

**Tipo** : Lectura  
**Capítulo** : El Big Data y la Ciencia de Datos

---

## I. OBJETIVO

Ampliar sus conocimientos sobre Big Data

## II. LECTURAS COMPLEMENTARIAS

Se sugiere revisar los siguientes enlaces para profundizar en los conceptos tratados en el presente capítulo.

a) Big Data y sus componentes

### ***Componentes de una plataforma Big Data (Fuente: IBM)***

Las organizaciones han atacado esta problemática desde diferentes ángulos. Todas esas montañas de información han generado un costo potencial al no descubrir el gran valor asociado. Desde luego, el ángulo correcto que actualmente tiene el liderazgo en términos de popularidad para analizar enormes cantidades de información es la plataforma de código abierto Hadoop.

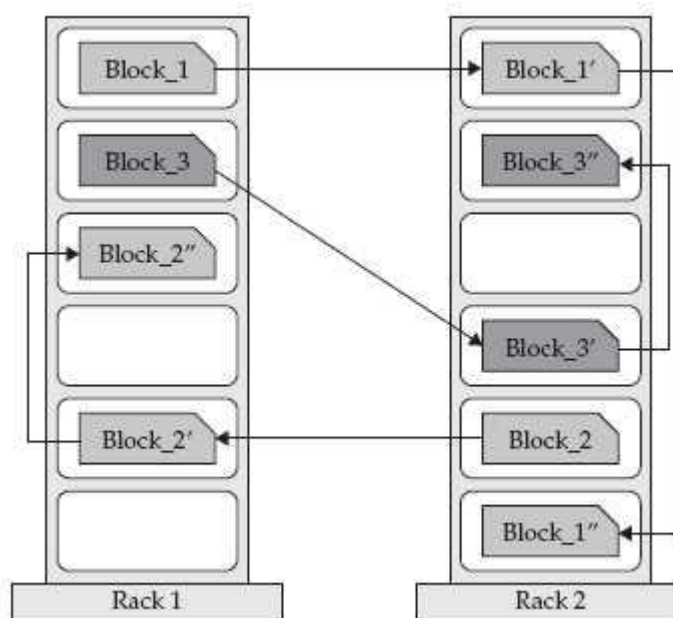
Hadoop está inspirado en el proyecto de Google File System(GFS) y en el paradigma de programación MapReduce, el cual consiste en dividir en dos tareas (mapper – reducer) para manipular los datos distribuidos a nodos de un clúster logrando un alto paralelismo en el procesamiento. Hadoop está compuesto de tres piezas: Hadoop Distributed File System (HDFS), Hadoop MapReduce y Hadoop Common.

### ***Hadoop Distributed File System(HDFS)***

Los datos en el clúster de Hadoop son divididos en pequeñas piezas llamadas bloques y distribuidas a través del clúster; de esta manera, las funciones map y reduce pueden ser ejecutadas en pequeños subconjuntos y esto provee de la escalabilidad necesaria para el procesamiento de grandes volúmenes.

La siguiente figura ejemplifica como los bloques de datos son escritos hacia HDFS. Observe que cada bloque es almacenado tres veces y al menos un bloque se almacena en un diferente rack para lograr redundancia.

Figura 2. Ejemplo de HDFS



### Hadoop MapReduce

*MapReduce* es el núcleo de Hadoop. El término MapReduce en realidad se refiere a dos procesos separados que Hadoop ejecuta. El primer proceso map, el cual toma un conjunto de datos y lo convierte en otro conjunto, donde los elementos individuales son separados en tuplas (pares de llave/valor). El proceso reduce obtiene la salida de map como datos de entrada y combina las tuplas en un conjunto más pequeño de las mismas. Una fase intermedia es la denominada Shuffle la cual obtiene las tuplas del proceso map y determina que nodo procesará estos datos dirigiendo la salida a una tarea reduce en específico.

La siguiente figura ejemplifica un flujo de datos en un proceso sencillo de MapReduce.

