Curso de Especialización de
Inteligencia Artificial y Big Data (IABD)

Big Data Aplicado

UD03. Ecosistema Hadoop.
Resumen.

JUAN ANTONIO GARCIA MUELAS

Recordar que, **Hadoop Core** está formado por:

- ✓ **HDFS**, que es la capa de **almacenamiento**.
- ✓ **YARN,** que es el **gestor de los procesos** que se ejecutan en el clúster.
- ✓ **MapReduce**, que es un **modelo de programación** para desarrollar tareas de procesamiento de datos.

Para dotar a Hadoop de **funcionalidades** para proyectos de Big Data, existen componentes opensouce de Apache, que junto con Hadoop core, **forman el ecosistema Hadoop**.

Cada componente es un proyecto Apache independiente.

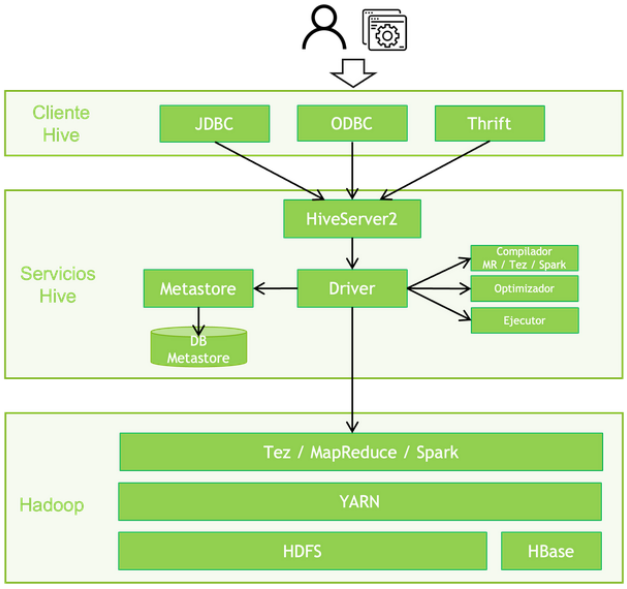
Los principales, podemos **clasificarlos en cuatro grandes grupos**:

- ✓ **Componentes de acceso y procesamiento de datos**: son aquellos que **ofrecen**, o una **forma de almacenar o acceder a los datos diferente de HDFS,** por ejemplo, permitiendo el acceso a datos aleatorios y unitarios, o mecanismos para poder procesar los datos de HDFS mediante lenguajes de consulta o mediante la programación de trabajos. **Los componentes de este tipo son:**

  - **Apache Pig**. Desarrollado por **Yahoo en 2006**. Es un motor de ejecución **sobre MapReduce (**ahora también sobre **Tez o Spark**) que ofrece un nivel de abstracción mayor, y que utiliza como **lenguaje Pig Latin**, que tiene **similitudes con SQL**. Ofrece comandos para filtrar, agrupar, leer, cargar, guardar o unir datos. Es extensible. Está **en desuso desde** la aparición de **Apache Spark**. Documentación.
  - **Apache Hive.** Desarrollado desde **Facebook**. Uno de los componentes más utilizados por usuarios no técnicos, utilizando un **lenguaje similar a SQL: HQL**. Permite **estructura relacional**, tiene mecanismos de seguridad, lee en diferentes formatos, está **orientado a consultas**. Documentación.

    Hive se ha convertido en **una de las herramientas más utilizadas** en Hadoop porque permite abrir el uso de Hadoop a todo tipo de usuarios, y porque permite integrar los datos de Hadoop al resto de herramientas de análisis de las empresas.

    Su arquitectura



Hive permite escribir aplicaciones **en varios lenguajes, incluidos Java, Python y C++.** Dispone de tres tipos de cliente:

- **Thrift Server**: es un protocolo e implementación para **publicar y consumir servicios binarios similar a RPC** para ser utilizado por cualquier lenguaje de programación.
- **Cliente JDBC**: Hive permite que las **aplicaciones Java** se conecten **mediante** el **controlador JDBC**.
- **Cliente ODBC:** el controlador ODBC de Hive permite que las aplicaciones basadas en el protocolo ODBC se conecten a Hive. **La mayoría** de las aplicaciones **que se integran con Hive utilizan este cliente** (**Excel**, **herramientas** de **Business Intelligence**, etc.).

Los **servicios se suelen ejecutar en un nodo frontera**. **Los principales** son los siguientes:

