
Universidad
Tecmilenio®

Universidad Tecmilenio

Infraestructura para Big Data

Hadoop HDFS: almacenamiento de datos y distribución

Semana 11

Introducción

HDFS (Hadoop Distributed File System) es uno de los componentes esenciales de Apache Hadoop.

Algunos de sus objetivos son acceder a datos de transmisión, alojar grandes cantidades de datos y lograr una recuperación rápida ante fallas en el hardware.

Hadoop se utiliza en empresas de diferentes rubros, de comercio electrónico, desarrollo de tecnologías de la información, plataformas de redes sociales, servicios de alojamiento web, sitios de búsqueda, entre otros.

Esta unidad servirá de complemento a la información descrita en temas anteriores.

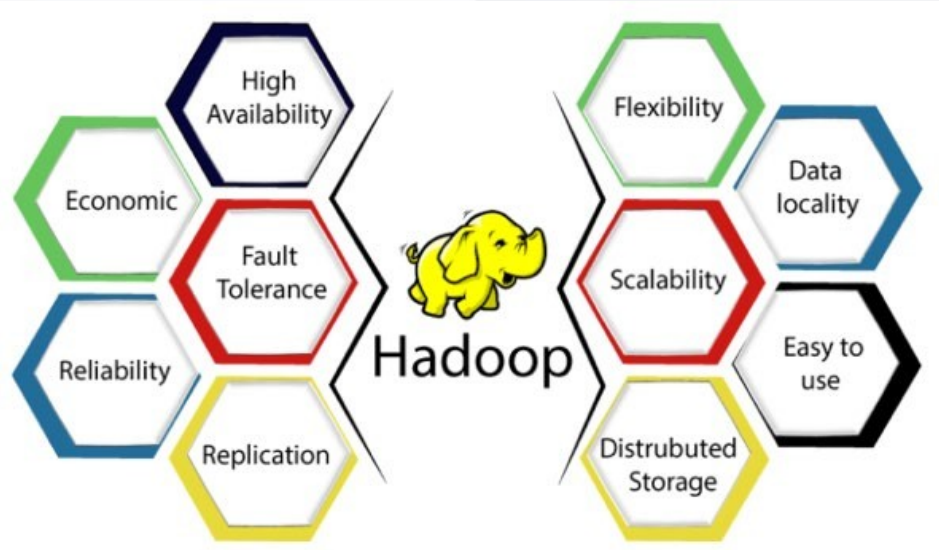
Explicación

HDFS es uno de los componentes principales de Hadoop. Entre las características más sobresalientes en cuanto a almacenamiento de datos se establecen las siguientes (Apache Hadoop, 2021):

1. HDFS tiene un enfoque de escritura único y de muchas lecturas para sus aplicaciones y archivos. No se pueden editar los archivos, pero sí agregar contenido al final de estos.
2. Un objetivo central de HDFS es la detección de fallas en el hardware y la recuperación automática y rápida de estas.
3. Replicación de datos: la cual es una característica sobresaliente de HDFS, permitiendo resolver el problema de la pérdida de los datos al momento de una falla en el hardware.
4. El procesamiento de los datos donde están almacenados, en lugar de moverlos, permitiendo que el rendimiento del sistema aumente y la congestión de la red se logre reducir.
5. Su gran portabilidad a través de plataformas heterogéneas de software y hardware, permitiendo que pueda utilizarse para una gran cantidad de aplicaciones.
6. Escalabilidad para procesar y guardar grandes cantidades de datos: los archivos promedio de HDFS cuentan con un tamaño desde gigabytes hasta terabytes, proporcionando así un almacenamiento de datos confiable.

Explicación

Características HDFS



Fuente: Swarupa, P. (2020). *Features of HDFS (Hadoop Distributed File System).* Recuperado de https://medium.com/@swarupachowdaryp

