Big Data no es simplemente un gran volumen de datos.
Aquí, la palabra Grande se refiere a un gran alcance de datos.
Se describe con **tres palabras que comienzan con la letra V: volumen, velocidad y variedad**. Pero el mundo analítico y de la ciencia de datos ha visto datos variando en otras dimensiones **además del fundamento tres V de big data, como veracidad, variabilidad, volatilidad, visualización y valor.**

Las diferentes Vs mencionadas hasta ahora se explican de la siguiente manera:
Volumen:
Se refiere a la **cantidad de datos generados en segundos**. 90% de los datos del mundo actual se han creado en los últimos dos años. Desde esa vez, los datos en el mundo se duplican cada dos años. Tan grande los volúmenes de datos son generados principalmente por máquinas, redes, redes sociales medios y sensores, incluidos los estructurados, semiestructurados y datos no estructurados.

Velocidad: Se refiere a la **velocidad a la que se generan los datos**, almacenado, analizado y movido. Con la disponibilidad de dispositivos conectados a Internet, las máquinas y sensores inalámbricos o cableados pueden transmitir sus datos tan pronto como se crean. Esto conduce a la transmisión de datos en tiempo real y ayuda a las empresas a tomar decisiones valiosas y rápidas.

Variedad: Esto se refiere a los **diferentes formatos de data**.Datos que solían ser almacenados en los formatos .txt, .csv y .dat de orígenes de datos como sistemas de archivos, hojas de cálculo y bases de datos. Este tipo de datos, que residen en un campo fijo dentro de un registro o archivo, se denomina datos estructurados. Hoy en día, los datos no siempre están en el formato estructurado tradicional.

Hoy en día, los datos no siempre están en el formato estructurado tradicional. El También se generan nuevas formas de datos semiestructurados o no estructurados por diversos métodos, como correo electrónico, fotos, audio, video, PDF, SMS, o incluso algo de lo que no tenemos ni idea. Estas variedades de datos los formatos crean problemas para almacenar y analizar datos. Este es uno de los principales desafíos que debemos superar en el dominio de big data.

Veracidad: Esto se refiere a la **calidad de los datos, como la confiabilidad**, sesgos, ruido y anormalidad en los datos. Los datos dañados son bastante normales. Podría originarse debido a una serie de razones, como errores tipográficos, falta o abreviaturas poco comunes, reprocesamiento de datos y fallas del sistema.

Sin embargo, ignorar estos **datos maliciosos podría conducir a datos inexactos** para el análisis y eventualmente a una decisión equivocada. Por lo tanto, asegúrese de que los datos son correctos en términos de audición de datos y la corrección es muy importante para el análisis de big data.

Variabilidad: Se refiere al **cambio de los datos**. Significa que la data podría tener diferentes significados en diferentes contextos. Esto es particularmente importante cuando se lleva a cabo un análisis de sentimientos(sentiment analysis). Los algoritmos de análisis son capaces de comprender el contexto y descubrir el significado exacto y valores de los datos en ese contexto.

Volatilidad: Se refiere a **cuánto tiempo los datos son válidos y almacenados**. Esto es particularmente importante para el análisis en tiempo real. Requiere un tiempo objetivo a determinar para que los analistas puedan centrarse en preguntas particulares y obtener un buen rendimiento del análisis.

Visualización: Esto se refiere a **la forma de hacer que los datos sean comprendidos**. La visualización no solo significa gráficos ordinarios o gráficos circulares; también hace que grandes cantidades de datos sean comprensibles en una vista multidimensional que es fácil de entender. La visualización es una forma innovadora de mostrar cambios en los datos. Requiere mucha interacción, conversaciones y esfuerzos conjuntos entre analistas de big data y expertos en el dominio de negocios para hacer que la visualización sea significativa.

Valor: Esto se refiere al **conocimiento obtenido del análisis de datos** en big data. El valor del big data se encuentra en cómo las organizaciones se convierten en empresas impulsadas por big data y como utilizan la información del análisis de big data para su toma de decisiones.