- **HiveServer**: **recibe las peticiones** de los clientes **e inicia** su **resolución**. Permite peticiones concurrentes y desde interfaces variados como JDBC, ODBC o Thrift.
- **Driver**: este componente **gobierna toda la ejecución de la petición**, utilizando el resto de servicios como Metastore, el compilador, etc.
- **Metastore**: contiene el **registro de todos los metadatos** de Hive y otros componentes que requieren aplicar un modelo sobre datos existentes en HDFS o HBase. Por ejemplo, almacena la información de las tablas definidas, qué campos tienen, con qué ficheros se corresponden, etc. Permite que aplicaciones distintas a Hive puedan usar la definición de tablas definida en Hive, o al revés.
- **Compilador**: **analiza la consulta**. Realiza análisis semánticos y verificación de tipos en los diferentes bloques de consulta y expresiones de consulta utilizando los metadatos almacenados en metastore y genera un plan de ejecución. El plan de ejecución creado por el compilador es el DAG (Gráfico acíclico dirigido), donde cada etapa es un trabajo de mapeo/reducción, operación en HDFS, una operación de metadatos. Es decir, este componente realiza la traducción de HQL a un programa MapReduce, Tez o Spark.
- **Optimizador**: realiza las **operaciones de transformación** en el plan de ejecución y divide la tarea para mejorar la eficiencia y la escalabilidad.
- **Ejecutor**: ejecuta el plan de ejecución creado por el compilador y mejorado por el optimizador en el orden de sus dependencias usando Hadoop.

Las tablas, pueden dividirse en particiones para reducir el volumen de datos a leer en diferentes operaciones, **pero el tamaño seguirá siendo el mismo.**

- **Apache Impala.** Desarrollada por **Cloudera y donada en 2015** a Apache Foundation. Implementada en **código de más bajo nivel** que Hive, ofrece **interfaz ODBC y consultas** con **HQL (Permite acceder a los datos de HDFS como si fuera una tabla.)**. También utiliza Metastore para la gestión de metadatos. Su **rendimiento respecto a Hive era superior**. Su **uso** es **muy bajo desde** la llegada de **Spark** (además de por los conflictos de recursos que mantenía con Yarn).
- **Apache HBase**. Surge para dar **soporte escalable al acceso aleatorio** a la información y para las operaciones de actualización y borrado.

  Siempre que haya que dar soporte a un sistema **operacional, Hbase** será mejor opción que Hive.

Es **tolerante a fallos**. Almacena la **información en arrays**. Por ello, **no ofrece SQL** para el acceso, sino una API, utilizando un **modelo no relacional con almacenamiento columnar** y una row-key por fila para identificarlas. Documentación.

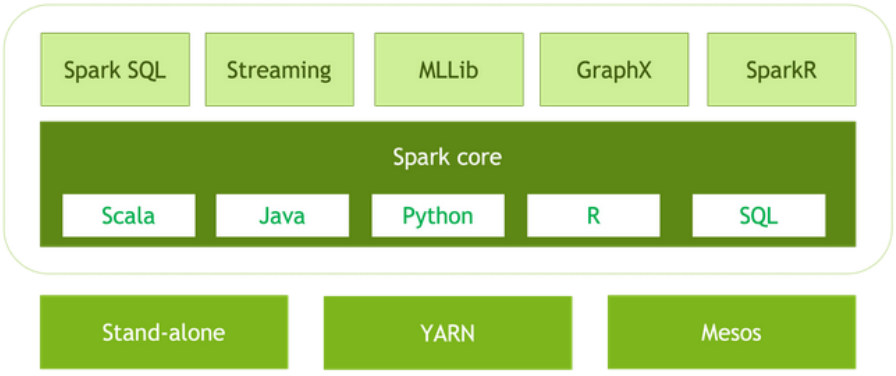
- **Apache Phoenix.** Hbase tiene como **desventaja la interfaz y el lenguaje de acceso.** Phoenix es una capa superior y **permite usar una interfaz con SQL mediante JDBC**. **Traduce las sentencias SQL en operaciones de escaneo en Hbase**. Documentación.
- **Apache Spark.** Es una **plataforma opensource** de **computación** y un conjunto de **librerías** para **procesamiento** en **paralelo** de **datos** sobre **sistemas distribuidos**. El proyecto con mayor número de contribuidores. El **estándar de facto para Big data**. Desarrollada en origen por la universidad de Berkeley y donada a la Apache Foundation en 2013.

Para saber más

Si quieres obtener más detalles sobre Spark, te recomiendo que visites dos webs:

- La página oficial de Apache de Spark.
- La página web de la empresa Databricks.

Es importante resaltar que Spark no reemplaza a Hadoop, sino que lo complementa, ya que **en la mayoría de las ocasiones**, **Spark se ejecuta sobre YARN,** es decir, sobre clústers Hadoop.



La principal abstracción de datos de Apache Spark es RDD.

Spark ofrece una serie de librerías para facilitar la construcción de aplicaciones distribuidas para propósitos más concretos:

- **Spark SQL**: librerías para **tratamiento de datos utilizando un lenguaje SQL**, con las operaciones que SQL ofrece en cuanto a agregaciones, filtrado, etc.
- **Structured Streaming**: librerías para realizar programas de **procesamiento de datos en tiempo real**.

  Tiene una **latencia mayor que Apache Flink** como norma general para procesamiento **en tiempo real**.
- **MLLib**: librerías para la **construcción de modelos de machine learning**.
- **GraphX**: librerías para la **construcción de modelos basados en grafos**, con algoritmos para facilitar sus operaciones (distancias entre nodos, caminos más cortos, etc.).
- **SparkR:** una librería para poder **conectar programas escritos en lenguaje R con** procesamiento en **Spark**.

