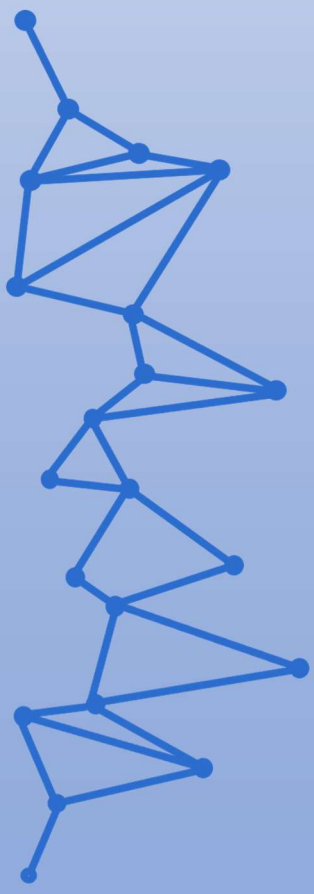
Curso de Especialización de
Inteligencia Artificial y Big Data (IABD)



Big Data Aplicado

UD01. Introducción a Apache Hadoop.
Resumen.

JUAN ANTONIO GARCIA MUELAS

Google File System (octubre 2003), almacena petabytes a bajo coste utilizando un modelo de almacenamiento distribuido. En 2004 publican como resolver el procesamiento sobre conjuntos de datos voluminosos mediante MapReduce.

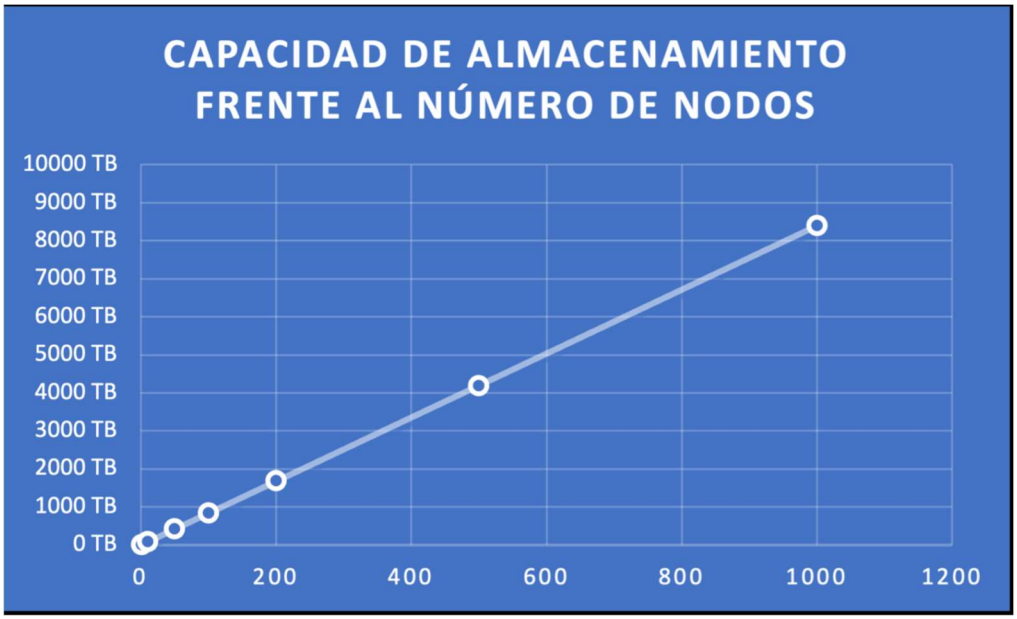
Surgía por esa época Apache Nutch (https://nutch.apache.org/) como motor de búsqueda. Poco después (2006) se llamaría Hadoop.

Hadoop es una plataforma que permite almacenar y procesar grandes volúmenes de datos.

Documentación: https://hadoop.apache.org/docs/stable/

Apache Hadoop es una **plataforma opensource** que ofrece la capacidad de **almacenar** y **procesar**, a "bajo" **coste**, grandes **volúmenes** de datos, sin importar su **estructura**, en un entorno **distribuido**, **escalable** y **tolerante a fallos**, basado en la utilización de **hardware commodity** y en un paradigma acercamiento del **procesamiento a los datos**.