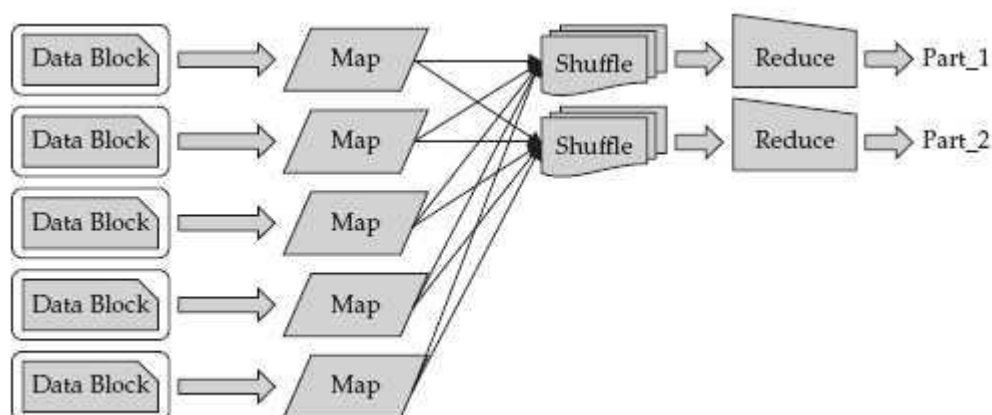


Figura 3. Ejemplo de MapReduce

## **Hadoop Common**

Hadoop Common Components son un conjunto de librerías que soportan varios subproyectos de Hadoop.

Además de estos tres componentes principales de Hadoop, existen otros proyectos relacionados los cuales son definidos a continuación:

### **Avro**

Es un proyecto de Apache que provee servicios de serialización. Cuando se guardan datos en un archivo, el esquema que define ese archivo es guardado dentro del mismo; de este modo es más sencillo para cualquier aplicación leerlo posteriormente puesto que el esquema está definido dentro del archivo.

### **Cassandra**

Cassandra es una base de datos no relacional distribuida y basada en un modelo de almacenamiento de <clave-valor>, desarrollada en Java. Permite grandes volúmenes de datos en forma distribuida. Twitter es una de las empresas que utiliza Cassandra dentro de su plataforma.

### **Chukwa**

Diseñado para la colección y análisis a gran escala de "logs". Incluye un toolkit para desplegar los resultados del análisis y monitoreo.

### **Flume**

Tal como su nombre lo indica, su tarea principal es dirigir los datos de una fuente hacia alguna otra localidad, en este caso hacia el ambiente de Hadoop. Existen tres entidades principales: sources, decorators y sinks. Un source es básicamente cualquier fuente de datos, sink es el destino de una operación en específico y un decorator es una operación dentro del flujo de datos que transforma esa información de alguna manera, como por ejemplo comprimir o descomprimir los datos o alguna otra operación en particular sobre los mismos.

### **HBase**

Es una base de datos columnar (column-oriented database) que se ejecuta en HDFS. HBase no soporta SQL, de hecho, HBase no es una base de datos relacional. Cada tabla contiene filas y columnas como una base de datos relacional. HBase permite que muchos atributos sean agrupados llamándolos *familias de columnas*, de tal manera que los elementos de una familia de columnas son almacenados en un solo conjunto. Eso es distinto a las bases de datos relacionales orientadas a filas, donde todas las columnas de una fila dada son almacenadas en conjunto. Facebook utiliza HBase en su plataforma desde noviembre del 2010.

### **Hive**

Es una infraestructura de data warehouse que facilita administrar grandes conjuntos de datos que se encuentran almacenados en un ambiente distribuido. Hive tiene definido un lenguaje similar a SQL llamado Hive Query Language(HQL), estas sentencias HQL son separadas por un servicio de Hive y son enviadas a procesos MapReduce ejecutados en el cluster de Hadoop.

El siguiente es un ejemplo en HQL para crear una tabla, cargar datos y obtener información de la tabla utilizando Hive:

```
1 CREATE TABLE Tweets (from_user STRING, userid BIGINT, tweettext STRING, retweets INT)
2 COMMENT 'This is the Twitter feed table'
3 STORED AS SEQUENCEFILE;
4 LOAD DATA INPATH 'hdfs://node/tweetdata' INTO TABLE TWEETS;
5 SELECT from_user, SUM(retweets)
6 FROM TWEETS
7 GROUP BY from_user;
```

### Jaql

Fue donado por IBM a la comunidad de software libre. Query Language for Javascript Object Notation (JSON) es un lenguaje funcional y declarativo que permite la explotación de datos en formato JSON diseñado para procesar grandes volúmenes de información. Para explotar el paralelismo, Jaql reescribe los queries de alto nivel (cuando es necesario) en queries de "bajo nivel" para distribuirlos como procesos MapReduce.

Internamente el motor de Jaql transforma el query en procesos map y reduce para reducir el tiempo de desarrollo asociado en analizar los datos en Hadoop. Jaql posee de una infraestructura flexible para administrar y analizar datos semiestructurados como XML, archivos CSV, archivos planos, datos relacionales, etc.