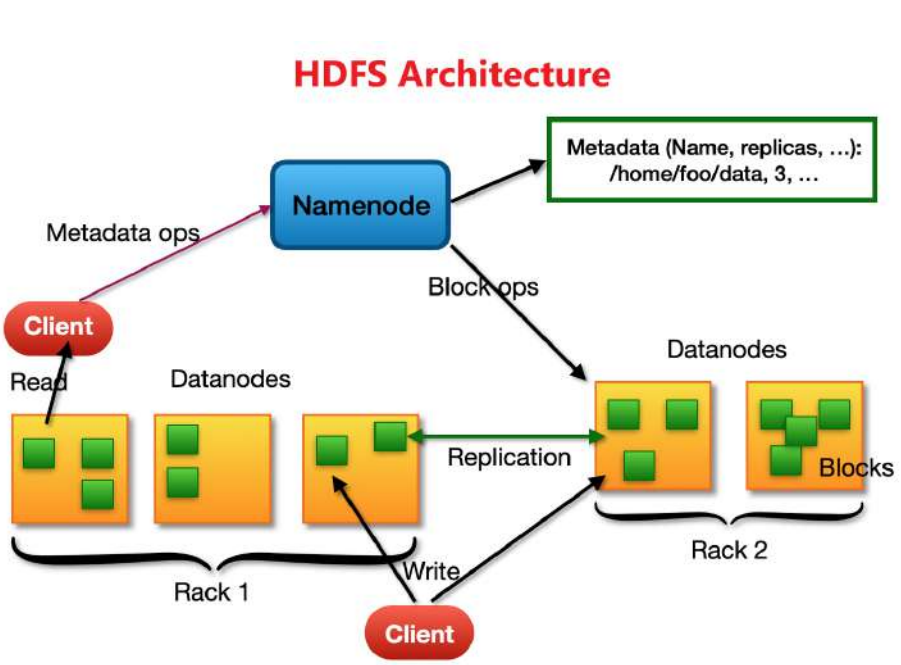
Daemon:

En los entornos UNIX o GNU/**Linux**, se denomina «**demonio**» o «daemon» a un programa no interactivo (es decir, que el usuario no puede controlar directamente) que **se encarga de procesos del sistema en un segundo plano**.

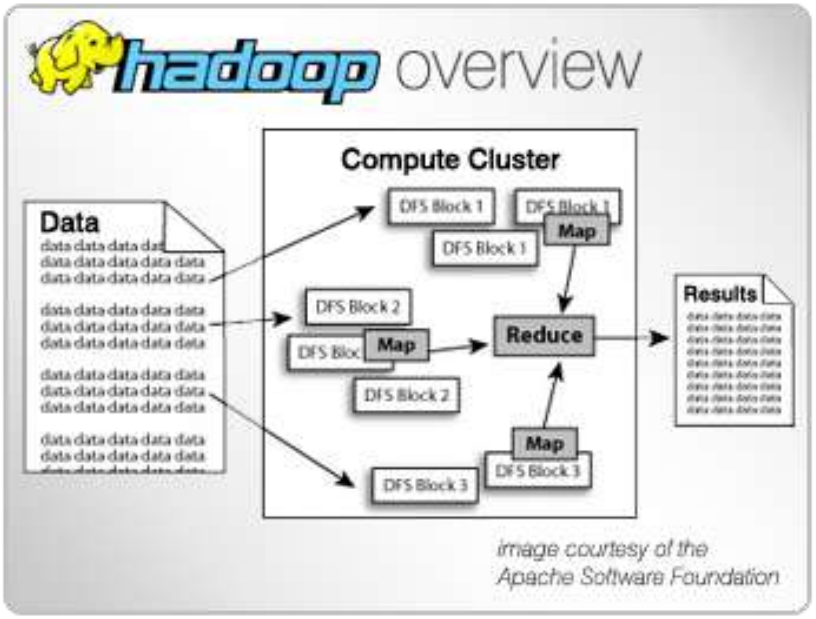
En términos de informática, Background, fondo o segundo plano se utiliza para nombrar a todos aquellos procesos o rutinas de ejecución que se realizan en segundo plano. Esto implica que el proceso se está llevando a cabo con una prioridad baja y no siempre tiene la CPU de forma secuencial ejecutando su código.

