Procesamiento Distribuido para Big Data

Tema 1: Introducción y motivación

Autor: Dr. Pablo J.
Villacorta Iglesias

Actualizado Maro 2020

INTRODUCCIÓN

Las tecnologías Big Data surgen para dar respuesta a las nuevas necesidades de la sociedad actual. Vivimos en un mundo interconectado, en el que el 90 % de la información existente, persistida en medios de cualquier tipo, se ha creado en los últimos 2 años. El crecimiento de la información producida en el mundo, por fuentes de todo tipo tanto físicas como electrónicas, es exponencial desde hace unos años. Aunque las estimaciones acerca del volumen divergen, la siguiente gráfica muestra de manera orientativa este fenómeno:

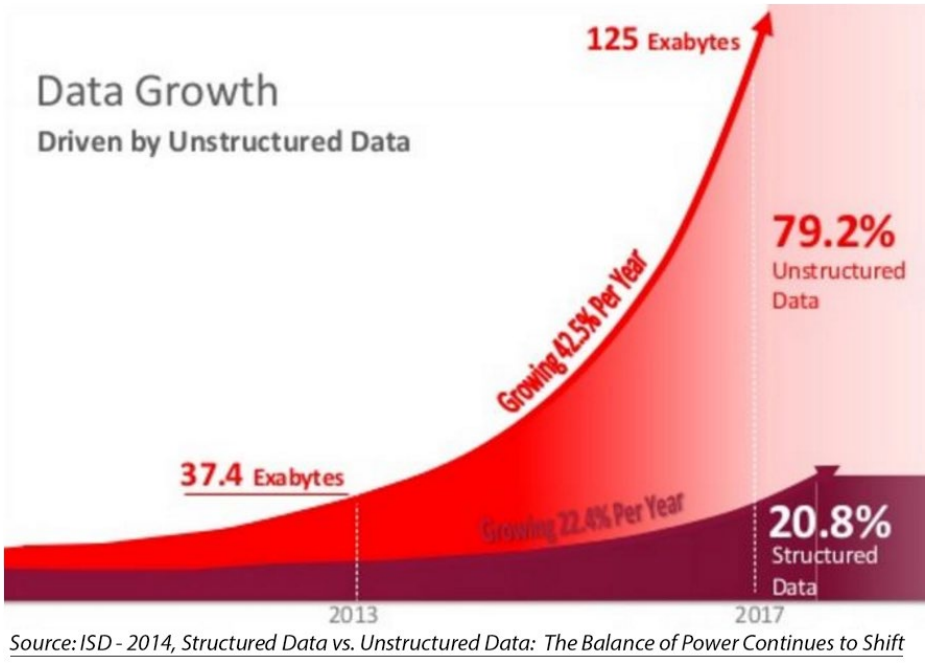

Fig. 1. Tasa de crecimiento estimado de datos en todo el mundo según Oracle

Casi el 80 % de los datos que se crean son generados por personas, y por ello suelen ser datos no estructurados. El 20 % restantes son datos estructurados generados por máquinas (datos de logs, sensores, IOT -Internet de las Cosas- en general) con el fin de ser procesados generalmente por otras máquinas.

Fuentes de datos en la actualidad

Existen principalmente tres tipos de situaciones que generan datos en la actualidad:

- La interacción entre humanos a través de un sistema informático que registra información mientras se produce la interacción. Ejemplos claros son el correo electrónico, los foros de Internet o las redes sociales, donde los datos los generamos los humanos al interactuar entre nosotros utilizando dichos medios, y los datos son almacenados y posteriormente procesados por máquinas.

- La interacción entre un humano y una máquina. El ejemplo más claro es la navegación en Internet: los servidores web generan logs con información sobre el proceso de navegación. Lo mismo ocurre al efectuar compras en alguna plataforma web de comercio electrónico o en banca online, donde cada una de nuestras transacciones queda registrada y será procesada después con el objetivo de estudiar nuestro comportamiento y ofrecernos productos mejores y más personalizados.

