

Almacenamiento de grandes volúmenes de datos

Deborah Famadas Rodríguez

26 de noviembre de 2024

Recordemos que es el procesamiento de grandes volúmenes de datos

- **Definición:** Conjunto de datos tan grandes y complejos que requieren tecnologías y métodos avanzados para su procesamiento y análisis.
- **Origen:** Proviene de fuentes diversas como redes sociales, sensores, transacciones financieras, etc.
- **Importancia:** Permite extraer información valiosa para la toma de decisiones estratégicas.

Las 5 V de los datos

- ① **Volumen:** Cantidad de datos generados y guardados
- ② **Velocidad:** En este contexto, la velocidad a la cual se generan y procesan los datos para cumplir las exigencias y desafíos de su análisis.
- ③ **Variedad:** El tipo y naturaleza de los datos para ayudar a las personas a analizar los datos y usar los resultados de forma eficaz. Los macrodatos usan textos, imágenes, audio y vídeo.
- ④ **Veracidad:** Calidad y fiabilidad de los datos. *Data cleansing* y *ETL* (*Extract, Transform, Load*) ayudan a mejorar la veracidad.
- ⑤ **Valor:** Beneficio obtenido al analizar los datos.

Relevancia del almacenamiento

- **Gestión Eficiente:** Un almacenamiento adecuado permite gestionar grandes volúmenes de datos sin perder rendimiento.
- **Acceso Rápido:** Facilita el acceso y recuperación de datos, esencial para análisis en tiempo real.
- **Escalabilidad:** Sistemas como *HDFS* permiten aumentar la capacidad de almacenamiento según la demanda.

Desafíos en el almacenamiento

- **Escalabilidad:** Cómo expandir el almacenamiento sin interrupciones.
- **Rendimiento:** Necesidad de altas velocidades de lectura/escritura.
- **Fiabilidad:** Garantizar que los datos estén disponibles y sean consistentes.
- **Seguridad:** Protección contra accesos no autorizados. Encriptación y control de acceso son esenciales.

Un poco de historia

- ▶ **1984:** *Teradata Corporation* comercializa el sistema de procesamiento paralelo DBC 1012, pionero en el manejo de GVD mediante procesamiento paralelo.
- ▶ **1992:** Teradata logra almacenar y analizar 1 terabyte de datos.
- ▶ **2000:** *Seisint Inc.* desarrolla un marco de intercambio de archivos distribuido para almacenamiento y consultas de datos estructurados y no estructurados.
- ▶ **2004:** Google publica el artículo sobre *MapReduce*, introduciendo un modelo revolucionario para el procesamiento paralelo de datos a gran escala.
- ▶ **2006:** Apache Hadoop adopta el modelo *MapReduce*.
- ▶ **2012:** Se lanza *Apache Spark*, mejorando las limitaciones de *MapReduce* y ofreciendo procesamiento en memoria para mayor velocidad.

Sistemas de archivos distribuidos (DFS)

¿Alguna vez te has preguntado cómo funcionan los sistemas de archivos distribuidos?

- Los sistemas de archivos distribuidos son una forma de almacenamiento de datos que se distribuye en múltiples nodos dentro de una red. En lugar de tener un único servidor centralizado que almacena todos los archivos, los sistemas de archivos distribuidos dividen la carga de almacenamiento entre varios nodos, lo que los hace más eficientes y escalables.

Sistemas de archivos distribuidos (DFS)

En esencia, un sistema de archivos distribuido consiste en tres componentes principales:

- **Servidores de archivos:** Son los encargados de almacenar físicamente los datos.
- **Clientes:** Los dispositivos o aplicaciones que acceden a los archivos.
- **Red de comunicación:** El medio que conecta los servidores con los clientes.

La arquitectura de estos sistemas está diseñada para proporcionar transparencia. Esto significa que, para el usuario final, el proceso de acceder a un archivo en un sistema distribuido debería ser idéntico a acceder a un archivo local.

Sistemas de archivos distribuidos (DFS)

Para lograr esto, un dfs debe poseer las siguientes características:

- **Copia de respaldo y recuperación automática:** realizado como medida preventiva contra fallas del medio y errores de usuario.
- **Movilidad de usuario:** Permitir el uso de un entorno de trabajo independiente sin necesidad de dispositivos de almacenamiento secundario.
- **Transparencia de escalabilidad:** Los usuarios accedan a los archivos sin preocuparse por los nodos de almacenamiento o su ubicación.
- **Manejo de la concurrencia:** El sistema de archivos distribuido gestiona la concurrencia, permitiendo que múltiples usuarios accedan y modifiquen archivos simultáneamente de manera coordinada y consistente.
- **Manejo de fallos:** Proporciona mecanismos para la detección y recuperación de fallos.