H**DFS (Hadoop Distributed File System) es el componente principal del ecosistema Hadoop**. Esta pieza hace posible almacenar data sets masivos con tipos de datos estructurados, semi-estructurados y no estructurados como imágenes, vídeo, datos de sensores, etc.

HDFS: The Hadoop Distributed File System particiona la data y la almacena a través de sus nodos cluster.
HDFS es utilizado para almacenar masivas cantidades de datos sobre un ambiente distribuido. HDFS almacena información de metadata desde el archivo y la data por separado. La data almacenada en el HDFS es escrito una vez pero leida multiples veces. Brinda una base para otras herramientas, como Hive, Pig, HBase y MapReduce para procesar la data.



● **YARN**:
Fue introducido en hadoop 2 y esta disponible en versiones más avanzadas.
Desacopla las funcionalidades de manejo de recursos y trabaja scheduling/monitoring en daemons separados.
Hadoop
Hadoop fue lanzado por primera vez por Apache en 20 11 como la versión 1.0.0, que sólo contenía HDFS y MapReduce.



Hadoop fue diseñado como una plataforma de computación (MapReduce) y almacenamiento (HDFS) desde el inicio. Con la creciente necesidad de análisis de big data, Hadoop atrae muchos otros programas para resolver preguntas de big data y se fusionan en un Ecosistema de big data centrado en Hadoop. El siguiente diagrama ofrece un resumen Descripción general del ecosistema de big data de Hadoop en la pila Apache

Apache Hadoop ecosystem

¿Por qué es importante Hadoop?

Capacidad de almacenar y procesar enormes cantidades de cualquier tipo de datos, al instante. Con el incremento constante de los volúmenes y variedades de datos, en especial provenientes de medios sociales y la Internet de las Cosas (IoT), ésa es una consideración importante.

Poder de cómputo. El modelo de cómputo distribuido de Hadoop procesa big data a gran velocidad. Cuantos más nodos de cómputo utiliza usted, mayor poder de procesamiento tiene.

Tolerancia a fallos. El procesamiento de datos y aplicaciones está protegido contra fallos del hardware. Si falla un nodo, los trabajos son redirigidos automáticamente a otros modos para asegurarse de que no falle el procesamiento distribuido. Se almacenan múltiples copias de todos los datos de manera automática.

Flexibilidad. A diferencia de las bases de datos relacionales, no tiene que procesar previamente los datos antes de almacenarlos. Puede almacenar tantos datos como desee y decidir cómo utilizarlos más tarde. Eso incluye datos no estructurados como texto, imágenes y videos.

Bajo costo. La estructura de código abierto es gratuita y emplea hardware comercial para almacenar grandes cantidades de datos.

Escalabilidad. Puede hacer crecer fácilmente su sistema para que procese más datos son sólo agregar nodos. Se requiere poca administración.

MapReduce:

MapReduce es un framework en Hadoop para procesar una cantidad masiva de

datos en paralelo en un entorno distribuido. Es un sistema fiable y tolerante a fallos proceso que se ejecuta en hardware común, lo que lo hace rentable.
El HDFS almacena archivos en bloques idénticos de tamaño fijo (como 64 MB o 128 MB). El framework de MapReduce accede y procesa estos bloques en un entorno paralelo distribuido.
MapReduce distribuye entre estos bloques de tamaño fijo para ejecutar en paralelo y luego agregarán la salida a escribir en almacenamiento externo, como HDFS o AWS S3. El MapReduce framework funciona en pares clave-valor; tiene dos partes principales, el Mapper y Reductor.