- La interacción entre máquinas. Varias máquinas intercambian información y la almacenan, con el objetivo de ser procesados por otras máquinas. Algunos ejemplos son sistemas de monitorización, en los que un sistema de sensores proporcionan la información que reciben a otras máquinas para que realicen algún procesado sobre ella.

Algunas cifras que resumen la cantidad de datos generados gracias a Internet se resumen en la siguiente imagen.

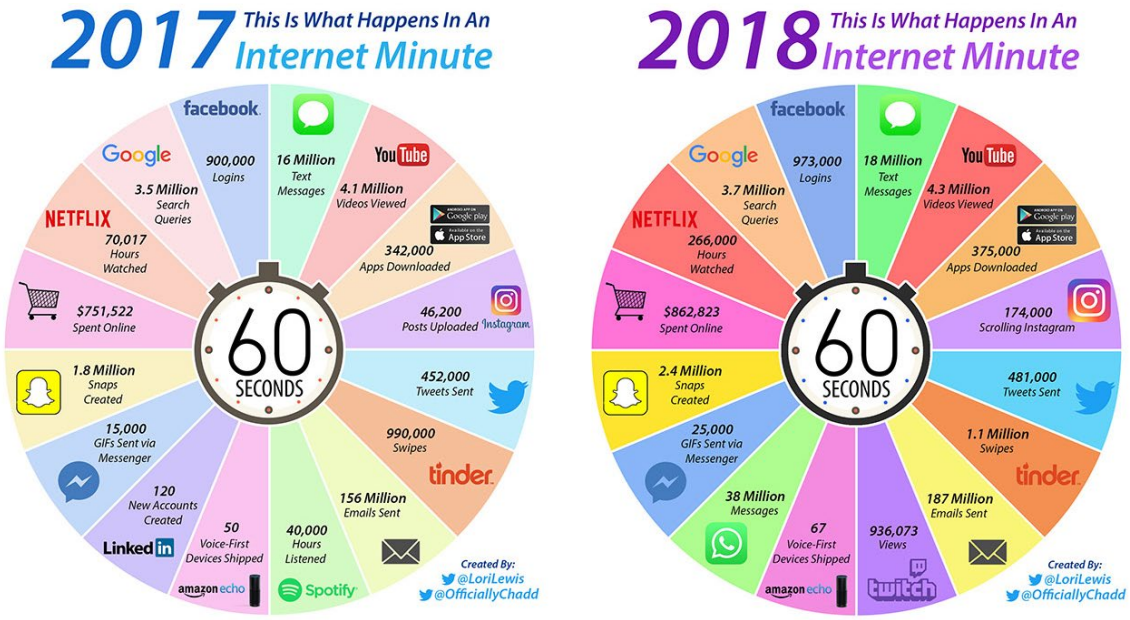

Fig 2. Eventos que suceden en Internet durante un minuto

Llama especialmente la atención el crecimiento experimentado por empresas como Netflix o Instagram.

La Transformación Digital

La conclusión global a la que podemos llegar **no** es que el mundo esté cambiando: **el mundo ya ha cambiado**, y lo podemos confirmar si examinamos hechos como los siguientes:

- La empresa que transporta a más personas en el mundo es Uber, que tiene 0 coches físicos.
- La empresa que más habitaciones reserva es Airbnb, que tiene 0 hoteles físicos.
- La empresa que más música vende es Spotify, que tiene 0 estudios de grabación.
- La empresa que vende más películas es Netflix, que tiene 0 estudios.

Estamos viendo que se producen *más interacciones digitales que físicas* entre las personas y las compañías que les dan servicio. Estas interacciones están generando datos masivos muy valiosos que hablan del comportamiento de los clientes, y permiten anticipar qué es lo que estos necesitan y van a demandar. De hecho, estamos evolucionando más rápido que las propias compañías, hasta el punto que se ha abierto una brecha entre las empresas físicas y las empresas digitales. Con el objetivo de llenar este espacio, surge la Transformación Digital.