Sistemas de archivos distribuidos (DFS)

Algunos que podríamos mencionar son:

- **Network File System (NFS):** Desarrollado originalmente por Sun Microsystems, es uno de los sistemas de archivos distribuidos más antiguos y ampliamente utilizados
- **Andrew File System (AFS):** Desarrollado en la Universidad Carnegie Mellon, se destaca por su capacidad para manejar grandes volúmenes de datos y usuarios.
- **GlusterFS:** GlusterFS es un sistema de archivos distribuido de código abierto que se destaca por su flexibilidad y escalabilidad.
- **Ceph:** Ceph es un sistema de almacenamiento distribuido unificado que proporciona almacenamiento de objetos, bloques y archivos en una sola plataforma.

Sistemas de archivos distribuidos Hadoop (HDFS)

El Hadoop Distributed File System (HDFS) es un sistema de archivos distribuido, escalable y portátil escrito en Java para el framework Hadoop. Este hace posible almacenar conjuntos de datos masivos con tipos de datos estructurados, semiestructurados y no estructurados. Está optimizado para almacenar grandes cantidades de datos y mantener varias copias para garantizar una alta disponibilidad y tolerancia a fallos.

Sistemas de archivos distribuidos Hadoop (HDFS)

- Los archivos se almacenan en bloques de 128 MB, y estos se distribuyen en los nodos que forman el clúster.
- Para conseguir una alta escalabilidad, se utiliza almacenamiento local que escala horizontalmente. Aumentar el espacio de almacenamiento supone añadir discos duros a nodos existentes o incorporar más nodos al sistema.
- En cuanto a la integridad de los datos, por defecto se almacenan 3 copias de cada bloque de datos. Aunque la replicación de datos no es necesaria para el funcionamiento de HDFS, almacenar solamente una copia podría suponer pérdida de datos por fallos o corrupción de archivos, comprometiendo la durabilidad del dato.
- Se trata de un sistema de archivos jerárquico; es decir, el usuario debe crear primero un directorio, dentro del cual se podrán crear, eliminar, mover o renombrar archivos.

Sistemas de archivos distribuidos Hadoop (HDFS)

La arquitectura de HDFS es de tipo maestro-esclavo. Esta basada en dos componentes principales: NameNodes y DataNodes.

- NameNode (NN): Es el maestro o nodo principal del sistema. Se encarga de gestionar su acceso y almacenar sus metadatos. Por tanto, necesita menos espacio de disco pero más recursos computacionales (memoria y CPU). Es el único nodo que conoce la lista de archivos y directorios del clúster.
- DataNode (DN): Corresponde con los nodos del clúster que almacenan los datos. Se encarga de gestionar el almacenamiento del nodo.

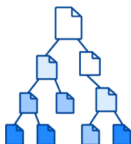
Bases de datos NoSQL

- NoSQL - 'Not only SQL'- es una categoría general de los sistemas de gestión de bases de datos que difiere de los RDBMS en diferentes modos:
 - Los datos almacenados no requieren estructuras fijas como tablas
 - Normalmente no soportan operaciones JOIN, ni garantizan completamente ACID (atomicidad, consistencia, aislamiento y durabilidad)
 - habitualmente escalan bien horizontalmente.
 - Los sistemas NoSQL se denominan a veces "no solo SQL" para subrayar el hecho de que también pueden soportar lenguajes de consulta de tipo SQL..

Bases de Datos NoSQL

Existen 4 tipos de bases de datos NoSQL

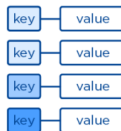
Document



Graph



Key-Value



Wide-column



Bases de datos NoSQL: Documentos

- Las bases de datos de documentos almacenan los datos como documentos.
- Pueden ser útiles en la gestión de datos semiestructurados y, por lo general, los datos se almacenan en los formatos JSON, XML o BSON.
- No es necesario que los esquemas de datos coincidan entre documentos.
- Ejemplos son MongoDB y CouchDB



