Engineer

30 días de datos eléctricos en 180 ms

Desarrollé variantes de agregación horaria del consumo eléctrico en Python, SQL y Bash + Jq, con una prueba en Python que procesó un conjunto JSONL de 30 días en 180 ms.

2025

180 ms

para agregar un mes de lecturas de contadores

Situación. La empresa procesa grandes volúmenes de datos eléctricos heterogéneos procedentes de varias fuentes, cada una con una frecuencia de comunicación distinta: desde intervalos horarios hasta intervalos de 15 minutos y, en algunos casos, marcas temporales irregulares. Esa variabilidad dificulta la creación de un conjunto de datos coherente y comparable. Para realizar análisis energéticos precisos, es necesario normalizar los datos en valores de consumo horario consistentes, agregados por zona.

Tarea. El objetivo era transformar eventos sin procesar de un conjunto de datos histórico en formato JSONL en cifras de consumo eléctrico alineadas por horas, con una fila por hora y zona. La tarea comprendía, en concreto:

- Alinear los datos con marca temporal a intervalos horarios estrictos.
- Agregar la producción eléctrica total de cada intervalo sumando los valores de la combinación de fuentes.
- Incorporar los intercambios transfronterizos sumando importaciones y restando exportaciones.
- Producir los valores finales en un formato estructurado y escalable, adecuado para análisis posteriores.
- La solución también debía ser suficientemente eficiente para cubrir periodos largos, de 30 días o más, y varios países o zonas.

Acción. Para abordar la tarea se implementaron tres variantes con Python, JQ — para procesar JSON en la línea de órdenes — y SQL, optimizadas para contextos distintos:

Se eligió Python por su flexibilidad, facilidad para manipular datos y capacidad para realizar transformaciones en memoria de forma eficiente. Con funciones de Python puro se construyó un flujo de procesamiento que:

- Analizaba los archivos JSONL y los convertía en estructuras tabulares.
- Remuestreaba las series temporales en intervalos horarios.
- Agregaba la producción y calculaba el consumo eléctrico neto (producción + importaciones – exportaciones).
- Exportaba los resultados a CSV o los cargaba en una base SQLite ligera para inspeccionarlos.

JQ permitió crear una solución rápida y con pocas dependencias para entornos de línea de órdenes mediante una cadena de filtros JQ.

En PostgreSQL se importaron los datos JSONL, se crearon tablas normalizadas y se escribió una serie de consultas SQL.

Para evaluar el rendimiento, se compararon las soluciones con conjuntos históricos de 1 día y de 30 días.

Resultado. La implementación Python resultó la más rápida y escalable, con estos tiempos:

- Procesamiento de 1 día en solo 34 ms.
- Procesamiento de 30 días en 180 ms.

Esto confirmó su idoneidad para periodos más largos manteniendo tiempos inferiores a un segundo. Además, el código era claro y fácil de mantener, ampliar al procesamiento de varias zonas o integrar en un flujo ETL.

En conjunto, el enfoque con varias herramientas demostró flexibilidad en su elección, una optimización eficaz del rendimiento y un tratamiento sólido de los problemas de alineación temporal y agregación en flujos de datos eléctricos reales.

Aplique esta experiencia a su proyecto

Utilice «30 días de datos eléctricos en 180 ms» como punto de partida para un reto similar en su organización. Los servicios que respaldan este trabajo están disponibles para su proyecto. Detectar las consultas y los índices que faltan en un sistema cada año más pesado y corregir lo que realmente importa.

Explore nuestros servicios Póngase en contacto →

Mire más allá del resultado

Lea la situación, las decisiones y el resultado de este ejemplo. Podemos hablar de qué partes se adaptan a sus restricciones y cuáles necesitan otro enfoque. 30 días de datos eléctricos en 180 ms — 180 ms para agregar un mes de lecturas de contadores

Explore nuestros servicios Póngase en contacto →

¿Necesita flujos de datos rápidos y eficientes?

Converse sobre su proyecto con la ingeniera responsable de la entrega. Acordamos por escrito el alcance, la base del precio y el traspaso antes de empezar a trabajar.

Contexto de la función: Ingeniera de plataformas y datos. Se apoya en Bases de datos, PostgreSQL, SQL, Ingeniería de datos, Flujos de datos (ETL/ELT), Análisis de datos, Automatización y CI/CD, Optimización del rendimiento y Python. Y ya que está aquí, también se nos da bien Optimización del rendimiento de bases de datos, Desarrollo de flujos de datos (ETL/ELT) y Análisis de datos y paneles de BI.

Póngase en contacto Explore nuestros servicios

Una parte de **una trayectoria más amplia** — y toda ella está en este sitio.

Cada entrada sigue la misma estructura: la situación, la tarea, lo que hicimos y lo que cambió.

Vea la trayectoria completa

Defina la mejora que necesita

Traiga un conjunto de datos representativo, las preguntas que debe responder y el cuello de botella actual. Podemos acordar cómo medir la calidad, la capacidad de procesamiento y la recuperación.

Vea cuánto cuesta Póngase en contacto →

¿Necesita flujos de datos rápidos y eficientes?

Pregunte por las decisiones detrás de «30 días de datos eléctricos en 180 ms». Hablará con las personas que pueden explicar el trabajo y valorar su propio proyecto.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

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

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/portfolio/a-month-of-meter-readings-in-180-milliseconds/