- ✓ El coste es más bajo que otros sistemas tradicionales de gestión de datos.
- ✓ Hadoop es escalable tanto en almacenamiento como en procesamiento.
- ✓ Hadoop no tiene unos requerimientos de hardware muy específicos.



Íñigo Sanz (Dominio Público)

Los **componentes core** principales de Hadoop son HDFS y YARN:

- ✓ **HDFS:** un sistema de ficheros (capa de almacenamiento) que almacena los datos en una estructura basada en espacios de nombres (directorios, subdirectorios, etc.).
- ✓ **YARN:** un gestor de recursos (capa de procesamiento) que permite ejecutar aplicaciones sobre los datos almacenados en HDFS.
- ✓ **MapReduce**: un sistema de procesamiento masivo de datos que se puede utilizar directamente, programando sobre su API, o indirectamente, con aplicaciones que lo utilizan de forma transparente.

Sin embargo, normalmente se identifica el nombre Hadoop con todo el ecosistema de **componentes independientes** que suelen incluirse para dotar a Hadoop de funcionalidades necesarias en proyectos Big Data empresariales, como puede ser la ingesta de información, el acceso a datos con lenguajes estándar, o las capacidades de administración y monitorización.

Estos componentes suelen ser proyectos opensource de Apache.

Los **principales componentes** o proyectos asociados al **ecosistema Hadoop** son los siguientes:

Nombre	Descripción
Apache Hive	Permite **acceder a ficheros de datos estructurados** o **semiestructurados** que están en HDFS como si fueran una tabla de una base de datos relacional, utilizando un lenguaje similar a SQL.
Apache Pig	Utilidad para definir **flujos de datos de transformación** o consulta mediante un lenguaje de scripting.
Apache HBase	Base de datos **NoSQL** de tipo columnar que permite el acceso aleatorio, atómico y con operaciones de edición de datos.
Apache Flume	Componente para ingestar streams de datos procedentes de sistemas real-time en Hadoop.
Apache Sqoop	Componente para **importar o exportar datos estructurados** desde bases de datos relacionales a Hadoop y viceversa.
Apache Oozie	Herramienta que **permite definir flujos de trabajo** en Hadoop así como su orquestación y planificación.
Apache ZooKeeper	Herramienta técnica que **permite sincronizar el estado** de los diferentes servicios distribuidos de Hadoop.
Apache Storm	Sistema de **procesamiento real-time** de eventos con baja latencia.
Apache Spark	Aunque habitualmente no se asocia al ecosistema Hadoop, Apache Spark ha sido el mejor complemento de Hadoop en los últimos años. Apache Spark es un **motor de procesamiento** masivo de datos muy eficiente que ofrece funcionalidades para ingeniería de datos, machine learning, grafos, etc.
Apache Kafka	Sistema de mensajería que permite recoger eventos en tiempo real así como su procesamiento.
Apache Atlas	Herramienta de **gobierno de datos** de Hadoop.
Apache Accumulo	Base de **datos NoSQL** que ofrece funcionalidades de acceso aleatorio y atómico.
Apache Mahout	Conjunto de librerías para desarrollo y ejecución de modelos de **machine learning** utilizando las capacidades de computación de Hadoop.
Apache Phoenix	Capa que **permite acceder a los datos** de HBase mediante interfaz **SQL**.
Apache Zeppelin	Aplicación web de notebooks que permite a los Data Scientists realizar **análisis y evaluar código** de forma sencilla, así como la colaboración entre equipos.
Apache Impala	Herramienta con funcionalidad **similar a Hive** (tratamiento de los datos de HDFS mediante SQL) pero con un rendimiento elevado (tiempos de respuesta menores).

No te preocupes si ves muchos componentes y piensas que es imposible dominar todos. En la realidad, los proyectos suelen utilizar sólo una pequeña parte de los componentes dependiendo de las necesidades.

Los más utilizados son: Apache Spark, Apache Hive y Apache Kafka, además de los componentes core: HDFS y YARN.