Bases de datos NoSQL: Clave-Valor

- Almacena datos como un conjunto de pares clave-valor en los que una clave sirve como un identificador único..
- Tanto las claves como los valores pueden ser cualquier cosa, desde objetos simples hasta objetos compuestos complejos.
- Ejemplo: Redis



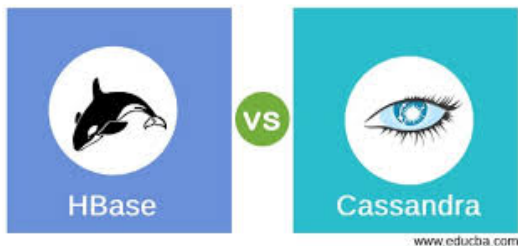
Bases de datos NoSQL: Grafos

- Aloja datos de un grafo de conocimiento.
- Los elementos de datos se almacenan como nodos, aristas y propiedades
- Cualquier objeto, lugar o persona puede ser un nodo.
- Una arista define la relación entre los nodos.
- Ejemplo: Neo4j



Bases de datos NoSQL: Columnas

- Almacenan información en columnas al contrario de las DDBB traicionales que los guardan por filas.
- Modelo tabular donde cada fila puede tener una configuracion diferente.
- Ejemplos son Cassandra y HBase



Teorema de CAP

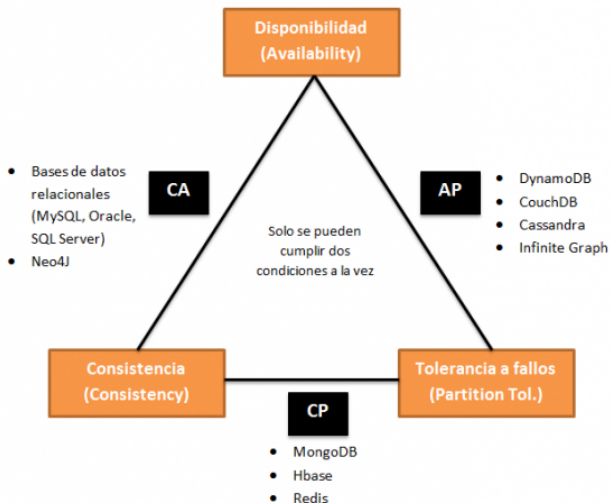
Teorema de Brewer: Es imposible para un sistema computacional distribuido ofrecer simultáneamente las siguientes 3 garantías:

- **Consistencia:** Todos los nodos ven los mismos datos al mismo tiempo.
- **Disponibilidad:** Garantizar que cada petición reciba respuesta sobre si tuvo éxito o no.
- **Tolerancia a la partición.** El sistema continúe funcionando a pesar de la pérdida de mensajes.

Equivalente a:

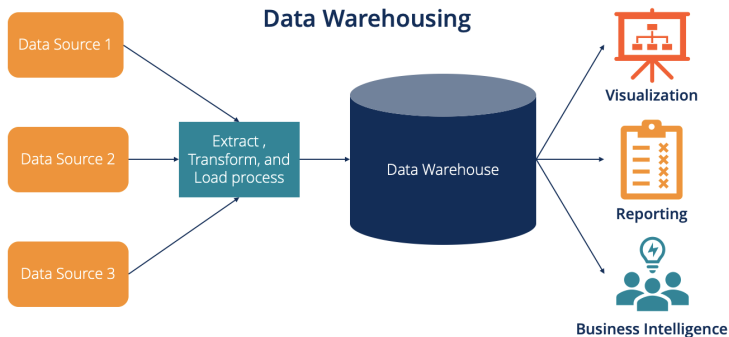
**You can have it good, you can have it fast, you can have it cheap:
Pick two.**

Teorema de CAP



Data Warehouses

Un almacén de datos o repositorio de datos (del inglés data warehouse) es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.



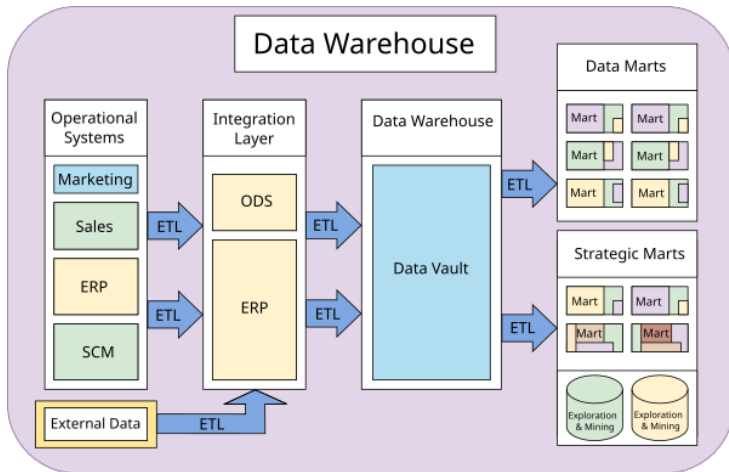
Data Warehouses

Según Bill Inmon, uno de los primeros autores en escribir sobre el tema, se define un data warehouse (repositorio de datos) en términos de las características del repositorio de datos:

- Orientado a temas: Los datos en la base de datos están organizados de manera que todos los elementos de datos relativos al mismo evento u objeto del mundo real queden unidos entre sí.
- Variante en el tiempo: Los cambios producidos en los datos a lo largo del tiempo quedan registrados para que los informes que se puedan generar reflejen en esas variaciones.
- No volátil: La información no se modifica ni se elimina, una vez almacenado un dato, este se convierte en información de sólo lectura, y se mantiene para futuras consultas.
- Integrado: La base de datos contiene los datos de todos los sistemas operacionales de la organización, y dichos datos deben ser consistentes.

Data Warehouse

Arquitectura



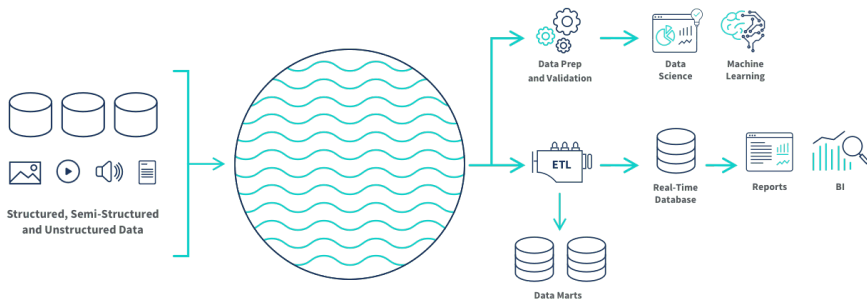
Data Marts

Es una versión específica del almacén de datos (data warehouse) centrados en un tema o un área. Son subconjuntos de datos con el propósito de ayudar a que un área específica pueda tomar mejores decisiones. Los datos existentes en este contexto pueden ser agrupados, explorados y propagados de múltiples formas para que diversos grupos de usuarios realicen la explotación de los mismos de la forma más conveniente según sus necesidades.

Data Lake

Un lago de datos (en inglés: data lake) es un sistema o biblioteca de datos almacenados en su formato bruto. Habitualmente, un lago de datos es un único repositorio de datos que incluye copias en bruto de datos del sistema, datos de sensores, datos sociales, etc. y datos transformados usados para tareas como generación de informes, visualización, analítica avanzada y aprendizaje automático. Un lago de datos puede incluir datos estructurados de bases de datos relacionales (filas y columnas), datos semiestructurados (CSV, logs, XML, JSON, etc.), datos sin estructurar (correos electrónicos, documentos, PDF, etc.) y datos binarios (imágenes, audio, vídeo, etc.).

Data Lake



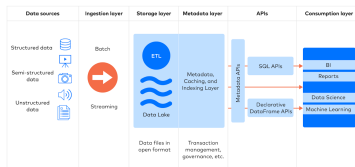
Data Lakehouse

Los almacenes de lago de datos (en inglés data lakehouses) son un enfoque híbrido que puede ingerir una variedad de formatos de datos en bruto como un lago de datos, pero proporcionar transacciones ACID y garantizar la calidad de los datos como un almacén de datos.

Según **Oracle**, los data lakehouses combinan el almacenamiento flexible de datos no estructurados de un lago de datos y las características y herramientas de gestión de los almacenes de datos.

Data Lakehouse

Data lakehouse architecture



Comparación entre lagos y almacenes de datos (1/3)

Característica	Almacenamiento de Datos	Lago de Datos
Estructura de datos	Datos estructurados con un esquema bien definido.	Almacena datos sin procesar, semiestructurados y estructurados.
Ingestión de datos	Implica un proceso ETL (Extraer, Transformar, Cargar) bien definido que estructura y limpia los datos antes de cargarlos en el almacén.	Permite la ingesta de datos en su forma cruda, sin necesidad inmediata de transformación. La transformación se puede aplicar según sea necesario.
Rendimiento	Optimizado para el rendimiento de consultas, a menudo utiliza técnicas como indexación y agregación previa para respuestas rápidas a consultas SQL.	Prioriza el almacenamiento de datos sobre el rendimiento de las consultas.