Estamos en la Era del Cliente, marcada por una sociedad interconectada en la que hemos reinventado cómo vivimos y cómo nos relacionamos. Esta conectividad digital hace que las noticias vuelen más que nunca, para bien y para mal. Un cliente insatisfecho tiende a compartir su experiencia mucho más que un cliente satisfecho, con el consiguiente perjuicio a la imagen de una marca. Además, los clientes que cambian de proveedor en un sector suelen hacerlo por una mala experiencia como clientes, más que estrictamente por un mal servicio, y lo más frecuente es que abandonen a su proveedor sin avisar ni quejarse explícitamente.

La Transformación Digital persigue tres objetivos:

- Centrarse en el cliente: pensar continuamente en lo que necesita y en mejorar (personalizar) su experiencia y sus interacciones con la compañía. Esto requiere *ingestar y analizar grandes cantidades de datos sobre su comportamiento*.
- Centrarse en canales digitales, especialmente dispositivos móviles, puesto que las interacciones digitales son las que generan mayor cantidad de datos analizables.
- Decisiones dirigidas por los datos (*data-driven*), para lo cual es necesaria la Ciencia de (Grandes) Datos (*Big Data Science*).

DEFINICIÓN DE LAS TECNOLOGÍAS BIG DATA

Para acometer estos objetivos, la mayor parte de las tecnologías existentes hasta hace pocos años (principios de los años 2000) no eran suficientes. El motivo es la necesidad de procesar, almacenar y analizar datos con ciertas características especiales (las denominadas tres V del Big Data).

Un proyecto es Big Data cuando implica alguna de las tres V's

- Volumen: cantidades de datos lo suficientemente grandes como para no poderse procesar con tecnologías tradicionales.
- Velocidad: flujos de datos que van llegando en tiempo real y tienen que procesarse de manera continua según se van recibiendo.
- Variedad: datos de fuentes diversas, estructuradas y no estructuradas (sean BBDD relacionales, no relacionales, datos de imágenes, sonido, etc) que tienen que ser manejados y cruzados de manera conjunta.

Una definición más ajustada y realista de un proyecto con tecnologías Big Data sería:

Un proyecto es Big Data cuando la mejor manera de resolverlo (más rápida, eficiente, sencilla) implica utilizar tecnologías Big Data

Podemos definir *Big Data* como

*Conjunto de **tecnologías** y **arquitecturas** para almacenar, mover, acceder y procesar (incluido analizar) datos que eran muy difíciles o imposibles de manejar con tecnologías tradicionales*

Las causas de esta imposibilidad pueden ser:

- Cantidades ingentes de datos inimaginables hace unos años
- Datos de fuentes diversas, heterogéneas, poco estructuradas como documentos o imágenes/sonido, que aun así necesitamos almacenar y consultar (NoSQL)
- Datos dinámicos recibidos y procesados según llegan (flujos de datos o *streams*)

Nótese que en la definición anterior se omiten de manera deliberada palabras como *algoritmo, inteligencia, Ciencia de Datos* o cualquier referencia a *qué hacer o cómo analizar y explotar esos datos*. Las tecnologías Big Data son tecnologías que permiten aplicar a datos masivos técnicas que ya existían, pero son tecnologías y no técnicas en sí mismas. Las técnicas de análisis pertenecen al ámbito de la Estadística, las Matemáticas, las Ciencias de la Computación y la Inteligencia Artificial. La mayoría de estas técnicas y algoritmos han existido desde mucho antes, algunas desde mediados del siglo XX. La diferencia es que ahora pueden aplicarse a cantidades de datos mucho mayores, de naturaleza heterogénea y se pueden obtener resultados en menos tiempo.

Por supuesto, las tecnologías Big Data no tienen nada que ver con mitos como:

Fig. 3. Todo lo que NO es Big Data. Falsos mitos extendidos entre la población.