Para solventar las dos dificultades de instalación, surgen las **distribuciones comerciales de Hadoop,** que contienen en un único paquete la mayor parte de componentes del ecosistema, resolviendo dependencias, añadiendo incluso utilidades, e incorporando la posibilidad de contratar soporte empresarial 24x7. Es decir, una distribución comercial ofrece:

- ✓ Un "instalador" de toda la plataforma, simplificando enormemente el proceso de instalación y despliegue de la plataforma.
- ✓ Un servicio de soporte 24x7 para resolver todas las incidencias que puedan aparecer en la plataforma en producción.
- ✓ Documentación más completa que la que se puede encontrar en los proyectos Apache.

Las **principales distribuciones** que aparecieron son:

- ✓ **Cloudera**: fue la primera distribución en salir al mercado (2009) y la que ha tenido un mayor número de clientes. Utiliza la mayor parte de componentes de Apache, en algún caso realizando algunas modificaciones, y añade algún componente propietario (Cloudera Manager, Cloudera Navigator, etc.).
- ✓ **Hortonworks**: surgió en 2012 y es una distribución que contiene, sin ninguna modificación, los componentes originales de Apache. Se fusionó con Cloudera en 2018.
- ✓ **MAPR**: rehízo la mayor parte de componentes utilizando los mismos interfaces pero reimplementando el core para ofrecer un mayor rendimiento. Cerró en 2019.

Además de las distribuciones mencionadas, es necesario añadir las soluciones Hadoop-as-a-Service de los proveedores de cloud:

- ✓ Amazon Elastic Map Reduce (EMR).
- ✓ Microsoft Azure HDInsight (y evoluciones).
- ✓ Google Dataproc.

Estas soluciones permiten levantar infraestructuras elásticas en pocos minutos en modalidad pago por uso., con un coste aproximado es de 0,25 - 2 € por nodo y hora.

Estas soluciones aportan algunas **ventajas** muy interesantes:

- ✓ **Reducen considerablemente el tiempo de aprovisionamiento** (instalación, configuración y despliegue) de infraestructuras Hadoop, de meses en el caso de instalaciones en la propia infraestructura de las empresas, a minutos en un proveedor cloud. Las empresas se encuentran inmersas en procesos de transformación digital donde prima lo que se conoce como el time-to-market, es decir, la rapidez para lanzar nuevas soluciones.
- ✓ Ofrecen **elasticidad**, es decir, cuando lanzas una plataforma Hadoop en la nube, si necesitas más capacidad o potencia, el proceso de escalar o incrementar el tamaño de la infraestructura es muy sencilla, y lo mismo ocurre si deseas reducir el tamaño de la plataforma.
- ✓ Ofrecen **pago por uso**: el coste suele ser en número de servidores por las horas que están levantados, por lo que por un lado no requiere una inversión inicial importante (comprar las máquinas, contratar el soporte por un año como mínimo, contratar a una empresa especialista para la instalación, etc.), y por otro, se paga sólo por el tamaño de la plataforma, que como hemos visto, puede adecuarse a la necesidad real en cada momento (elasticidad).

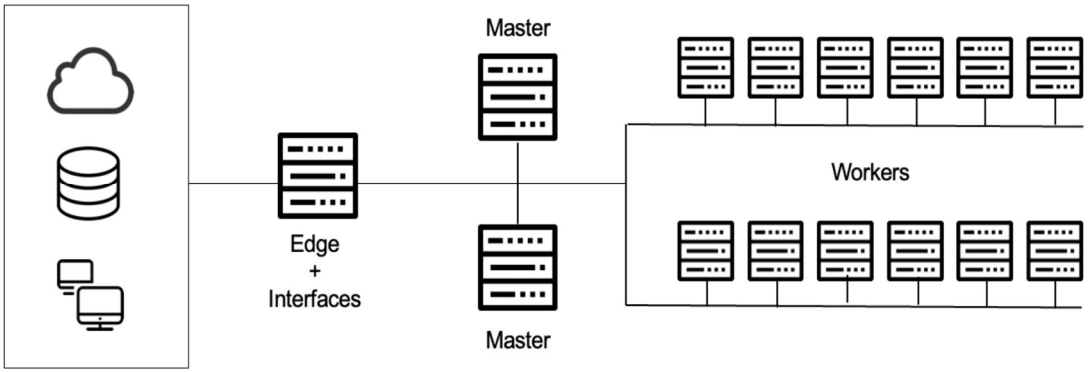
En resumen, las principales ventajas son una reducción del riesgo (no hay inversión inicial) y un incremento de la agilidad.