Replicación de los datos

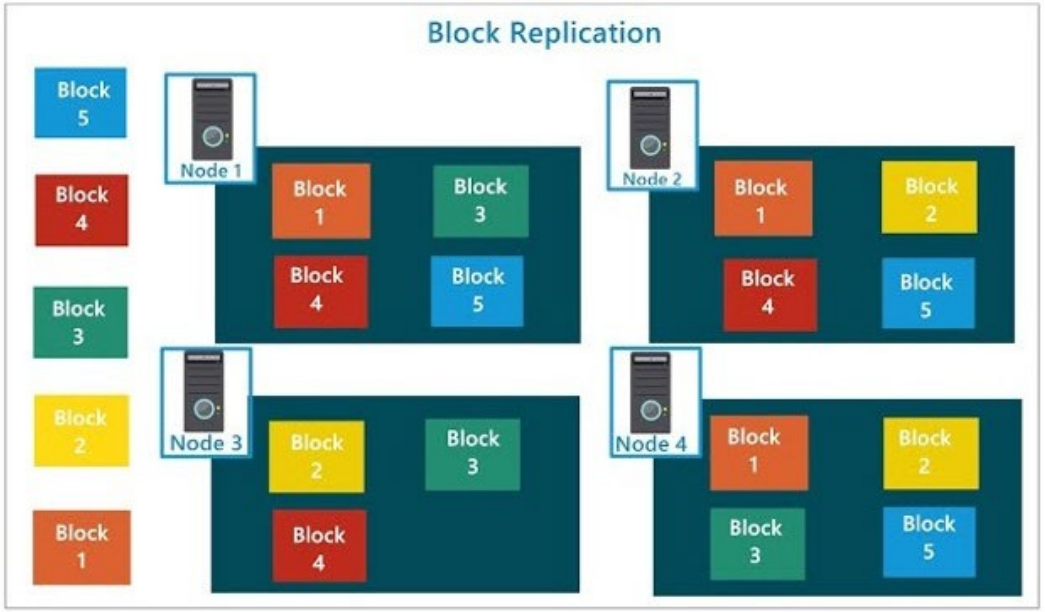
La falla en los sistemas es una posibilidad latente, por lo que HDFS está diseñado para guardar archivos de gran tamaño de forma confiable en máquinas en un clúster grande, realizando una replicación de los ficheros existentes para soportar la tolerancia a las fallas.

Se puede precisar el número de las replicaciones de un archivo cuando es creado (y este parámetro se puede ajustar de forma posterior). Los archivos en HDFS son de una sola escritura y tienen estrictamente un solo escritor en cualquier momento (Apache Hadoop, 2019).

Explicación

NameNode es el encargado de tomar todas las decisiones relacionadas con la replicación de bloques. Este recibe regularmente un Heartbeat y un Blockreport de cada uno de los DataNodes del clúster.

La recepción de un Heartbeat trae consigo que el DataNode se desempeñe de forma adecuada (de igual forma que un latido del corazón nos indica que este funciona). Un Blockreport abarca una lista de todos los bloques en un DataNode.



Fuente: Software Testing Help. (2020). *Hadoop HDFS – Hadoop Distributed File System*. Recuperado de https://www.softwaretestinghelp.com/hadoop-distributed-file-system/

Para conocer un ejemplo de agrupación de datos en Python, ve el siguiente video:

Dirección de Producción de Contenidos. (2021, 27 de enero). *Big Data Tema 6* [Archivo de video]. Recuperado de https://youtu.be/jm0AAatWZbA

Explicación

Para introducir la cuestión de almacenamiento interno compartido en arquitecturas Hadoop, debemos centrarnos en el núcleo de este *framework* diseñado para operar en clave de Big Data. Su composición consta de:

Implementación Map/Reduce o capa de procesamiento:

Map/Reduce procesa grandes cantidades de información de un modo sencillo para el usuario, gracias al almacenamiento interno compartido, muy fácil de utilizar, ya que la complejidad está oculta a los usuarios.

Capa de almacenamiento HDFS:

Es un sistema de archivos distribuido, desarrollado en Java por Doug Cutting, que constituye la capa de almacenamiento en un clúster Hadoop.

Cierre

Apache Hadoop se enfrenta con éxito a complejidades de alto volumen, velocidad y variedad de los datos, permitiendo cargar, analizar y almacenar petabytes de información mediante el análisis por lotes y el procesamiento distribuido.