ORIGEN DE LAS TECNOLOGÍAS DE PROCESAMIENTO DISTRIBUIDO

La primera empresa que fue consciente del aumento de los datos que se estaban generando en Internet fue Google, debido a que su buscador debe ser capaz de indexar las webs nuevas para que puedan ser encontradas. En 2003 publicaron un artículo de investigación[1] que se hizo mundialmente famoso. En él explicaban el sistema de archivos distribuido *Google File System* (GFS), donde presentaban por primera vez la idea de utilizar ordenadores convencionales conectados entre sí (formando un *cluster*) para poder almacenar archivos que ocupaban más que un solo disco duro. A esto se le denomina ***commodity hardware***: máquinas no especialmente potentes, similares a las que tienen los usuarios domésticos, pueden conectarse entre sí para trabajar conjuntamente para resolver tareas de mayor envergadura. GFS fue la base del sistema de archivos distribuido HDFS que veremos en temas posteriores.

En 2004, Google publicó otro artículo[2] que se popularizó rápidamente donde explicaban un modelo de programación (MapReduce) aplicable a un cluster de ordenadores para procesar en paralelo archivos almacenados en su sistema de archivos GFS. Google también publicó una biblioteca de programación de código abierto implementando dicho paradigma. Su principal punto fuerte era la abstracción de todos los detalles de hardware, redes y comunicación entre nodos para que el usuario pudiera centrarse en el desarrollo de la lógica de la aplicación distribuida de manera sencilla. Durante muchos años, MapReduce ha sido el estándar de desarrollo de software Big Data a nivel comercial.

En 2002, el ingeniero Doug Cutting desarrolló una conocida herramienta de búsqueda llamada Nutch. Cuando fue contratado por Yahoo en 2006, lideró el proyecto Hadoop, que se independizó de Nutch e incorporó las ideas de los artículos de Google, creando así un ecosistema de herramientas y bibliotecas de programación para ejecutar MapReduce en clusters de ordenadores (principalmente en lenguaje Java). Yahoo donó Hadoop a la Apache Software Foundation en 2008 para convertirlo en un proyecto open-source. La siguiente línea temporal resume los acontecimientos de este período.

[1] http://static.googleusercontent.com/media/research.google.com/es//archive/gfs-sosp2003.pdf
[2] http://static.googleusercontent.com/media/research.google.com/es//archive/mapreduce-osdi04.pdf

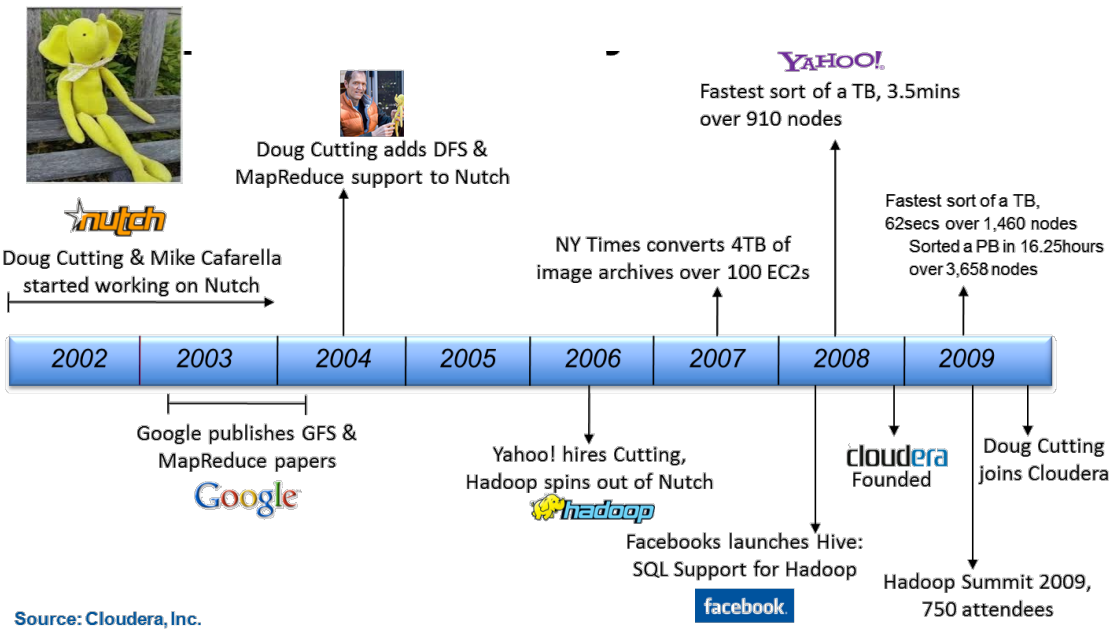

Fig. 4. Origen y evolución de Hadoop en los primeros años

El logotipo de Hadoop, un elefante amarillo, hace referencia a un elefante de peluche llamado Hadoop de una de las hijas de Doug Cutting.

