Resumen Tema 1: Big Data y Complejidad Computacional

Concepto de Big Data:

El Big Data se refiere a la enorme cantidad de datos generada cada segundo por múltiples fuentes, como redes sociales, dispositivos IoT, plataformas de transmisión, empresas y gobiernos. Estos datos, que se miden en Zettabytes (ZB), están aumentando a una velocidad sin precedentes, y aunque representan solo una fracción de la información teórica posible, el volumen gestionado es impresionante en comparación con el pasado. Es fundamental que las empresas cuenten con infraestructura de alto rendimiento para procesar estos datos en un tiempo razonable, ya que la generación de datos es constante y demanda capacidad de respuesta rápida. Existen varios aspectos críticos en Big Data, conocidos como las 'V's': Valor, Veracidad, Viabilidad, Visualización, Vulnerabilidad.

Modelos de negocio en Big Data:

Usuarios de datos: Empresas que utilizan los datos para mejorar sus procesos y productos, aplicando Big Data para capturar valor internamente. Por ejemplo, tiendas en línea que analizan el comportamiento de compra para personalizar recomendaciones a clientes. Proveedoras de datos: Empresas que recopilan, empaquetan y venden datos a terceros, generando valor al agregar información de varias fuentes. Ejemplo: compañías que venden información de salud de dispositivos como relojes inteligentes a aseguradoras. Facilitadores de Big Data: Compañías que proveen infraestructura y servicios tecnológicos que permiten a otras implementar soluciones de Big Data sin trabajar directamente con los datos. Ejemplo: proveedores de infraestructura en la nube como Amazon Web Services o Microsoft Azure.

Complejidad computacional:

A medida que el volumen de datos se duplica anualmente, la capacidad de procesamiento de los sistemas sigue la Ley de Moore, donde la potencia de los procesadores se duplica cada 18 meses.

Sin embargo, esto no es suficiente para procesar la cantidad de datos que crece a mayor velocidad. La complejidad de los algoritmos también es un factor importante. En lugar de medir su coste computacional en función del hardware, se mide en términos del número de operaciones necesarias. La notación Big O (O(n^2), por ejemplo) se usa para expresar la complejidad y crecimiento de un algoritmo a medida que el tamaño de los datos (n) crece. Existen dos tipos de problemas complejos: P (tiempo polinomial): Problemas que pueden resolverse en tiempo razonable. NP (tiempo no determinista polinomial): Problemas para los cuales no existe un algoritmo polinomial conocido, aunque su solución es verificable en tiempo polinomial. Dentro de NP, los problemas NP-completos son los más complejos y no se ha probado que puedan resolverse eficientemente.

Escalabilidad y algoritmos:

El algoritmo RSA, que se basa en la dificultad de factorizar números grandes, ilustra los retos de la escalabilidad. Factorizar números suficientemente grandes para romper la seguridad RSA requiere una capacidad computacional extraordinaria. Un ejemplo con el supercomputador IBM Roadrunner muestra cómo incluso los sistemas más avanzados tienen limitaciones: aunque fue capaz de realizar 1000 billones de operaciones por segundo, factorizar números de más de 50 dígitos en tiempo razonable sigue siendo muy difícil. Este ejemplo subraya que la escalabilidad depende tanto de la potencia de la máquina como de la complejidad del algoritmo.

Computación distribuida:

La computación distribuida permite dividir y ejecutar tareas entre múltiples computadoras en distintas ubicaciones, trabajando de forma coordinada para resolver problemas o ejecutar tareas. Este modelo permite manejar mayores volúmenes de datos al dividirlos en paquetes más pequeños que pueden procesarse más eficientemente en varios nodos. Un ejemplo del modelo MapReduce divide el procesamiento de datos en dos fases: Map (Mapeo): Los datos se distribuyen en diferentes nodos, que trabajan en sus propios subconjuntos y generan resultados intermedios.

Reduce (Reducción): Los resultados de cada nodo se combinan para obtener un resultado final. Esta técnica es usada en sistemas Big Data como Hadoop para manejar grandes volúmenes de datos distribuidos. La computación distribuida permite mejorar significativamente la eficiencia en cálculos, como el cálculo de medias, al reducir el número total de operaciones gracias a la paralelización de tareas.