Bibliografía

- Apache Hadoop. (2021). *HDFS Architecture*. Recuperado de https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html
- Software Testing Help. (2020). *Hadoop HDFS – Hadoop Distributed File System*. Recuperado de https://www.softwaretestinghelp.com/hadoop-distributed-file-system/
- Swarupa, P. (2020). *Features of HDFS (Hadoop Distributed File System)*. Recuperado de https://swarupachowdaryp.medium.com/features-of-hdfs-hadoop-distributed-file-system-6d7cd8712dfb

Universidad Tecmilenio

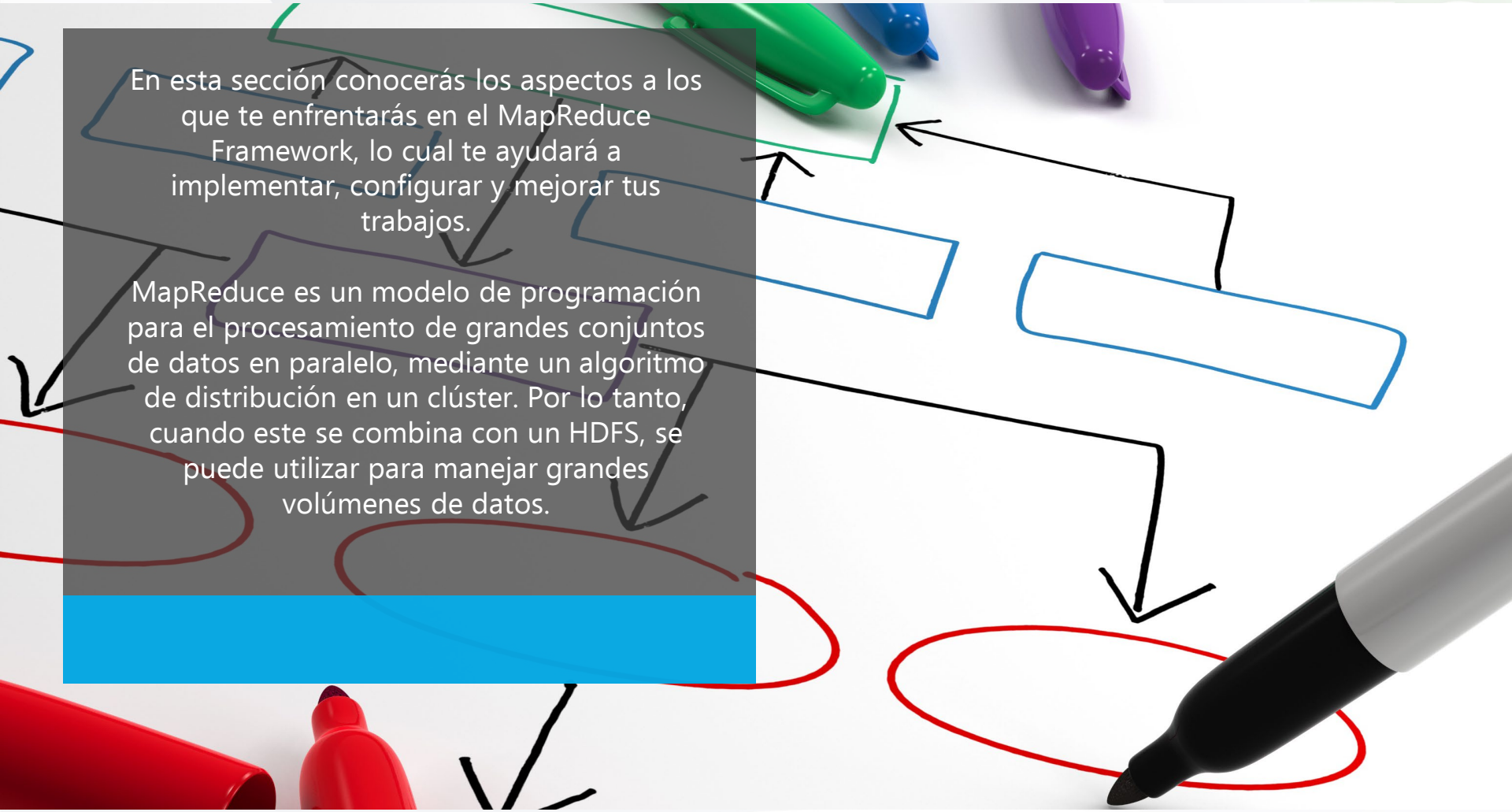
Infraestructura para Big Data

Fundamentos de MapReduce (Hadoop File System)

Semana 11

Introducción

Universidad Tecmilenio



En esta sección conocerás los aspectos a los que te enfrentarás en el MapReduce Framework, lo cual te ayudará a implementar, configurar y mejorar tus trabajos.

