Escuela de Datos Vivos

Curso Data Engineer: Creando un pipeline de datos

Módulo E - Clase 11

Agenda

- ML con Pablo Casas
- Arquitectura
- Exámen final

Arquitectura de Datos

Arquitectura Lambda

La Arquitectura Lambda, surgió en el año 2012 y se atribuye a Nathan Marz. La definió en base a su experiencia en sistemas de tratamiento de datos distribuidos durante su etapa como empleado en las empresas Backtype y Twitter.

Su objetivo era tener un **sistema robusto tolerante a fallos, tanto humanos como de hardware, que fuera linealmente escalable y que permitiese realizar escrituras y lecturas con baja latencia**.

Arquitectura Lambda

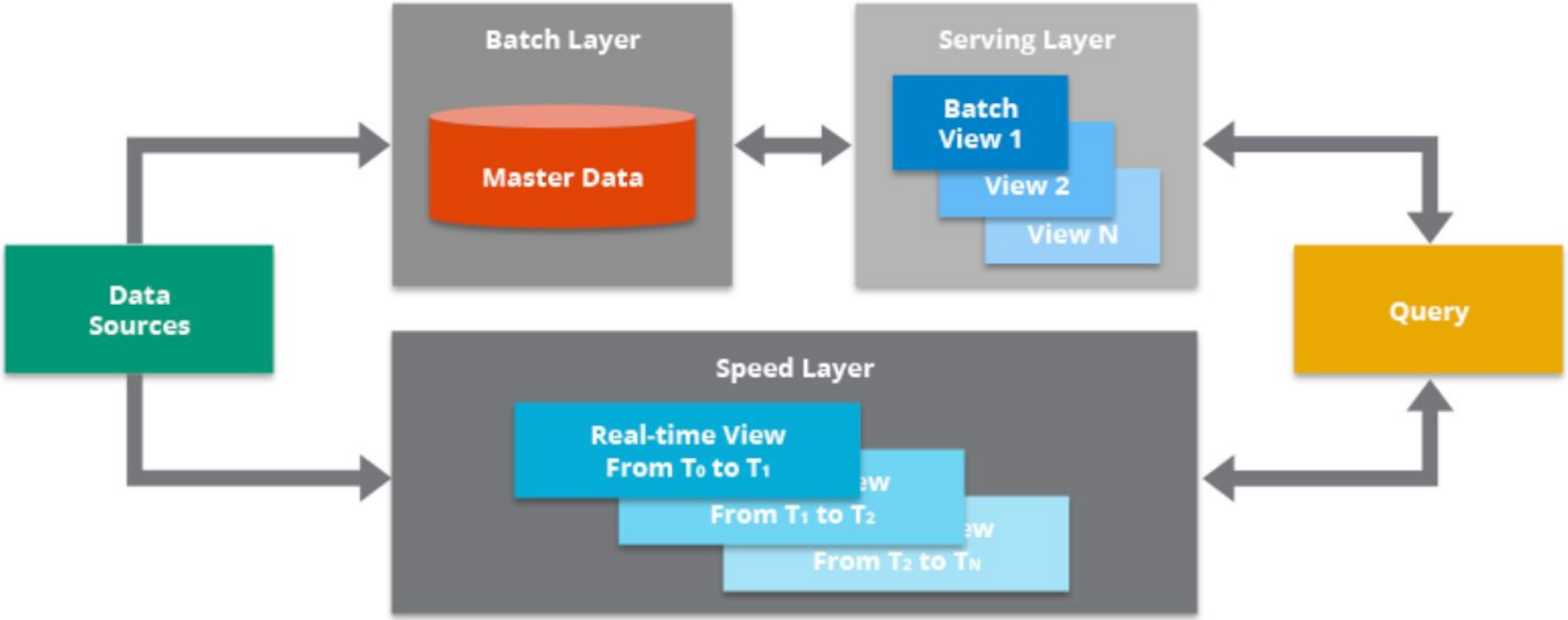
Componentes clave:

Capa de velocidad (Speed Layer): Procesamiento en tiempo real de datos entrantes. Utiliza herramientas como Apache Kafka y Apache Spark Streaming para procesar datos de manera incremental y proporcionar resultados con baja latencia (segundos a minutos). Ideal para análisis en tiempo real, paneles de control y detección de anomalías.

Capa por lotes (Batch Layer): Procesamiento periódico de grandes conjuntos de datos históricos. Utiliza herramientas como Apache Hadoop y Apache Hive para procesar datos de manera batch, lo que permite realizar análisis complejos y cálculos a gran escala. Ideal para generar informes, análisis retrospectivos y entrenamiento de modelos de aprendizaje automático.