Es un componente clave del procesamiento masivo de cantidades de datos en paralelo. Brinda mecanismos para manejar grandes datasets como el batch de una manera confiable, disponible y con ambientes tolerantes a fallas.
MapReduce separa los datos en partes independientes, las cuales son procesadas en paralelo por el map task y <Key, value> a un reducer que los agrega antes de almacenarlos en el HDFS.
Hadoop map reduce no es sencillo para trabajar, y tampoco es partidario de la reusabilidad. Un desarrollador puede pasar horas construyendo programas mapReduce para analizar datasets directamente, que impacta la productividad.
Considerando estos desafios, Facebook creo Hives para abordar la productividad de la complejidad de MapReduce.

HiveQL toma menos tiempo para procesar y analizar la data HDFS.
● Hive traduce el lenguaje HiveQl en un codigo MapReduce
● Hive almacena la información schema, como los nombres de las tablas, los nombres de las columnas, tipo de datos e información de las particiones.
●Hive puede dar soporte a muchos formatos de archivos y a formatos de datos.

	RDBMS	Hive
Language	PL/SQL	Open and read a file
CRUD	Insert, update, and delete	Insert overwrite; no update or Delete(***available with Hive Transaction in Hadoop3***)
CRUD Transaction	Yes	Yes (optional configuration in Hadoop 3)
Latency	Seconds, milliseconds	Minute or more
Indexes	Any number of indexes, very important for performance	No indexes, data is always scanned (in parallel). However, defined partitioned can improve the performance.
Data Size	TBs	PBs

HIVE QUERY/ CONSULTAS HIVE
Las consultas Hive (Hive QL) se comportan como SQL. Soporta consultas SQL como select, joins (como inner join, left, outer join, and right outer join), grupos cartesianos por, y uniendo todo. Soporta también varios comandos parecidos a los DBMS, tales como: mostrar las tablas, crear tablas y describir tablas.
DATA STORAGE/Almacenamiento de datos

Las tablas en Hive están asociadas con la tabla de datos para los directorios HDFS y mapean los directorios HDFS de datos.
● Table: A logical structure in Hive maps the data from HDFS directories.
● Partition: A table's partition is stored in sub-directories in HDFS directories.
● Buckets: A bucket is stored in a file within the partitioned table's directory.

Table 8.3 Complex Hive Data Types

Type	Size	Example
ARRAY	Ordered sequence of same type access by indexing ARRAY<data_type>	["red","yellow","green"]
MAP	A key-value collection. MAP<primitive_type, data_type>	{1:"red",2:"yellow"}
UNION	UNIONTYPE<data_type, data_type, ...> Available from Hive 0.870	{1:["red","orange"]}
STRUCT	Similar to JSON type object and field can be accessed by dot. STRUCT<col_name : data_type [COMMENT col_comment], ...> {1, "red"}	

Antes de Hive, las organizaciones utilizaban RDBMS-basadas en warehouse data para analizar y procesar grandes datasets. Pero cuando la data empezó a crecer de gigabytes a terabytes y luego a petabytes, era inadecuado manejar tal crecimiento en los datasets. Como resultado Facebook empezó utilizando Hadoop, que mejoró el desempeño del procesamiento. No solo no era menos costoso sino que mejoró el rendimiento
Hive brinda conceptos familiares de tablas, columnas y particiones y un subconjunto de SQL para la data no estructurada mientras mantiene la flexibilidad de Hadoop.
Hive fue lanzado en el 2010, utilizado por amazon, ibm, yahoo, netflix y FINRA (Financial Industry Regulatory Authority).Inicialmente utilizaba mapreduce como herramienta de ejecución pero luego Tez y herramienta Spark.