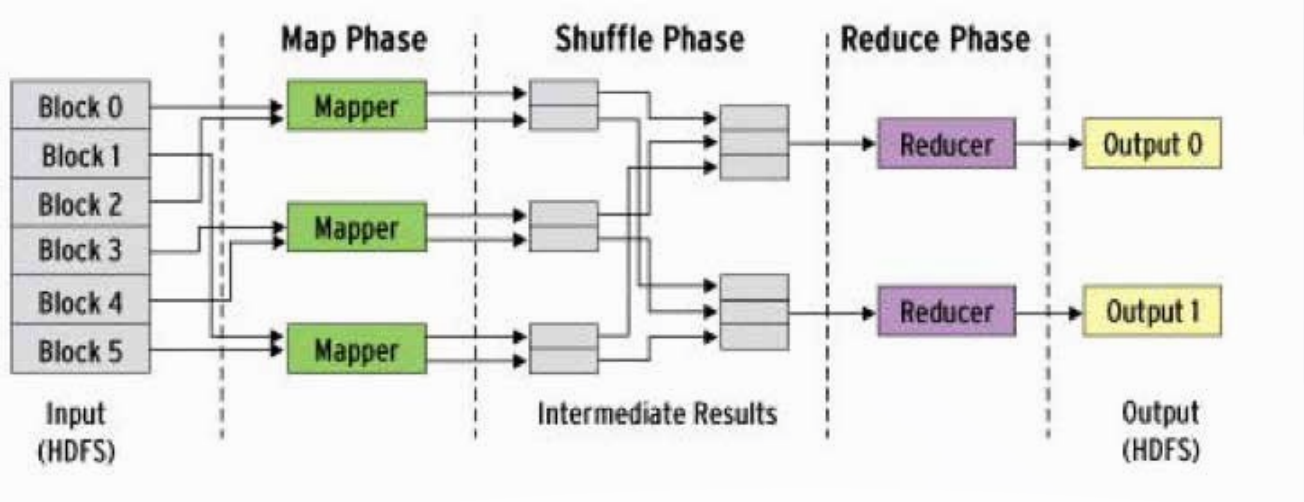
MapReduce es un modelo de programación para el procesamiento de grandes conjuntos de datos en paralelo, mediante un algoritmo de distribución en un clúster. Por lo tanto, cuando este se combina con un HDFS, se puede utilizar para manejar grandes volúmenes de datos.

Explicación

Aspectos básicos de MapReduce

Este sistema tiene como base las tecnologías de almacenamiento de datos distribuidos, en donde se llevan a cabo operaciones de tipo map y reduce en los nodos.

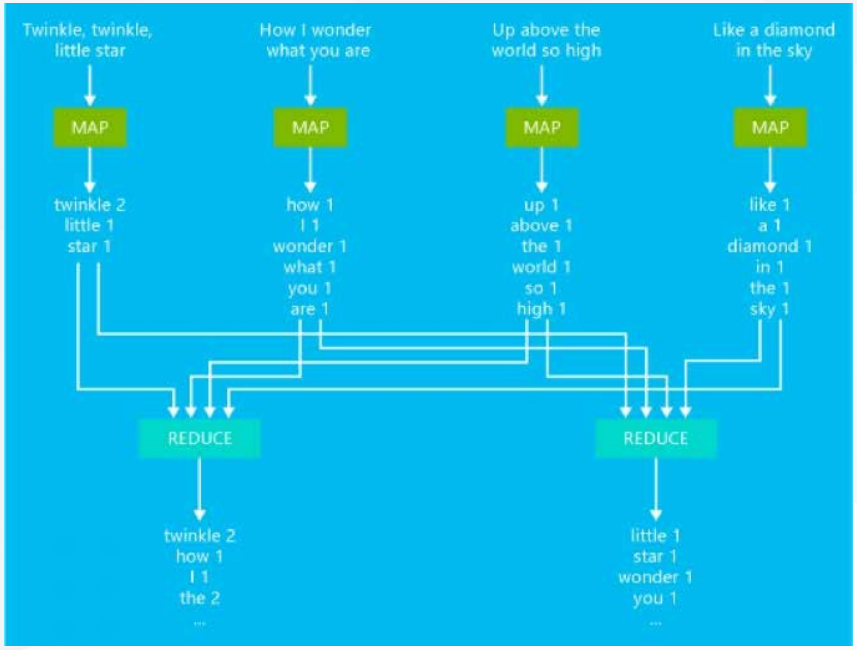
Recuerda que el HDFS es un gestor para realizar el almacenamiento de los ficheros divididos en bloques de datos, lo cual puede ser en una base de datos, un sistema externo, o bien, el mismo sistema de almacenamiento (Fernández, 2020).



