Lección 5.1 Arquitecturas Big Data:

Hasta ahora hemos visto qué es big data, qué características tienen sus fuentes y las posibles aplicaciones para la mejora del negocio, en lo que se denomina Inteligencia de Negocio (o BI, Business Intelligence). A continuación veremos las arquitecturas y herramientas que han surgido para dar soporte a Big Data.

1. Necesidad de arquitecturas específicas para Big Data

Existen 3 necesidades básicas a la hora de definir una arquitectura específica de Big Data

- Las 5v's o características del Big Data hacen que las arquitecturas previamente existentes para el procesamiento y análisis de datos no sean adecuadas en Big Data
    - (Hasta ahora...) Almacenes de Datos (DW) sobre Bases de datos relacionales (ej. MySQL, Microsoft SQL Server, Oracle)
- El aumento del Volumen es uno de los principales responsables pero no el único
    - Las bases de datos relacionales distribuidas ya permiten el procesamiento de grandes volúmenes de datos. Ej. Microsoft SQL Server hasta 524 terabytes
- (Junto al Volumen) La Variedad y la Velocidad son los dos otros grandes problemas que motivan la aparición de nuevas arquitecturas para dar soporte a Big Data

2. Necesidad de arquitecturas específicas para Big Data

- La Variedad implica:
    - Soporte para el procesamiento **eficiente** de la nuevas fuentes semi estructuradas y no estructuradas, permitiendo su integración con las estructuradas
    - Escalabilidad ante la aparición de nuevos tipos de fuentes
- La Velocidad implica:
    - Necesidad de dar soporte a la adquisición y procesamiento de datos Streaming, (Recordemos: generados y distribuidos de forma continua) además de por lotes (batch processing, vimos ambos tipos de fuentes en la lección 3.3 del módulo 3 cuando hablamos de velocidad)
    - En algunos casos se requiere que el **procesamiento y análisis** de grandes volúmenes de datos, de cualquier nivel de estructura, se lleve a cabo en **tiempo**

real (Incluso cuando son transmitidos en Streaming) → Implica una elevada potencia de procesamiento

- Para dar soporte a los nuevos requisitos surgen tres nuevos tipo de arquitecturas
    - Hadoop / MapReduce
    - Bases de datos NoSQL
    - SGBDR Extendidos

3. Hadoop / MapReduce

Una de las arquitecturas más usadas por las empresas y desarrolladores que implementan Big Data es...

- **Apache Hadoop** es un entorno de **código abierto** que en su forma más básica implementa Hadoop **HDFS** y el algoritmo **Map Reduce**
    - Hadoop HDFS es el sistema de archivos distribuido que conforma el corazon de la arquitectura
    - Map Reduce: Modelo de programación para el procesamiento de datos en paralelo el cual es simple pero con una gran potencia.
- Esta arquitectura está pensada para ser instalado en un cluster
    - Lo que nos proporciona que sea fácilmente escalable en cuanto al volumen de almacenamiento y capacidad de procesamiento manteniendo un coste bajo y de crecimiento lineal
    - La combinación HDFS / MapReduce permite procesar cientos de Gigabytes de datos, e incluso Terabytes, en menos de un minuto (es muy eficiente)
- Sobre el núcleo HDFS / MapReduce, se han desarrollado diversas herramientas y lenguajes de programación para facilitar la gestión y análisis de Big Data
    - ej. Apache Flume, Apache Pig, Apache Cassandra, Apache Hive....

(En las siguientes lecciones, estudiaremos en profundidad este entorno y las distintas herramientas que se han desarrollado para el mismo)

4. Bases de datos NoSQL

Además del entorno Hadoop en los últimos años ha aparecido un nuevo tipo de bases de datos para dar soporte al Big Data: Las Bases de Datos NoSQL

- **BD NoSQL**: Bases de datos que rompen una o más reglas de las bases de datos relacionales (teoría del modelo relacional) para dar soporte a la nuevas características introducidas por Big Data
    - (Por lo general) Destinadas al almacenamiento de información no relacional (fuentes semi estructuradas y no estructuradas) por lo que no implementan el lenguaje de consulta SQL. (Salvo excepciones como Apache Hive o Google Big Query)
    - No requieren la normalización de los datos[1]
    - Relajan las restricciones ACID (Atomicity, Consistency, Isolation, Durability) (Remitir a la teoría de Bases Datos, poner algun enlace en documento adicional, a wikipedia por ejemplo o referencia a libro de Codd)
- Estas bases de datos están optimizadas para las lecturas y escrituras aleatorias
    - A diferencia de MapReduce sobre HDFS en el entorno Hadoop, optimizado para la lectura secuencial de grandes volúmenes de datos para tareas de análisis
    - (Esto significa que son…) Adecuadas para su uso con aplicaciones que requieran almacenar y procesar datos de fuentes Big Data **para tareas distintas del análisis**. Ej. Gestión de datos de tipo semi estructurados y no estructurados generados y usados en una aplicación web. Un posible ejemplo es una red social o aplicaciones de tipo geográficas.
- Existen distintos de tipos e implementaciones de BD NoSQL:
    - Clave-Valor (Apache Cassandra), modelo de columnas (Apache Hbase), documentales (Mongo DB) , gráfos (Neo4j)...
    - El entorno Hadoop puede ser considerado en sí mismo como un entorno para la ejecución de distintos tipos de BD's NoSQL

Veremos estas arquitecturas más en detalle en la lección 6. (Los tipos los comentamos en esa lección)

5. SGBDR Extendidos

Además de las bases de datos NoSQL y el entorno Hadoop existe un tercer tipo de arquitecturas para dar soporte a Big Data

[1] Recordemos que las bases de datos se normalizan para:
- Evitar la redundancia de los datos.
- Disminuir problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

- Sistemas de Gestión de Bases de Datos (SGBDR) Extendidos: Evolución de la tecnología de bases de datos relacionales para dar soporte al procesamiento y análisis de Big Data
    - Añaden nuevos tipos de datos que permiten almacenar en un campo (columna hablando más técnicamente) información semi estructurada y, en algunos casos, sin estructura alguna.
    - Soporte para definición de complejas funciones para análisis estadísticos (UDF, User Defined Function) que se ejecutan lo más cerca posible de los datos (Al igual que es posible en Hadoop como veremos en posteriores lecciones donde explicaremos con mayor detalle el concepto de UDF)
- Además esta arquitectura implementa Bases de datos distribuidas y procesamiento MPP (Massively Parallel Processing)
    - Procesamiento muy eficiente a cambio de un alto coste de licencia y menor escalabilidad frente a Big Data que soluciones como el entorno Hadoop
- Algunos ejemplos: Vertica de HP o Greemplun de EMC