Sin embargo, estas soluciones cloud presentan algunas **desventajas**:

- ✓ Se produce un efecto que se denomina vendor lock-in, es decir, la barrera para salir de una solución cloud a otra de otro fabricante cloud o a un Hadoop propio, es elevada. Por ejemplo, los proveedores cloud aplican un cargo por sacar los datos fuera de su entorno.
- ✓ Las soluciones que ofrecen no suelen ser estándar, sino adaptaciones de Hadoop que han realizado los proveedores.
- ✓ El coste puede ser mucho más elevado y de hecho, difícilmente se conoce a priori al utilizar fórmulas de cálculo de los costes que añaden a veces variables que no se pueden estimar (por ejemplo, el consumo de CPU que vamos a tener).

Al un **conjunto de servidores** que trabajan en conjunto para implementar las funcionalidades de Apache Hadoop se le denomina **clúster**, y a cada uno de los servidores que forman parte del clúster se le denomina **nodo**.

A partir de ahora, cuando usemos la palabra "**clúster** de Hadoop" debes pensar en el conjunto de servidores que forman la plataforma que está en ejecución, y cuando usemos la palaba "nodo" debes pensar en cada uno de los servidores que componen el clúster.



Íñigo Sanz (Dominio público)

Los nodos pueden ser de tres tipos diferentes:

- ✓ Nodos **worker**, que realizan los trabajos. Por ejemplo, para el almacenamiento, cada worker se ocupará de almacenar una parte, mientras que para la ejecución de trabajos, cada worker realiza una parte del trabajo.
- ✓ Nodos **master,** que controlan la ejecución de los trabajos o el almacenamiento de los datos. Son los nodos que controlan el trabajo que realizan los nodos worker, por ejemplo, asignando a cada worker una parte del proceso o de los datos a almacenar, vigilando que están realizando el trabajo y no están caídos, rebalanceando el trabajo a otros nodos en caso de que un worker tenga problemas, etc.
- ✓ Nodos **edge** o **frontera** que hacen de puente entre el clúster y la red exterior y proporcionan interfaces, ya que normalmente un clúster Hadoop no tiene conexión con el resto de servidores e infraestructura de la empresa, por lo que toda la comunicación desde el exterior hacia el clúster se canaliza a través de los nodos frontera, que además, ofrecen los APIs para poder invocar a servicios del clúster.

A la hora de seleccionar el hardware para montar un clúster, hay requerimientos diferentes para los nodos maestros y los worker:

- ✓ Los **nodos master** deben ser más resistentes a fallos de hardware ya que ejecutan servicios de clúster críticos. La pérdida de un nodo master, si bien no supone a priori una pérdida de servicio, suele ser una operación compleja para los administradores. Por otro lado, como los nodos master no almacenan datos generales, sino los datos necesarios para la operativa del clúster, no tienen unos requerimientos de almacenamiento muy complejos. Por último, es preciso señalar que el tamaño del clúster determina las exigencias de los nodos master, es decir, para clusters pequeños, los nodos master deben controlar poco nodos worker, así que no requieren grandes cantidades de memoria o CPUs muy veloces, mientras que clústers con un gran número de nodos requieren nodos master muy potentes. En general, los nodos master suelen tener la siguiente configuración:
  - **Disco**: suelen disponer de 2 o 4 discos montados en RAID (RAID 1, RAID 10 o RAID 5), es decir, con modelos en los que un disco es réplica (espejo) del otro, de tal forma que en caso de fallo, se dispone de una copia exacta. La capacidad de los discos suele ser de 2 a 4 terabytes.
  - **CPU**: suelen montar 2 CPUs de 6-8 cores por CPU. Este es uno de los elementos más importantes de un sistema master, ya que los servicios que ejecuta suelen ser muy intensivos en CPU y memoria, con poco gasto de almacenamiento.
  - **Memoria**: lo habitual es disponer de una capacidad de 128 o 256 gigabytes de memoria RAM de alta calidad.
  - **Red**: la red es un elemento crítico en cualquier sistema distribuido, con diferentes nodos unidos por una red de comunicaciones, por lo que una red lenta puede suponer un cuello de botella en el rendimiento general del sistema. Lo habitual es encontrar una red de 10 gigabits por segundo en par duplicado, consiguiendo 20 gigabits, aunque no es difícil encontrar redes de alto rendimiento, como las de tipo Infiniband, que consiguen velocidades superiores a 50 gigabits por segundo.
  - **Fuente de alimentación**: lo habitual es montar fuentes de alimentación redundantes, para garantizar el suministro eléctrico en la medida de lo posible.
- ✓ En cuanto a los **nodos worker,** suelen realizar tareas de almacenamiento, para las que se intenta maximizar la capacidad de cada nodo, y de procesamiento, para las que se intenta que tenga una capacidad de ejecución alta. En general, se asume que estos nodos fallan, por lo que la inversión se realiza en almacenamiento y procesamiento, en lugar de en otro tipo de elementos que no dan rendimiento sino resiliencia (fuentes de alimentación dobles, etc.). En general, los nodos worker tienen la siguiente configuración:
  - **Disco**: los discos suelen montarse sin replicación, ya que la replicación de los datos se realiza a nivel de HDFS. Normalmente, la configuración de los discos suele ser lo que se conoce como JBOD (Just a bunch of disks = sólo un montón de discos). En esta configuración, cada disco es independiente, es decir, no hay discos que son espejo de otros, y simplemente añaden su capacidad de almacenamiento a la general del nodo. Lo habitual es que cada nodo worker tenga un elevado número de discos, habitualmente en un número similar al del