Años después, y motivado por las deficiencias de Hadoop en ciertas tareas, un investigador llamado Matei Zaharia creó una nueva tecnología open-source de procesamiento distribuido llamada Apache Spark durante su tesis doctoral en Berkeley. Estudiaremos Spark en profundidad en temas posteriores. Como adelanto, MapReduce ha sido reemplazado por Spark casi en su totalidad. Las herramientas que lo utilizaban como motor de ejecución han sobrevivido gracias a que el motor de ejecución es una pieza intercambiable en muchas de ellas, y han sabido adaptarlo a Spark.

EL ECOSISTEMA HADOOP

La idea básica que hay tras las tecnologías de procesamiento distribuido es la siguiente:

> *Es posible procesar grandes cantidades de datos de forma distribuida entre varias máquinas interconectadas (cluster), cada una no necesariamente muy potente (commodity hardware). Si se necesita más potencia de cálculo o más capacidad de almacenamiento, basta con añadir más máquinas al cluster.*

Siguiendo esta filosofía, y teniendo como punto de partida el sistema de archivos distribuido GFS (que en Hadoop se transformó en HDFS – Hadoop Distributed File System) y el paradigma MapReduce, se crearon un conjunto de herramientas distribuidas, cada una con un propósito específico pero todas interoperables entre sí. Muchas de ellas están en desuso porque MapReduce ha sido reemplazado por Spark.

Fig. 5. El cluster Mare Nostrum 4, en el Barcelona Supercomputing Center (CSIC). Cada armario se denomina rack, y cada bandeja es un ordenador (nodo).

