Lección 5.4 Apache Pig:

En lecciones anteriores estudiamos los fundamentos del entorno Hadoop/Map Reduce, así como las herramientas que surgen sobre la base del mismo para facilitar la gestión y análisis de Big Data. Una de esas herramientas es Apache pig.

1. Introducción

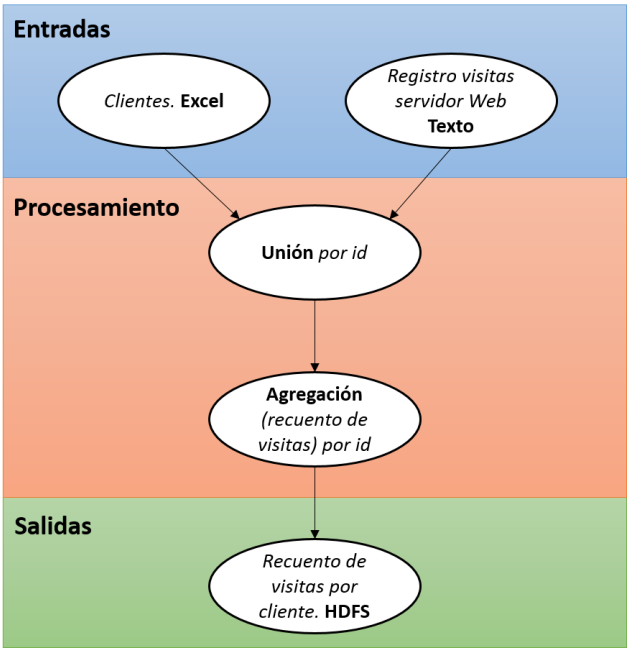
- **Apache Pig:** Plataforma de código abierto, y por tanto, extensible, para la carga, manipulación y transformación de datos en Hadoop
    - Pig Latin es el lenguaje de programación para la implementación de tales procesos
- Al igual que otras herramientas del entorno Hadoop, por ejemplo Hive, también proporciona una **capa de abstracción sobre el núcleo de Hadoop** (recordemos -> núcleo: HDFS y MapReduce)
    - Convierte, de forma automática y transparente al programador, los scripts (programas) escritos en Pig Latin en trabajos o "jobs" Map Reduce que pueden ser ejecutados en Hadoop (sobre el motor de Map Reduce) aprovechando el paralelismo de la arquitectura distribuida.

2. Pig Latin. Flujos de datos

- **Pig Latin** es un **lenguaje de flujos de datos**: Permite a los usuarios describir como, de forma paralela, los datos procedentes de una o más fuentes de datos, pueden ser leídos, procesados y escritos en una o más salidas (almacenamiento. ej: archivos HDFS)
    - Los programas de diseño de procesos ETL (Extracción, Transformación y Carga, los cuales ya vimos cuando hablamos de la integración y verificación de la calidad en módulos anteriores) se basan también en la definición de flujos de datos.
        - La implementación de procesos ETL es una las aplicaciones principales de Pig aunque no la única.
    - Estos flujos de datos pueden representar sencillos procesos, como el recuento de palabras del ejemplo que vimos cuando hablamos de Map Reduce (lección 2

de este mismo módulo), o complejos flujos de datos donde múltiples entradas de datos son unidas, los datos son particionados en múltiples flujos y procesados por diferentes operadores.

- **De forma matemática, un script de PIG describe un DAG (Grafo Acíclico Dirigido)**, donde las aristas son los datos y los nodos son los operadores que permiten procesar los datos.
    - **Hace que Pig Latin sea muy diferente de cualquier lenguaje de programación que hayamos visto antes.**
        - **No tiene sentencias condicionales (ej. if else) ni permite definir búcles**



3. Pig Latin. Operadores y Funciones

El lenguaje de implementación de flujos de datos Pig Latin ...

- Incluye operadores y funciones para la mayoría de las operaciones de manipulación de datos tradicionales (como las que ya comentamos en la lección 4.5 cuando hablamos de los procesos de integración de datos)
    - Unión o “join”, ordenación “sort” o filtrado “filter”, agregación “group by” …

- Pero, al mismo tiempo, permite a los usuarios desarrollar sus propias funciones para la lectura, transformación y escritura de datos
    - **Las denominadas funciones de usuario UDF,** característica que soportan muchas de las arquitecturas y herramientas que dan soporte a Big Data
- Además existe un repositorio llamado Piggybank[1] donde la comunidad de desarrolladores de Apache Pig contribuye al proyecto añadiendo nuevas funciones

4. ¿Para qué es útil PIG?

Nuestra experiencia como desarrolladores de aplicaciones Big Data para Inteligencia de Negocio nos ha llevado a considerar Pig como una de las herramientas más útiles del entorno Hadoop.

En este sentido, los usos principales de esta herramienta son:

- **Desarrollo de procesos ETL** (Extracción, Transformación y Carga)
    - Más potencia y control sobre los flujos de datos que herramientas de diseño de ETL que soportan ejecución en paralelo sobre Map Reduce, como Pentaho Data Integration, sin aumentar mucho la complejidad del diseño de los ETL
- **Exploración o investigación de datos en bruto**
    - Permite trabajar con fuentes de datos en las que el modelo de datos es desconocido, incompleto o inconsistente
    - Facilita el trabajo con datos anidados (ej. en estructuras XML o JSON)
    - Soporta la Integración durante el análisis (característica también denominada como integración “al vuelo” (on the fly) por algunos autores)
        - Permite realizar análisis sobre los datos antes de que estos hayan sido limpiados y cargados en un almacén de datos (ej. DW Big Data como Hive o una BD con tecnología MPP)

[1] https://cwiki.apache.org/confluence/display/PIG/PiggyBank

5. Algunas consideraciones

- Pig está orientado al procesamiento de lotes[2] de grandes volúmenes de datos de forma secuencial[3] (al igual que Map Reduce pues al final Pig solo es una abstracción y los scripts se transforman en procesos MapReduce).
    - Si necesitamos procesar Gigabytes o Terabytes de datos Pig es una buena elección
    - Pero si queremos recuperar y actualizar (update) un registro o un pequeño grupo de ellos en orden aleatorio (no secuencial), Pig no es buena lección. En ese caso lo más recomendado es el uso de una BD NoSQL (ej. HBase, Cassandra, MongoDB)
- Dado que Apache Pig se ejecuta sobre el núcleo de Hadoop (HDFS y MapReduce) la entrada de datos se hace desde y hacia el clúster HDFS
    - Archivos almacenados HDFS con cualquier nivel de estructura y formato: JSON, XML, texto, CSV
    - También soporta entrada de datos (y salida en algunos casos) de tablas del almacén de datos Hive o la BD NoSQL Hbase.
        - Gracias a la herramienta HCatalog (la cual ya citamos brevemente en la lección anterior)
- Pig no dispone de conectores con bases de datos relacionales o con herramientas de BI
    - Ej. No podemos conectar con una herramienta de BI externa para realizar análisis mediante tablas OLAP o cuadros de mando.
    - En ese caso, tras la exploración y transformación de los datos, podemos elegir un destino como Hive o HBase que si disponen de conectores (ej. ODBC o API's de Servicios Web)

[2] Batch processing, recordemos el módulo de las fuentes de datos
[3] Al igual que Map Reduce pues al final Pig solo es una abstracción y los scripts se transforman en procesos MapReduce