número de cores totales, por lo que es normal encontrar nodos worker con 10-12 discos de gran capacidad (3-4 terabytes).

- **CPU**: las CPUs montadas en los nodos worker suelen ser de gama media, con un par de CPUs por nodo, y 6-8 cores por CPU.
- **Memoria**: en cuanto a requerimientos de memoria, ya que ésta va a ser usada para la ejecución de tareas, el mínimo de memoria suele estar en torno a 64 gigabytes, siendo lo habitual encontrar nodos de 128 o 256 gigabytes de memoria RAM.
- **Red:** la red a la que los nodos worker están conectados suele ser la misma de los nodos master, con un ancho de banda igual, de 10-20 gigabits por segundo.
- **Fuente de alimentación**: no se suele invertir en exceso en fuentes de alimentación muy robustas o redundadas, ya que se asume que el sistema es capaz de tolerar fallos de nodos sin pérdida de servicio, siendo más interesante invertir en CPU y memoria, y resolver los problemas que pudieran aparecer en los nodos worker a demanda, sin que exista una pérdida de servicio.

Por lo tanto, el hardware típico donde se ejecuta un cluster Hadoop es el siguiente:

Tipo de nodo	Disco	CPU	Memoria	Red	Coste aproximado
Master	2 HD x 2-3 TB RAID	2 CPU x 8 cores	256 Gb RAM	20 Gbps	5.000 - 15.000 € / nodo
Worker	12 HD x 2-3 TB JBOD	2 CPU x 8 cores	256 Gb RAM		3.000 - 12.000 € / nodo
Edge	2 HD x 2-3 TB RAID	2 CPU x 8 cores	256 Gb RAM		5.000 - 10.000 € / nodo

Bajo coste

La implantación de una plataforma Hadoop tiene asociado tres tipos de coste:

- ✓ Coste del hardware: contempla la compra de los servidores, los elementos de red, etc.
- ✓ Coste del soporte empresarial: en caso de implantar una distribución, que suele ser lo habitual, los costes del soporte suelen estar en torno a 5.000 a 15.000 euros por nodo y año.
- ✓ Coste de los servicios profesionales o de consultoría para ayudar en el proceso: estos costes dependen de la complejidad de la organización y del tamaño del clúster a implantar.

A modo de ejemplo, implantar un clúster de 50 nodos worker, con 4 nodos master y 2 nodos frontera, tendría un coste aproximado de:

- ✓ Coste del hardware: aproximadamente 430.000 euros.
- ✓ Coste del soporte: aproximadamente 350.000 euros al año.

Ejercicio Resuelto

Actualmente en un banco los datos de los diferentes canales (telefónico, oficinas, banca online, etc.) no son compartidos, de manera que cuando un usuario llama al teléfono de atención del cliente para poner una reclamación, si al día siguiente va a la oficina, el director de la oficina no conoce la existencia de dicha reclamación y no puede hacer un tratamiento especial al cliente.