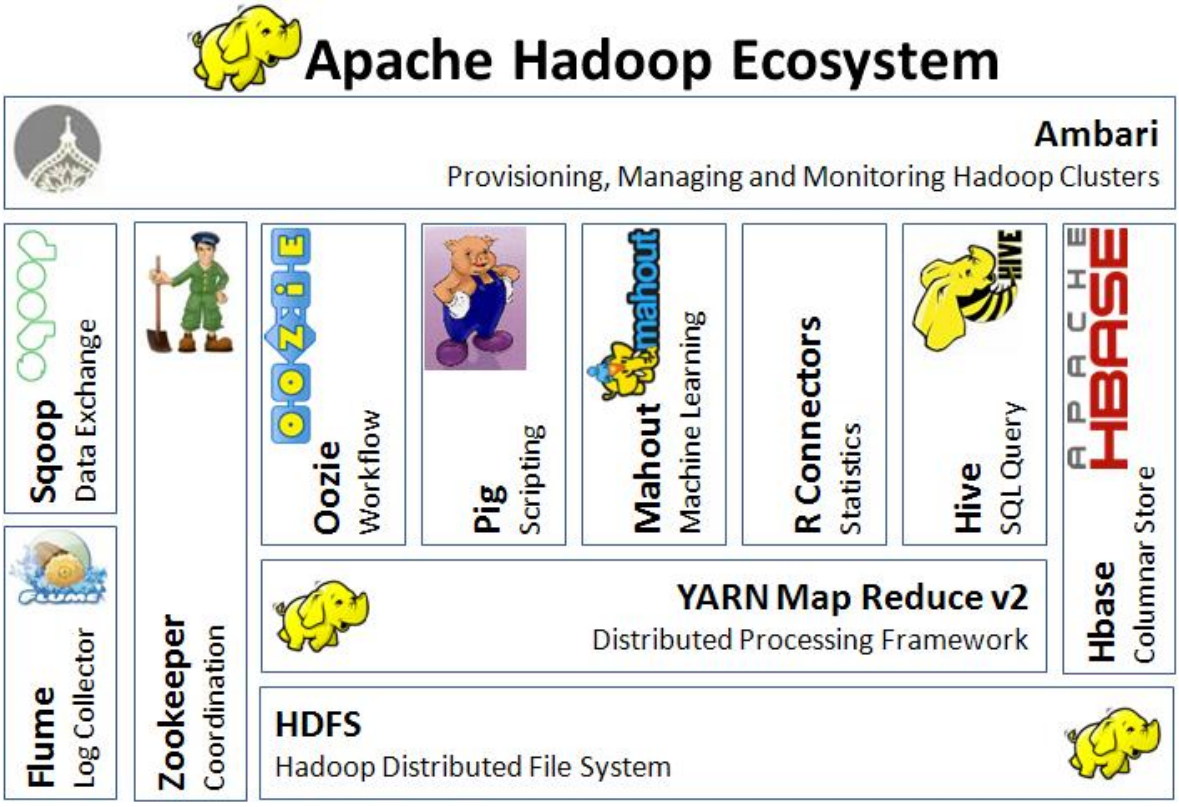

Fig. 6. El ecosistema open-source Hadoop para procesamiento distribuido

Sin ánimo de ser exhaustivos, damos una breve descripción de cada una:

- HDFS: Sistema de archivos distribuido que estudiaremos en el tema siguiente
- MapReduce: paradigma de programación para un cluster de ordenadores (forma de estructurar programas y también biblioteca de programación que se ejecuta sobre el cluster).
- Flume: herramienta para tratamiento de logs.

- Sqoop: herramienta para migración de grandes cantidades de datos desde BBDD convencionales a HDFS.
- Zookeeper: coordinador.
- Oozie: herramienta para planificación y ejecución de flujos de datos.
- Pig: herramienta para programar flujos de datos con sintaxis similar a SQL pero de mayor nivel de granularidad.
- Mahout: biblioteca de algoritmos de Machine Learning. Originalmente estaba programada con MapReduce, por lo que tenía un rendimiento pobre, pero actualmente soporta otros backend como Spark.
- R Connectors: herramientas para conectar MapReduce con el lenguaje de programación R. En desuso al igual que
- Hive: herramienta para manejar datos almacenados en HDFS utilizando lenguaje SQL. Originalmente utilizaba MapReduce como motor de ejecución. Actualmente soporta Spark y Apache Tez.
- HBase: base de datos NoSQL de tipo columnar, que permite entre otras cosas tener registros (filas) de longitud y número de campos variable.

De ellas, nos centraremos en este curso en tres herramientas:

- HDFS: (Hadoop Distributed File System): sistema de archivos distribuido inspirado en el GFS de Google, que permite distribuir los datos entre distintos nodos de un clúster, gestionando la distribución y la redundancia de forma transparente para el desarrollador que vaya a hacer uso de esos datos.

- MapReduce: manera de programar y también, motor de ejecución de tareas que corren de forma distribuida en los diferentes nodos del clúster Hadoop. La forma en la que los datos se distribuyen en diferentes sub-tareas y cómo estas se asignan a cada máquina resulta transparente para el desarrollador.

- Apache Spark: motor de procesamiento distribuido y bibliotecas de programación distribuida de propósito general, que opera siempre en la memoria principal (RAM) de los nodos del cluster. Desde hace unos años ha reemplazado totalmente a MapReduce al ser mucho más rápido.

Distribuciones de Hadoop

Al estar constituido Hadoop por un conjunto de herramientas diferentes, cada una requiere su propia instalación y configuración en el cluster para poder operar con otras ya instaladas. Este proceso era tedioso y requería bastantes conocimientos. Con el fin de simplificar esta tarea, surgieron las *distribuciones de Hadoop*, que eran conjuntos de herramientas del ecosistema Hadoop empaquetadas juntas en versiones compatibles y perfectamente interoperables entre ellas, distribuidas con un único software donde no hay necesidad de instalarlas por separado.