Fuente: Hornung, T., Przjaciel, M., y Schätzle, A. (s.f.). *Giant Data:MapReduce and Hadoop.* Recuperado de https://www.admin-magazine.com/HPC/HPC/Articles/MapReduce-and-Hadoop/(language)/eng-US3

Explicación

Por su parte, Microsoft Azure provee un framework para escribir trabajos (o tareas) para diversos procesos de una gran cantidad de datos, en donde los datos de entrada son fraccionados en secciones independientes. Por tanto, cada sección se procesa en paralelo a través de los nodos en el clúster.



Fuente: Microsoft. (2021). *What is Apache Hadoop in Azure HDInsight?* Recuperado de https://docs.microsoft.com/en-us/azure/hdinsight/hadoop/media/apache-hadoop-introduc tion/hdi-word-count-diagram.gif

Un trabajo de MapReduce consiste en dos funciones (Microsoft, 2021):

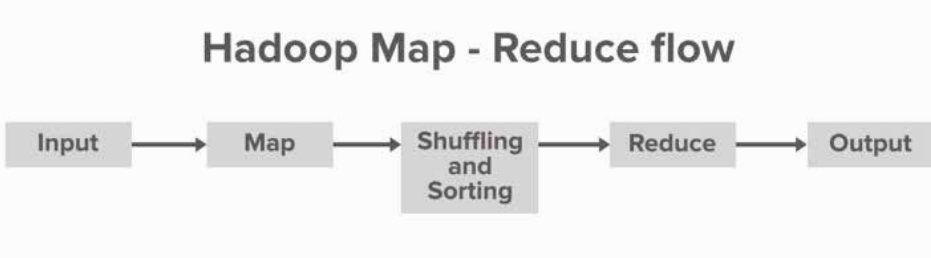
Reducer: consume las tuplas emitidas por el mapper y ejecuta un resumen de la operación para crear un resultado más pequeño, combinando resultados desde los datos del mapper.

Mapper: consume los datos de entrada, los analiza (usualmente realizando operaciones de filtrado y ordenamiento) y emite tuplas (pares llave-valor) como resultado.

Ejemplo básico de trabajo en MapReduce.

Explicación

El framework MapReduce está compuesto de los siguientes **módulos** (Moreno, 2017):



Fuente: Mali, D. (2020). *Hadoop MapReduce* Data Flow. Recuperado de https://www.geeksforgeeks.org/hadoop-mapreduce-data-flow/

`Get Input`	Divide los datos de input en secciones de tamaño apropiado y asigna una sección a cada función map.
`Función map`	Toma las tuplas de llave/valor, las procesa y genera cero o más pares de tuplas de salida.
`Función de partición`	Obtiene una secuencia *hash* de la clave de la partición.
`Función de comparación`	Ordena las entradas a la función de reducción.
`Función de reducción`	Obtiene la lista de valores de salida para pasar la clave como parámetro.
`Output writer`	Escribe la salida de la reducción en las tablas asignadas para almacenamiento.

Explicación

Administración de memoria

MapReduce permite ajustar los parámetros de memoria para cada tarea que se va a ejecutar, así como para cada subproceso que es lanzado consecutivamente, usando *mapred.{map|reduce}.child.ulimit.*

El valor para mapred.{map|reduce}.child.ulimit debe señalarse en kilobytes (KB). Asimismo, este valor debe ser más grande o igual que el -Xmx pasado a la máquina virtual y asignado antes de iniciar la máquina virtual (Hadoop, 2020).

Parámetros de memoria

Name	Type	Description
mapred.task.maxvmem	int	A number, in bytes, that represents the maximum Virtual Memory task-limit for each task of the job. A task will be killed if it consumes more Virtual Memory than this number.
mapred.task.maxpmem	int	A number, in bytes, that represents the maximum RAM task-limit for each task of the job. This number can be optionally used by Schedulers to prevent over-scheduling of tasks on a node based on RAM needs.

Fuente: Hadoop. (2020). *MapReduce Tutorial.* Recuperado de https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

Explicación

Fases en Hadoop MapReduce

En un trabajo de Hadoop MapReduce, los datos de entrada suelen separarse en porciones independientes, los cuales se gestionan por los *mappers* de forma paralela.
Posteriormente, los resultados del map son organizados (siendo estos la entrada para los *reducers*).

De manera habitual, las entradas y salidas de los trabajos se guardan en un sistema de ficheros, siendo iguales los nodos de almacenamiento y de cómputo (Fernández, 2020).