Asimismo, la información sobre la navegación que hacen los usuarios en la web, al ser un volumen muy grande de información (cada click se almacena por millones de usuarios y páginas vistas) no se procesa. Lo mismo ocurre con otra información como el detalle de los pagos con tarjeta (localización, comercios, etc.), que por su volumetría no se procesa.

Otra información que maneja el banco, como emails o transcripción de llamadas, por su naturaleza, no son procesadas.

¿Tendría algún beneficio desplegar una plataforma Hadoop en el banco? En caso de ser beneficioso, ¿qué casos de uso por ejemplo podrían implementarse que ahora no se implementen?

Retroalimentación

Este caso de uso es un ejemplo típico donde implementar una plataforma Hadoop puede servir a una empresa para mejorar su negocio:

- ✓ Por limitaciones técnicas, no se está compartiendo información de un cliente entre varios canales. Hadoop podría almacenar los datos de todos los canales, habilitando tener una ficha única de cliente, y por lo tanto, permitiendo que por ejemplo, cuando un cliente va a una oficina y le atiende un gestor, éste pueda consultar toda su actividad, y en el caso que se menciona, por ejemplo, podría preguntarle por la reclamación que puso el día anterior, ayudándole a resolverla, y por lo tanto, aumentando el nivel de satisfacción del cliente.
- ✓ Con la tecnología actual, no se está analizando la información no estructurada, por ejemplo, de las llamadas de los clientes o los emails. Hadoop podría almacenar esta información y un equipo de Data Scientists podría analizar todo el volumen de este tipo de datos para extraer automáticamente cuáles son los principales motivos de las quejas, predecir cuándo va a haber un mayor volumen de quejas, o prescribir qué acciones se podrían implementar para mejorar la satisfacción.
- ✓ Combinar toda la información de los clientes podría permitir conocer mejor a los clientes, y además, generar modelos predictivos que incorporen toda la actividad del cliente con el banco, para por ejemplo estimar qué clientes serían más propensos a contratar algún tipo de producto, o qué clientes tienen mayor riesgo de fuga.
- ✓ Se podrían desarrollar casos de uso utilizando toda la información existente como por ejemplo, decidir en qué zonas se debería instalar un cajero automático analizando la ubicación de los pagos con tarjeta o las extracciones en cajeros de la competencia, o analizar el nivel de riesgo de un comercio en base al tipo de clientes que pagan con tarjeta en el establecimiento (nivel de renta, saldo en la cuenta, etc.).

En fin, como ves, Hadoop podría ser una buena tecnología para habilitar una gran cantidad de casos de uso que con las tecnologías tradicionales, un banco no puede abordar, al menos con un coste razonable. ¿Por qué crees que los bancos fueron los primeros en utilizar masivamente Hadoop? La respuesta es obvia, la tienes en todo el texto anterior.

Ejercicio Resuelto

Una compañía de intermediación de seguros gestiona una cartera de 300.000 clientes. Para cada cliente almacena información sobre sus datos de contacto y las pólizas que tiene contratadas. Sobre estos datos, la dirección quiere tener cuadros de mando en los que poder obtener información sobre evolución de las pólizas contratadas, el desempeño de cada sucursal, etc. Además, les gustaría tener modelos predictivos que les permitan adelantarse a la demanda o prever clientes que podrían darse de baja.

¿Tendría algún beneficio desplegar una plataforma Hadoop en la compañía?En caso de ser beneficioso, ¿qué casos de uso por ejemplo podrían implementarse que ahora no se implementan?

Retroalimentación

En este caso, a priori parece que Hadoop no sería la mejor tecnología a implantar por varias razones:

- ✓ El volumen de datos es pequeño. Aunque 300.000 clientes puedan parecer muchos, cualquier base de datos relacional es capaz de manejar este volumen de información.
- ✓ Los datos son sólo estructurados, no hay necesidad de analizar datos no estructurados o de otro tipo.
- ✓ Los casos de uso que se prentende abordar, que son la elaboración de cuadros de mando o el desarrollo de modelos predictivos, pueden ser perfectamente abordables con herramientas de visualización y herramientas de machine learning que no requieren capacidades Big Data.