Capa de servicio (Serving Layer): Proporciona acceso a los resultados procesados tanto de la capa de velocidad como de la capa por lotes. Permite a las aplicaciones y usuarios acceder a la información actualizada en tiempo real y a los resultados históricos procesados.

Arquitectura Lambda



Arquitectura Kappa

El término Arquitectura Kappa, fue introducido en 2014 por Jay Kreps en su artículo Questioning the Lambda Architecture.

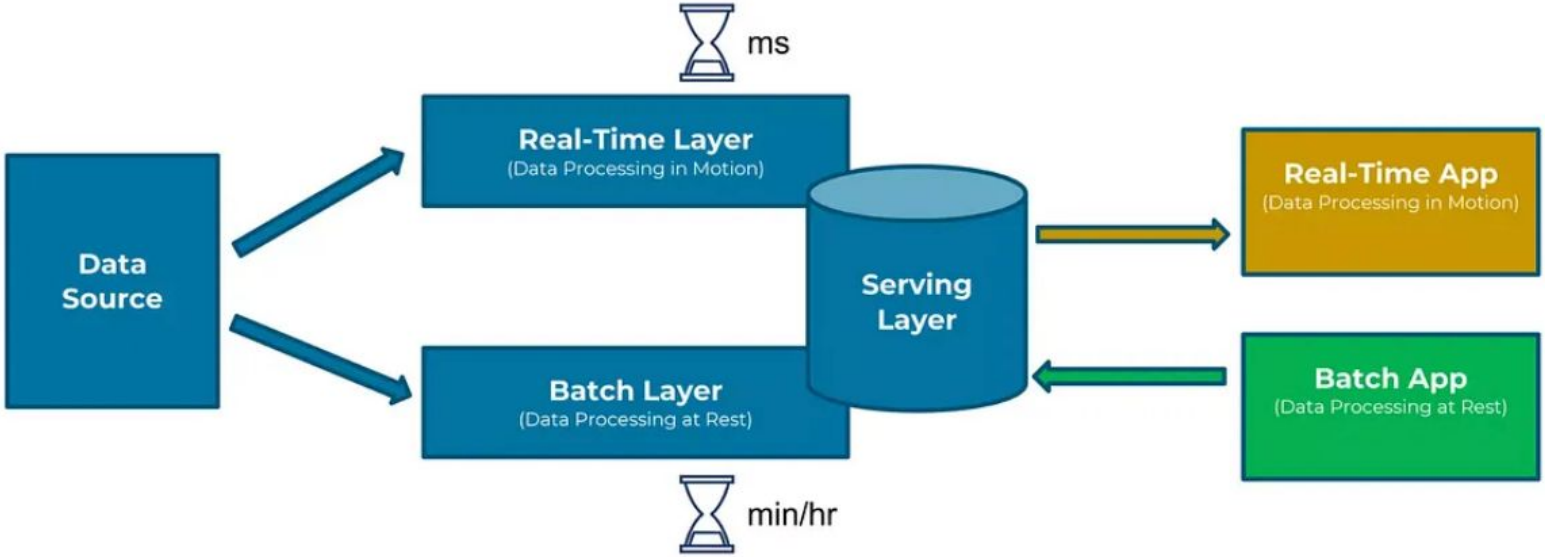
En él señala los **posibles puntos "débiles" de la Arquitectura Lambda** y cómo solucionarlos mediante una evolución. Su propuesta consiste en **eliminar la capa batch** dejando solamente la capa de streaming.

Esta capa, a diferencia de la de tipo batch, no tiene un comienzo ni un fin desde un punto de vista temporal y **está continuamente procesando nuevos datos** a medida que van llegando.