Fase Map

Aquí es donde los componentes denominados mappers generan pares (clave, valor), realizando un filtrado y transformación de los datos originales.

Fase Reduce

Administra la incorporación de los valores generados por los mappers de tipo clave-valor, en función de su clave. Cada reducer genera su fichero de salida de forma independiente.

Para conocer un ejemplo sobre el uso de MapReduce en la consola de Azure, ve el siguiente video:

Dirección de Producción de Contenidos. (2021, 27 de enero). *Big Data Tema 7* [Archivo de video]. Recuperado de https://youtu.be/qDiwYPfSg_U

Cierre

En el procesamiento de Big Data, MapReduce y Hadoop pueden ser una técnica extremadamente novedosa a la hora de desarrollar soluciones para negocios complejos y cambiantes.

Asimismo, sus usos van en ascenso, los cuales se pueden apreciar en el lanzamiento de diversos productos desarrollados por empresas líderes en el negocio del Internet, como Google y Amazon.

Bibliografía

- Fernández, O. (2020). *¿Qué es Hadoop MapReduce? Introducción.* Recuperado de https://oscarfmdc.medium.com/qu%C3%A9-es-hadoop-mapreduce-introducci%C3%B3n- aprenderbigdata-com-99021bf24a03
- Hadoop. (2020). *MapReduce Tutorial.* Recuperado de https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
- Hornung, T., Przjaciel, M., y Schätzle, A. (s.f.). *Giant Data: MapReduce and Hadoop.* Recuperado de https://www.admin-magazine.com/HPC/HPC/Articles/MapReduce-and-Hadoop/(language)/eng-US
- Mali, D. (2020). *Hadoop MapReduce – Data Flow.* Recuperado de https://www.geeksforgeeks.org/hadoop-mapreduce-data-flow/
- Microsoft. (2021). *What is Apache Hadoop in Azure HDInsight?* Recuperado de https://docs.microsoft.com/en-us/azure/hdinsight/hadoop/apache-hadoop-introduction
- Moreno, A. (2017). *¡Entiende MapReduce y Hadoop!* Recuperado de http://timeofsoftware.com/entiendemapreduceyhadoop/#:~:text=El%20framework%20MapReduce%20est%C3%A1%20compuesto,pares%20clave%2Fvalor%20de%20salida

Universidad Tecmilenio

Infraestructura para Big Data

Análisis de datos con Hadoop

Semana 11

Introducción

El análisis de datos es de suma importancia para las organizaciones, ya que gracias a este pueden tomar decisiones y estrategias de negocio de forma más ágil y precisa, permitiendo, entre otras cosas, lo siguiente (Gil, 2020):

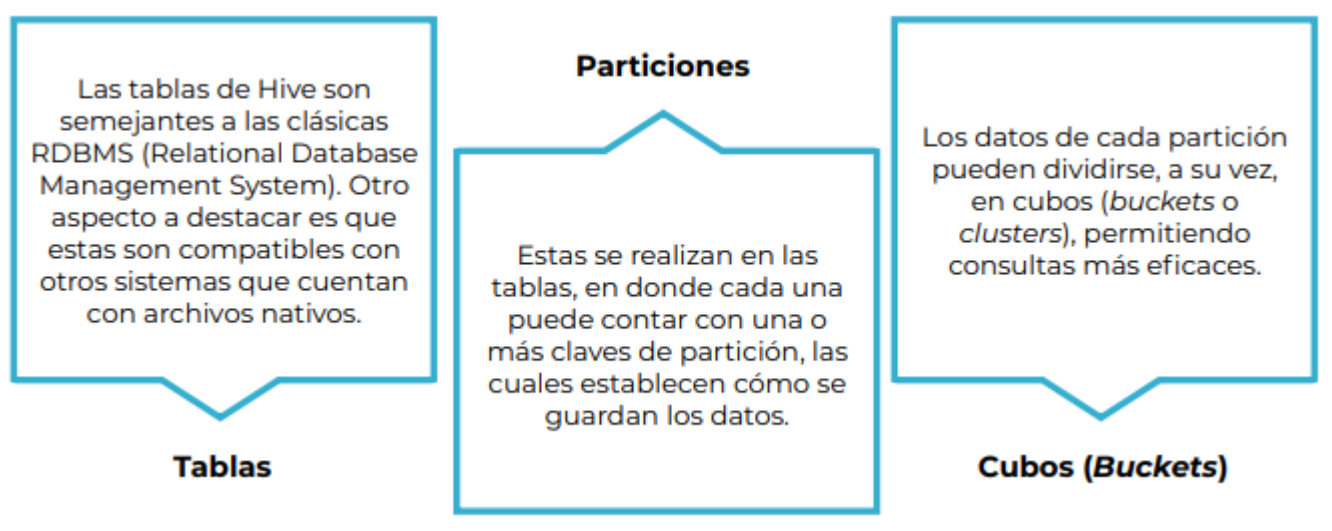
- ✓ Optimizar procesos.
- ✓ Aumentar la satisfacción tanto de los clientes como de los colaboradores.