- ✓ **Componentes de ingesta y flujos de trabajo**: son herramientas que **permiten intercambiar datos con el exterior**, es decir, tanto obtener datos del exterior, como poder exportar datos de Hadoop hacia otros sistemas, o herramientas que permiten automatizar los procesos o flujos de trabajo cuando se incorporan datos, por ejemplo, programando acciones que se ejecutan con cierta periodicidad. Los componentes de este tipo son:
  - **Apache Sqoop.** Es una herramienta diseñada para **transferir datos entre Hadoop y repositorios relacionales**, como bases de datos o sistemas mainframe. Es una de las herramientas más sencillas(Importar y exportar son las dos operaciones principales) y considerada (aunque aún en uso) que **ha llegado al final de su vida útil en 2021.** Documentación.
  - **Apache Flume.** Es un **servicio distribuido y confiable para** recoger, agregar y mover **datos generados de forma continua y atómica, como** es el caso de **los logs**. Documentación.
  - **Apache Oozie.** Facilita el **lanzamiento de flujos de trabajo. Utiliza hDPL (simil XML).** Sirve para **automatizar diferentes procesos** que se ejecutan de forma planificada dentro de un clúster Hadoop. El **ejemplo** más típico son las **ingestas de datos,** que habitualmente se realizan en **ventana nocturna,** y los posteriores procesos de transformación de datos o **enviar un email al administrador cuando una ingesta automática ha fallado**. Documentación.
- ✓ **Interfaces y herramientas de trabajo**: facilitan el trabajo con Hadoop ofreciendo interfaces visuales para poder acceder a los datos, utilizar funcionalidades o administrar el sistema. Los componentes de este tipo son:
  - **Apache Hue**. Uno de los pocos componentes que **no es gobernado por Apache**, **aunque sí es** un proyecto **opensource.** Sencillo y muy utilizado al no necesitar acceder a Hadoop por consola.

    Para saber más

    En la página oficial de Hue puedes encontrar más información sobre la herramienta.

    En esta página podrás encontrar una demo funcional de Hue, con la que poder practicar, incluso sentencias Hive.
  - **Apache Zeppelin**. Es una **herramienta web para notebooks. Muy usado por los data scientist**. Permite visualizar los resultados en modos de gráficos, guardar y compartir o colaborar con otros desarrolladores. Documentación.
  - **Apache Ambari**. Permite instalar un clúster Hadoop **mediante un wizard**, **facilitando mucho esta labor,** gestionar sus servicios, seguridad o monitorizar una gestión que sería inmanejable sin Ambari (**parar un nodo de un clúster Hadoop porque está dando problemas**).
  - **Cloudera Manager**. Muy **similar a Ambari,** pero es una herramienta propietaria de Cloudera.
- ✓ **Procesamiento en streaming**: como tipo especial, estos componentes permiten resolver los casos de uso de **análisis o procesamiento de datos en tiempo real,** es decir, cuando los datos no están en reposo, sino que se van incorporando en Hadoop y hay que realizar un análisis de estos "al vuelto". Los componentes de este tipo son:
  - Apache **Kafka**.

- Apache **Spark**.
- Apache **Flink** (structured streaming).
- Apache **Storm**.

¿Qué herramienta de procesamiento real-time elegir?

Al analizar las diferentes herramientas de procesamiento en tiempo real sobre Hadoop, suelen surgir dudas sobre cuál utilizar.

A la hora de elegir, es necesario conocer los requisitos del caso de uso o los casos de uso que vamos a implementar. En caso de que se requiera una latencia baja, lo habitual es elegir entre Apache Flink y Apache Storm.

En caso contrario, la decisión muchas veces está motivada por el conocimiento que haya en la organización, es decir, si ya estamos utilizando Apache Spark para otros casos de uso que no son de tiempo real, lo habitual es elegir Apache Spark (Structured Streaming) para no complicar la arquitectura y porque permite reaprovechar tanto código como conocimiento interno.

Para saber más

Si quieres conocer más detalles sobre los frameworks de procesamiento en tiempo real, a continuación tienes los enlaces a las páginas oficiales:

- Página oficial de Apache Storm.
- Página oficial de Apache Spark (Structured Streaming).
- Página oficial de Apache Flink.

Debes conocer

Hay muchos **tipos de roles** que trabajan dentro de una plataforma Big Data, los más importantes son los siguientes:

- El **ingeniero de datos o data engineer** es el que **hace las ingestas de los datos** en crudo, tal cual están en los sistemas origen, **y los procesa** para que puedan ser datos utilizados en los análisis.
- El **científico de datos o data scientists** es un **analista** que utilizando técnicas basadas en inteligencia artificial y otras técnicas de análisis de datos, **puede investigar y solucionar preguntas del negocio** que suelen requerir modelos predictivos, prescriptivos o incluso cognitivos.
- Los **analistas de negocio o business analyst** son personas que, teniendo un **conocimiento alto del negocio**, pero un conocimiento menor de la tecnología, **son capaces de resolver preguntas de negocio**, pero **utilizando herramientas más sencillas**, que no requieren desarrollar algoritmos ni programas complejos.

Guía práctica Hive y Pig.