Comparación entre lagos y almacenes de datos (2/3)

Característica	Almacenamiento de Datos	Lago de Datos
Evolución del esquema	Los esquemas son relativamente estáticos y los cambios pueden requerir un esfuerzo y una planificación importantes.	Permite la lectura del esquema, lo que permite flexibilidad para acomodar cambios en los datos sin la necesidad de cambios de esquema iniciales.
Uso	Se utiliza principalmente para análisis de datos estructurados, inteligencia empresarial e informes.	Se utiliza para una amplia gama de análisis, incluidos análisis avanzados, ciencia de datos, aprendizaje automático y exploración de datos.

Comparación entre lagos y almacenes de datos (3/3)

Característica	Almacenamiento de Datos	Lago de Datos
Costo	Normalmente implica mayores costos de almacenamiento y consultas, ya que los datos a menudo se duplican e indexan para mejorar el rendimiento.	A menudo es rentable para almacenar grandes volúmenes de datos sin procesar, pero los costos pueden aumentar con el procesamiento y las transformaciones de los datos.
Calidad de los Datos	Enfatiza la calidad, coherencia y precisión de los datos, a menudo a través de prácticas estrictas de gobernanza de datos.	Ofrece flexibilidad y puede requerir esfuerzos adicionales para garantizar la calidad y coherencia de los datos.

Comparación entre lagos y almacenes de datos

DATA WAREHOUSE



 Dataedo /cartoon

DATA LAKE



Plotr@Dataedo

Almacenamiento en Memoria

Una base de datos en memoria (IMDb, o sistema de base de datos en memoria principal (MMDB) o base de datos residente en memoria) es un sistema de gestión de bases de datos que se basa principalmente en la memoria principal para el almacenamiento de datos informáticos. Se contrapone a los sistemas de gestión de bases de datos que emplean un mecanismo de almacenamiento en disco.

Almacenamiento en Memoria (Apache Spark)

- **Concepto:**

- Spark almacena datos en memoria RAM para acelerar el procesamiento.
- Utiliza **RDDs** (Resilient Distributed Datasets) para datos inmutables y distribuidos.
- **Persistir RDDs:** Mantiene los datos en memoria o en disco.
- **Niveles de Almacenamiento:** Desde memoria hasta disco con replicación.

- **Herramientas:**

- **DataFrames y Datasets:** API de alto nivel para manipulación de datos.

Almacenamiento de datos

Casos de ejemplo

Almacenamiento en Facebook

- **Desafío:**
 - Manejo de petabytes de datos generados diariamente por usuarios.
- **Soluciones:**
 - **Haystack:** Sistema de almacenamiento para fotos.
 - **Cassandra:** Para mensajería en tiempo real.
- **Herramientas:**
 - *Presto:* Motor de consultas SQL distribuido.
 - *Apache Hive:* Almacenamiento y análisis de big data.
- **Resultados:**
 - Alta disponibilidad y rendimiento.
 - Escalabilidad para soportar el crecimiento continuo.

Almacenamiento en Netflix

- **Desafío:**

- Streaming de video de alta calidad a nivel global.

- **Soluciones:**

- **Amazon S3:** Almacenamiento principal de contenido.
- **Open Connect:** Red propia de distribución de contenido (CDN).

- **Herramientas:**

- *Apache Kafka:* Procesamiento de logs y eventos en tiempo real.
- *AWS Lambda:* Para procesamiento serverless.

- **Resultados:**

- Entrega eficiente de contenido.
- Escalabilidad y disponibilidad global.

Almacenamiento en Twitter

- **Desafío:**

- Procesamiento en tiempo real de millones de tweets por minuto.

- **Soluciones:**

- **Hadoop:** Para almacenamiento y procesamiento batch.
- **Manhattan DB:** Sistema de base de datos interno para baja latencia.

- **Herramientas:**

- *Apache Storm:* Procesamiento de streams en tiempo real.
- *Apache Heron:* Sucesor de Storm para mejorar el rendimiento.

- **Resultados:**

- Procesamiento eficiente de datos en tiempo real.
- Mejora continua en la escalabilidad y rendimiento.

Gracias!!!