Una de las herramientas para el análisis de datos en Big Data es Apache Hive, la cual es una infraestructura de *data warehouse* (almacén de datos) para Hadoop, que ayuda en la administración y consulta de grandes bloques de información, los cuales se alojan en un almacenamiento distribuido.

Explicación

Para comprender claramente un software, es necesario conocer sus características. En este caso, **Apache Hive** tiene tres diferentes estructuras (o formatos) para organizar los datos (Camacho, 2021):

Apache Hive también cuenta con Metastore, que es el repositorio central de metadatos.



Explicación

HiveSQL

Ahora verás algunas operaciones básicas que se pueden ejecutar con HiveSQL (HQL), como crear tablas, particiones y ejecutar condiciones lógicas o aritméticas con los datos. Usar HQL ayuda a descargar información que esté almacenada en una tabla dentro de un directorio.

Gracias al almacenamiento de datos en crudo (*data lake*, *raw data*) se puede trabajar con ellos para hacer consultas a partir de información consolidada y contextualizada.

Hadoop posibilita partir de una serie de datos crudos para hacer una lectura de datos profunda, en función de diferentes necesidades y tantas veces como sea necesario.

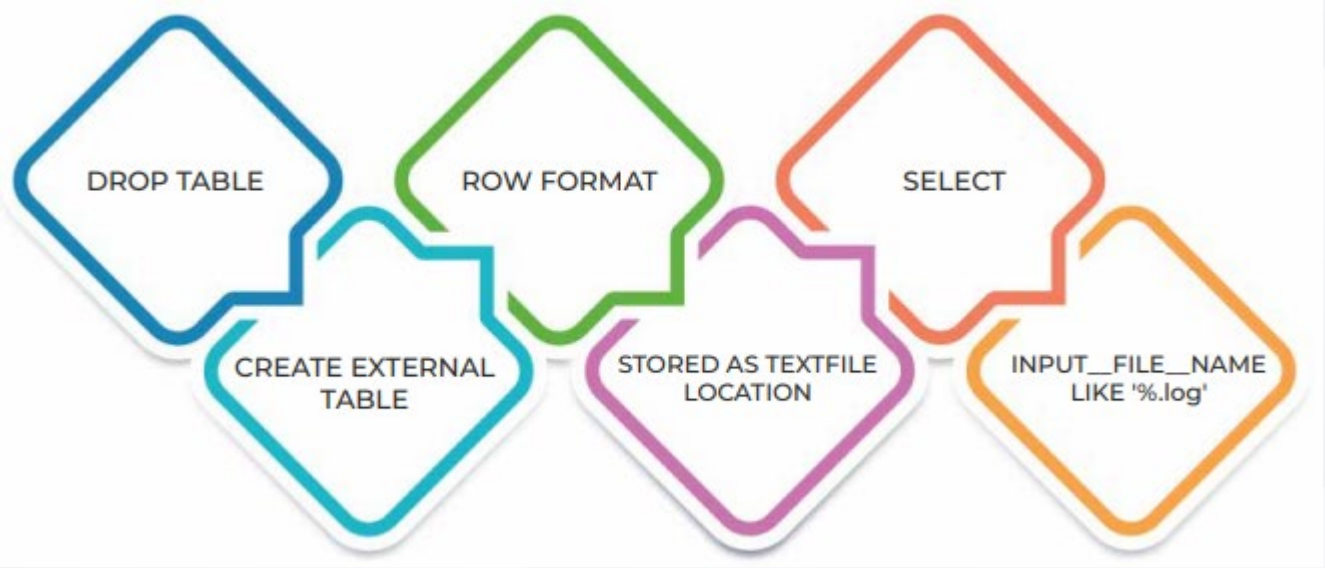
Entre las diferentes formas de utilizar Apache Hive con HDInsight, se encuentran las siguientes (Microsoft, 2020-a):

Use this method if you want...	... interactive queries	... batch processing	... from this client operating system
HDInsight tools for Visual Studio Code.	✓	✓	Linux, Unix, Mac OS X, or Windows.
HDInsight tools for Visual Studio.	✓	✓	Windows.
Hive View.	✓	✓	Any (browser based).
Beeline client.	✓	✓	Linux, Unix, Mac OS X, or Windows.
REST API.		✓	Linux, Unix, Mac OS X, or Windows.
Windows PowerShell.		✓	Windows.