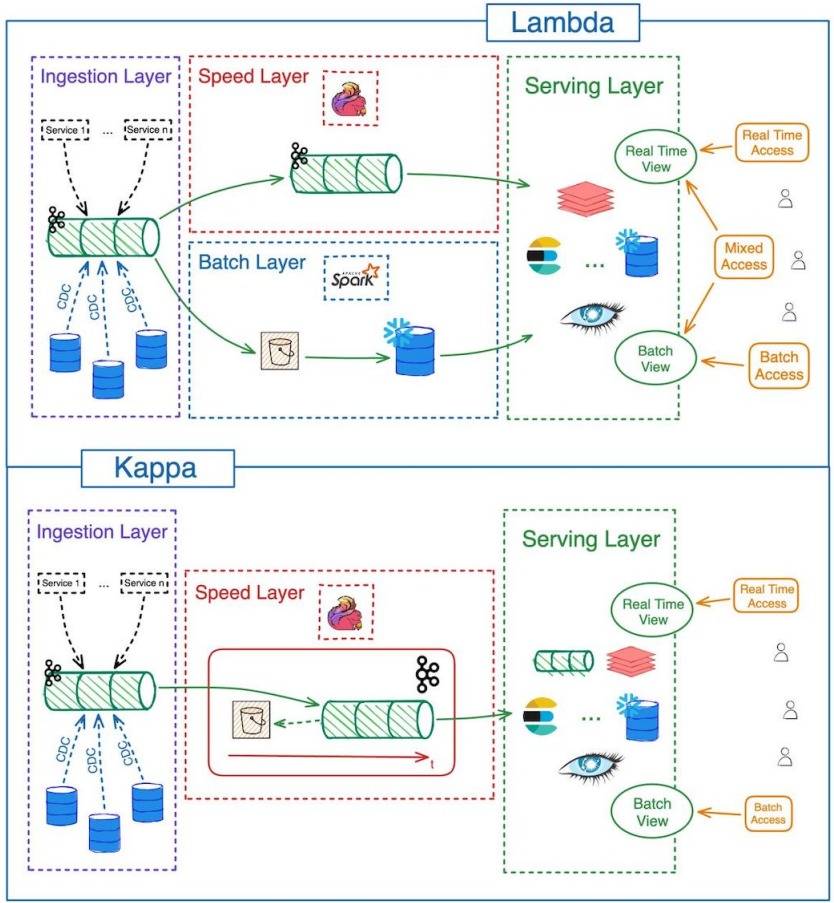
Como un **proceso batch se puede entender como un stream acotado**, podríamos decir que el procesamiento batch es un subconjunto del procesamiento en streaming.

Esta evolución consiste en una simplificación de la Arquitectura Lambda, en la que se elimina la capa batch y **todo el procesamiento se realiza en una sola capa denominada de tiempo real o Real-time Layer**, dando soporte a procesamientos tanto batch como en tiempo real.

Arquitectura Kappa



Kappa vs Lambda



Kappa vs Lambda

Aspect	Kappa Architecture	Lambda Architecture
Processing Paradigm	Stream Processing	Hybrid (Batch and Stream)
Layers	Single Layer (Ingestion & Processing)	Three Layers (Batch, Speed, & Serving)
Real-Time Focus	Primarily Real-Time	Mixes Real-Time and Batch
Batch Processing	Minimal or None	Essential (Handled by Batch Layer)
Complexity	Simple and Straightforward	More Complex
Fault Tolerance	Depends on Stream Processing	Built-in Fault Tolerance
Use Cases	IoT Data, Fraud Detection	Recommendation Systems, Analytics

Práctica Arquitectura

Ejercicio de arquitectura

Te acaban de contratar en una empresa de la industria minera como Data Engineer/Data Architect para delinear su arquitectura y sugerir qué herramientas deberían utilizar para ingestar la data, procesar la información, almacenarla en un datawarehouse, orquestar y realizar Dashboards que ayuden a la toma de decisiones basadas en datos.

Luego de realizar algunas reuniones con el team de analitica de la empresa pudimos relevar:

Sistema ERP: SAP con una base de datos Oracle

Sistema de Producción: App desarrollada "in house" con una base de datos Postgres.

Fuentes externas: un proveedor que realiza algunas mediciones de la calidad de las rocas le deja todos sus análisis en un bucket de AWS S3 con archivos Avro.

Mediciones en tiempo real: Utilizan +100 sensores de mediciones de vibración por toda la mina para detectar movimiento del suelo y se podrían utilizar para predecir posibles derrumbes.

Apps Mobile: La empresa cuenta con una app mobile donde trackean todos los issues pendientes con maquinaria de la mina.

Ejercicio de arquitectura

Desarrollar una arquitectura, que sea escalable, robusta, que sea orquestada automáticamente, que contemple seguridad, calidad, linaje del dato, que sea utilizada para procesar tanto información batch como información en tiempo real.

Responder las siguientes preguntas:

1. Utilizarían infraestructura on premise o en la nube?
2. ETL o ELT? Por qué?
3. Que herramienta/s utilizarían para ETL/ELT?
4. Que herramienta/s utilizarían para Ingestar estos datos?
5. Que herramienta/s utilizarían para almacenar estos datos?
6. Como guardarán la información, OLTP o OLAP?
7. Que herramienta/s utilizarían para Data Governance?
8. Data Warehouse, Data Lake o Lake House?
9. Qué tipo de información gestionarán, estructurada, semi estructurada, no estructurada?
10. Con que herramienta podrían desplegar toda la infraestructura de datos?

