секунд до 615, запускаючи перевірки від найдовших до найкоротших через пул воркерів із максимум чотирма одночасними перевірками, після того як виміряла, що алфавітний порядок коштував 320 секунд проти 224.

Скоротила пакунок стилів із 87 КБ до 31 КБ

Скоротила пакунок стилів із 87 КБ до 31 КБ, вийнявши з нього шрифт у base64, і позбулася 756 КБ у 22 файлах, які публікувалися під час кожного розгортання і на які ніщо не посилалося.

Вбудувала кодувальник QR на 522 рядки

Перенесла в репозиторій власний кодувальник QR — код Ріда-Соломона над GF(256), фіксоване розташування модулів, вісім шаблонів маски — на 522 рядках, замість додавати третю залежність середовища виконання, і перевірила його, зчитавши готову матрицю незалежно написаним декодером.

Обмежений потік подій у застосунку macOS

Обмежила буфери подій і журналювання після зафіксованого інциденту macOS зі зростанням на 41 МБ за секунду й 111 ГБ стиснених сторінок на машині з 36 ГБ; зафіксований повний тестовий запуск зменшив журнал ядра з 610 996 до 1 411 рядків.

Увімкнула сувору конкурентність Swift 6 поверх C

Перейшла на повну сувору перевірку конкурентності Swift 6 без акторів, з'єднавши блокувальний цикл подій C з головним актором через одного виробника, одного споживача й один порядок, — з'ясувавши, що окрема задача на кожну подію губить порядок, від якого залежить інтерфейс.

Пришвидшила набір тестів до 8,9 секунди

Скоротила набір тестів від шести тестів, що перевищували ліміт у шістдесят секунд, до 135 успішних за 8,9 секунди, профілюючи головний потік і прибравши два виклики, у яких він перебував 3 989 із 4 017 вибірок.

Спрямуйте інженерні зусилля

Надайте репрезентативний набір даних, запитання до нього та поточне вузьке місце. Узгодимо вимірювання якості, пропускної здатності та відновлення.

Дивитися, скільки це коштує Зв'язатися з нами →

Розгляньте тему у своєму контексті

Розкажіть, як «Оптимізація продуктивності» стосується вашої системи. Ми допоможемо обрати відповідну послугу та визначити обсяг роботи.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/categories/performance/