### Lucene

Es un proyecto de Apache bastante popular para realizar búsquedas sobre textos. Lucene provee de librerías para indexación y búsqueda de texto. Ha sido principalmente utilizado en la implementación de motores de búsqueda (aunque hay que considerar que no tiene funciones de "crawling" ni análisis de documentos HTML ya incorporadas). El concepto a nivel de arquitectura de Lucene es simple, básicamente los documentos (*document*) son divididos en campos de texto (*fields*) y se genera un índice sobre estos campos de texto. La indexación es el componente clave de Lucene, lo que le permite realizar búsquedas rápidamente independientemente del formato del archivo, ya sean PDFs, documentos HTML, etc.

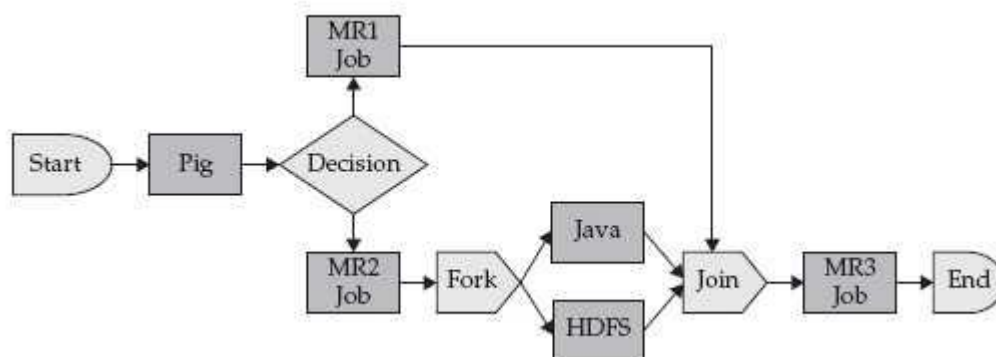
### Oozie

Como pudo haber notado, existen varios procesos que son ejecutados en distintos momentos los cuales necesitan ser orquestados para satisfacer las necesidades de tan complejo análisis de información.

Oozie es un proyecto de código abierto que simplifica los flujos de trabajo y la coordinación entre cada uno de los procesos. Permite que el usuario pueda definir acciones y las dependencias entre dichas acciones.

Un flujo de trabajo en Oozie es definido mediante un grafo acíclico llamado *Directed Acyclical Graph (DAG)*, y es acíclico puesto que no permite ciclos en el grafo; es decir, solo hay un punto de entrada y de salida y todas las tareas y dependencias parten del punto inicial al punto final sin puntos de retorno. Un ejemplo de un flujo de trabajo en Oozie se representa de la siguiente manera:

Figura 4. Flujo de trabajo en Oozie



### Pig

Inicialmente desarrollado por Yahoo para permitir a los usuarios de Hadoop enfocarse más en analizar todos los conjuntos de datos y dedicar menos tiempo en construir los programas MapReduce. Tal como su nombre lo indica al igual que cualquier cerdo que come cualquier cosa, el lenguaje *PigLatin* fue diseñado para manejar cualquier tipo de dato y *Pig* es el ambiente de ejecución donde estos programas son ejecutados, de manera muy similar a la relación entre la máquina virtual de Java (JVM) y una aplicación Java.

### **ZooKeeper**

ZooKeeper es otro proyecto de código abierto de Apache que provee de una infraestructura centralizada y de servicios que pueden ser utilizados por aplicaciones para asegurarse de que los procesos a través de un cluster sean serializados o sincronizados.

Internamente en ZooKeeper una aplicación puede crear un archivo que se persiste en memoria en los servidores ZooKeeper llamado *znode*. Este archivo *znode* puede ser actualizado por cualquier nodo en el cluster, y cualquier nodo puede registrar que sea informado de los cambios ocurridos en ese *znode*; es decir, un servidor puede ser configurado para "vigilar" un *znode* en particular. De este modo, las aplicaciones pueden sincronizar sus procesos a través de un cluster distribuido actualizando su estatus en cada *znode*, el cual informará al resto del cluster sobre el estatus correspondiente de algún nodo en específico.

Como podrá observar, más allá de Hadoop, una plataforma de Big Data consiste de todo un ecosistema de proyectos que en conjunto permiten simplificar, administrar, coordinar y analizar grandes volúmenes de información.