Explicación

Universidad Tecmilenio

Algunos ejemplos de declaraciones de HiveQL son los siguientes:



Para conocer la descripción de cada declaración anterior, así como algunos ejemplos de consulta de Hive, revisa el siguiente enlace:

Microsoft. (2021). *What is Apache Hive and HiveQL on Azure HDInsight?* Recuperado de https://docs.microsoft.com/en-us/azure/hdinsight/hadoop/hdinsight-use-hive

Explicación

A continuación, se presentan algunos ejemplos de *scripts* de consulta de Hive para la exploración de datos (Microsoft, 2020-b):

Para conocer ejemplos, ve los siguientes videos:

Dirección de Producción de Contenidos. (2021, 27 de enero). *Big Data Tema 8 1* [Archivo de video]. Recuperado de https://youtu.be/Afhsyi0gbbQ

Dirección de Producción de Contenidos. (2021, 27 de enero*). Big Data Tema 8 2* [Archivo de video]. Recuperado de https://youtu.be/PQIS_Q3K8bk

Dirección de Producción de Contenidos. (2021, 27 de enero). *Big Data Tema 8 3* [Archivo de video]. Recuperado de https://youtu.be/KLIphUdcx9U

Script	¿Qué es lo que hace?
SELECT <partitionfieldname>, count(*) from <databasename>.<tablename> group by <partitionfieldname>;	Permite obtener el número de observaciones por partición.
SELECT to_date(<date_columnname>), count(*) from <databasename>.<tablename> group by to_date(<date_columnname>);	Permite obtener el número de observaciones por día.
SELECT distinct <column_name> from <databasename>.<tablename>	Posibilita conseguir los niveles de una columna de categorías.
<column_a>, <column_b>, count(*) from <databasename>.<tablename> group by <column_a>, <column_b>	Ayuda a obtener el número de niveles de combinación de dos columnas de categorías.
SELECT <column_name>, count(*) from <databasename>.<tablename> group by <column_name>	Permite conseguir la distribución para columnas numéricas.
SELECT a.<common_columnname1> as <new_name1>, a.<common_columnname2> as <new_name2>, a.<a_column_name1> as <new_name3>, a.<a_column_name2> as <new_name4>, b.<b_column_name1> as <new_name5>, b.<b_column_name2> as <new_name6> FROM (SELECT <common_columnname1>, <common_columnname2>, <a_column_name1>, <a_column_name2>, FROM <databasename>.<tablename1>) a join (SELECT <common_columnname1>, <common_columnname2>, <b_column_name1>, <b_column_name2>, FROM <databasename>.<tablename2>) b ON a.<common_columnname1>=b.<common_columnname1> and a.<common_columnname2>=b.<common_columnname2>	Posibilita extraer registros de la combinación de dos tablas.

Cierre

Apache Hive es uno de los componentes de software que puede ejecutarse con Hadoop, permitiendo la consulta de datos almacenados, donde la programación es similar a la que se utiliza en las bases de datos.

Entre sus beneficios encontramos que está diseñado para ser rápido en el manejo de datos y ser conocido, dado que es similar a SQL y escalable.

Bibliografía

- Camacho, J. (2021). *Hive Tutorial.* Recuperado de https://cwiki.apache.org/confluence/display/Hive/Tutorial
- Gil, J. (2020). *Tableau: La importancia del análisis de datos.* Recuperado de https://www.hiberus.com/crecemos-contigo/tableau-la-importancia-del-analisis-de-datos/
- Microsoft. (2021-a). *What is Apache Hive and HiveQL on Azure HDInsight?* Recuperado de https://docs.microsoft.com/en-us/azure/hdinsight/hadoop/hdinsight-use-hive
- Microsoft. (2021-b). *Exploración de los datos de las tablas de Hive con consultas de Hive.* Recuperado de https://docs.microsoft.com/es-es/azure/architecture/data-science-process/explore-data-hive-tables