Empresas como Cloudera, Hortonworks (estas dos fueron competencia mutua durante mucho tiempo hasta que acabaron por fusionarse en 2018) o MapR nacieron para crear

distribuciones de Hadoop y añadirles, en algunos casos, herramientas propietarias totalmente nuevas que no pertenecían al ecosistema Hadoop, o incluso modificaciones propias del código fuente original de las herramientas de Hadoop para solucionar fallos o añadir características avanzadas. Todas las distribuciones de Hadoop de estas empresas tienen versiones open-source y de pago. La siguiente tabla compara sus características:

	Cloudera	Hortonworks	MapR
Componentes	Apache modificados y añadidos	Sólo Apache oficiales	Apache y añadidos
Versiones	Open-source (CDH) y de pago	Sólo 100 % open-source	Open-source y de pago
Sistema operativo	Linux (Windows: VM Ware)	Linux y Windows	Linux (Windows: VM Ware)
Año de creación	2008	2011	2009
Observaciones	Es la más extendida. Certificación muy popular.	Única para Windows, y única 100 % open-source	La más rápida y fácil de instalar

Tabla 1. Comparación entre las principales distribuciones de Hadoop en la actualidad

Cloudera, por ejemplo, incluye versiones de los componentes que son anteriores a la última versión oficial disponible, además de modificaciones del código oficial que solucionan incidencias reportadas pero aún no resueltas en el código oficial de las Apache. Además, incluye en su distribución ciertas herramientas propias como por ejemplo Clouder Data Science Workbench (CDSW). En el caso de Hortonworks, podemos examinar los componentes y la versión de cada uno incluida en la distribución Hortonworks Data Platform (HDP):

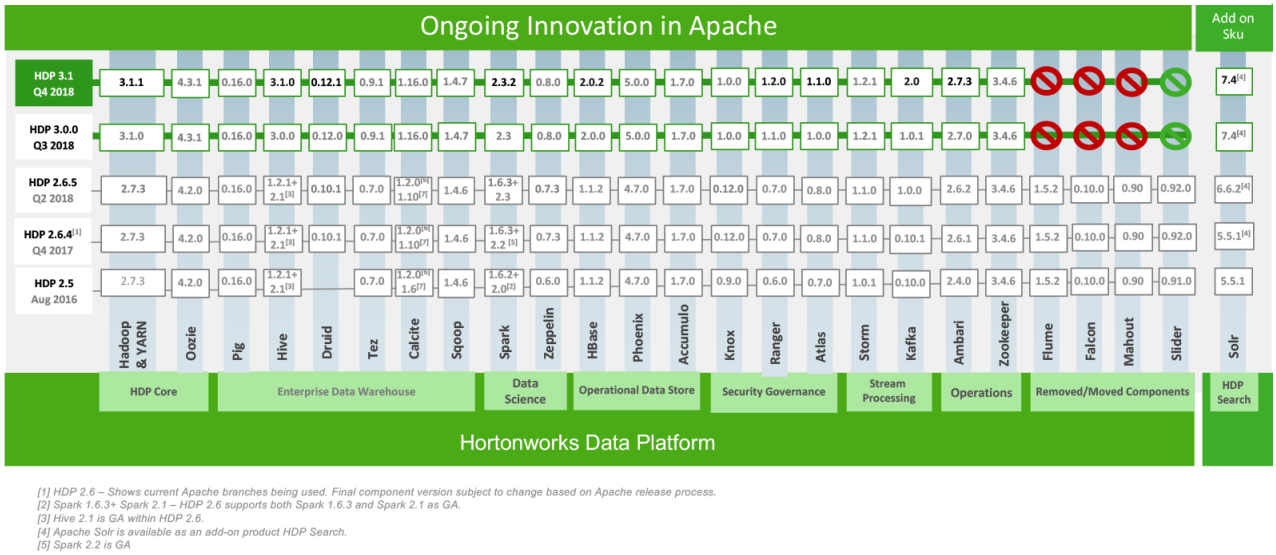

Fig. 7. Componentes de las distintas versiones de la distribución HDP

BIBLIOGRAFÍA RECOMENDADA

Tom White: *Hadoop, The Definitive Guide*. O'Reilly, 2015. Capítulo 1.