Descripción general de Hive
Hive es un estándar para consultas SQL sobre petabytes de datos en Hadoop. Eso proporciona acceso similar a SQL a los datos en HDFS, lo que permite que Hadoop se utilice como un
almacén de datos. El lenguaje de consulta de Hive (HQL) tiene una semántica similar y funciona como SQL estándar en la base de datos relacional, de modo que
Los analistas de bases de datos experimentados pueden tenerlo fácilmente en sus manos. Consulta de Hive
el lenguaje puede ejecutarse en diferentes motores informáticos, como MapReduce, Tez, y Spark.
La estructura de metadatos de Hive proporciona una estructura de alto nivel similar a una tabla en la parte superior
de HDFS. Admite tres estructuras de datos principales, tablas, particiones y Cubos. Las tablas corresponden a directorios HDFS y se pueden dividir en particiones, donde los archivos de datos se pueden dividir en buckets. Metadatos de Hive la estructura suele ser el esquema del concepto Schema-on-Read en Hadoop,

lo que significa que no es necesario definir el esquema en Hive antes de almacenar datos en HDFS.
La estructura de metadatos de Hive proporciona una estructura de alto nivel similar a una tabla en la parte superior
de HDFS. Admite tres estructuras de datos principales, tablas, particiones y buckets.

Las tablas corresponden a directorios HDFS y se pueden dividir en particiones, donde los archivos de datos se pueden dividir en buckets.
La estructura de Metadatos de Hive suele ser el esquema del concepto Schema-on-Read en Hadoop,lo que significa que no tiene que definir el esquema en Hive antes de almacenar datos en HDFS. Aplicar metadatos de Hive después de almacenar datos trae más flexibilidad y eficiencia a su trabajo de datos. La popularidad de los metadatos de Hive lo convierten en la forma de facto de describir big data y es utilizado por muchos herramientas en el ecosistema de big data.

Hive proporciona un modelo de consulta simple y optimizado con menos codificación que MapReduce HQL y SQL tienen una sintaxis similar
El tiempo de respuesta de la consulta de Hive suele ser mucho más rápido que otros en el mismo volumen de grandes conjuntos de datos
Hive admite la ejecución en diferentes marcos informáticos
Hive admite la consulta ad hoc de datos en HDFS y HBase
Hive admite funciones, scripts y procedimientos java/scala definidos por el usuario idiomas para ampliar su funcionalidad

```
Create a table using various data types (> indicates the beeline
interactive mode):
 > CREATE TABLE employee (
 > name STRING,
 > work_place ARRAY<STRING>,
 > gender_age STRUCT<gender:STRING,age:INT>,
 > skills_score MAP<STRING,INT>,
 > depart_title MAP<STRING,ARRAY<STRING>>
 > )
 > ROW FORMAT DELIMITED
 > FIELDS TERMINATED BY '|'
 > COLLECTION ITEMS TERMINATED BY ','
 > MAP KEYS TERMINATED BY ':'
 > STORED AS TEXTFILE;
 No rows affected (0.149 seconds)
```

Hue:

Para simplificar el proceso de creación, mantenimiento y ejecución de muchos tipos de trabajos de Hadoop, Hue (Hadoop User Experience) ofrece una GUI web para los usuarios de Hadoop. Básicamente, se compone de varias aplicaciones que interactúan con los componentes de Hadoop, y también tiene un SDK abierto para permitir la creación de nuevas aplicaciones.

Bibliografía:

1. *Ken Cochrane, Jeeva S. Chelladhurai, and Neependra K. Khare. 2018. Docker Cookbook: Over 100 practical and insightful recipes to build distributed applications with Docker, 2nd Edition (2nd. ed.). Packt Publishing.*
2. *Kumar, N. 2021 ,Big Data Using Hadoop and Hive, 9781683926450 2021 Mercury Learning & Information*
3. *Du, D., Apache Hive Essentials 9781782175056,2015 Packt Publishing, Limited.*