Exámen final

Ejercicio 1

Ejercicio 1

Ejercicio 1

Ejercicio 1

Ejercicio 1

Ejercicio 1

Ejercicio 2

Ejercicio 2

Ejercicio 2

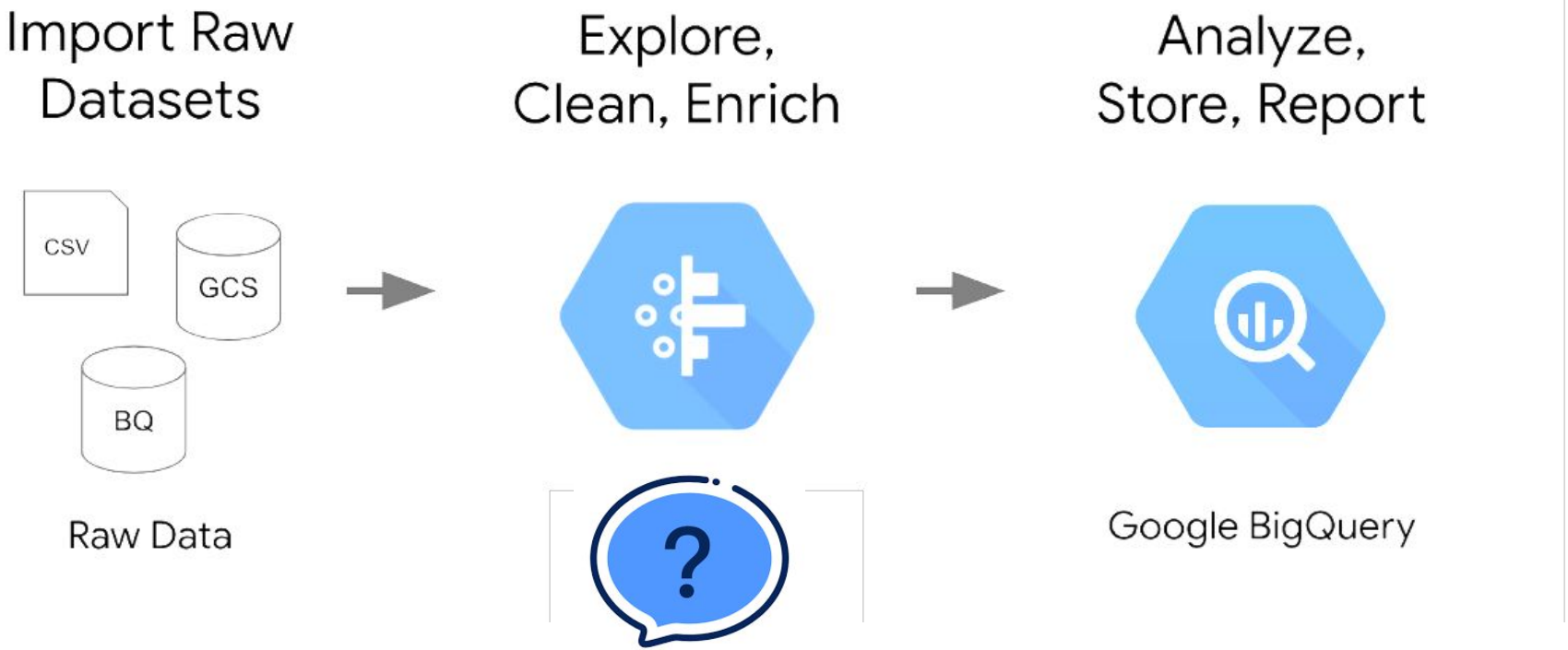
Ejercicio 2

Ejercicio 2

Ejercicio 2

Ejercicio 2

Ejercicio 3


Import Raw Datasets
Explore, Clean, Enrich
Analyze, Store, Report
CSV
GCS
BQ
Raw Data
?
Google BigQuery

Ejercicios 1 y 2

- Ponerse en ror de DE
- Elaborar conclusiones
- Elaborar recomendaciones de negocio
- Proponer diferentes arquitecturas (si aplica)

Ejercicio 3

- Trabajo de investigación
- Elaborar un Laboratorio
- Contestar preguntas técnicas
- Proponer una arquitectura según el problema planteado

Preguntas ?

Gracias